Information Technology CourseModule
Software Engineering
by Damir Dobric / Andreas Pech

Creating Text from images with OCR API

1st Akanksha Venkatesh Venkatesh Baitipuli
akanksha.venkatesh-baitipuli@stud.fra-uas.de

2nd Babitha Nadar
babitha.nadar@stud.fra-uas.de

Abstract— ***Optical Character Recognition (OCR) is a technology used for text extraction from images, but low-quality images, noise, and distortion affect its performance. This paper discusses an efficient method of enhancing the OCR accuracy to extract printed text from images by applying combinations of preprocessing techniques. OpenAI is used to compute text embeddings of the extracted text, which is further used to calculate cosine similarity between preprocessing techniques. The study is performed on images of bills, car license plates, and passport documents. The text from each of the preprocessing techniques, like binarization, grayscale, rotation, invert, canny, mirror, and histogram, shall be assessed through cosine similarity, dictionary-based accuracy, and confidence score for mean page word count to determine the best preprocessing methodology. There is a ranking method that ranks procedures based on text similarity, dictionary accuracy, and confidence without allowing misleading similarities and low-accuracy results. The proposed method ensures that only the most accurate preprocessing technique is chosen, resulting in higher OCR accuracy.***

Keywords—Optical Character Recognition (OCR), Preprocessing technique, Text Extraction, Text Embedding, Cosine Similarity, Dictionary accuracy, mean confidence.

I. Introduction

Optical character recognition (OCR) is a category of artificial intelligence that reads and understands text in images and converts it into digitized form. Unlike the human brain, which can easily recognize any text/ characters from an image. OCR is a very convenient and widely used technology in various applications. This technology, these days, normally extracts text from paper documents scanned or captured in image formats such as .png, .jpg, and .jpeg. There are various sorts of text appearing in an image based on its size, style, and complex background, making it difficult to retrieve text [1].

This recognition software was developed because of the amount of data present in the world, which is growing exponentially day by day. All this data cannot be stored physically and must be stored digitally. This can be done by using Automatic Character Recognition, which utilizes OCR. OCR is utilized for different purposes, including information extraction. OCR has been used for mail sorting, bank cheque reading, and signature verification. Besides, it can be used by organizations for automated form processing where huge data is available in printed form. Other uses of OCR also include processing utility bills, passport verification, and automated number plate recognition. But originally this software was created to help blind or specifically abled people back in 1914, which allowed these people to have handwritten text read to them out loud by a machine. The technology earlier was not advanced and did not have sufficient computing power; hence, advancing the OCR led to various difficulties concerning speed and accuracy [2]. In today's world, many OCR engines are used, including Google Drive OCR, Tesseract, Transym, and OmniPage, but most of them are paid versions.

Tesseract is an open-source deep learning-based text recognition model. Originally, it was developed by Hewlett-Packard in the 1980s. It is widely used for OCR frameworks on the internet. It operates on a step-by-step basis with different phases that can be integrated into various applications. It supports a wide variety of languages, scripts, and font coverage with defined accuracy. Tesseract shows better accuracy than Transyam OCR, although it does not always operate at the same speed. These engines are used to translate digital images into editable formats. Modern wireless technology eliminates the need for a device to have a built-in OCR system. Instead, data can be transferred over the network to be processed and results returned quickly. Online OCR is a web-based OCR converter that can act as a complete system via the Internet [2].

Tesseract OCR improves its ability to recognize text through a specific operation pipeline. The process starts by using adaptive thresholding to convert images into binary versions that separate text areas from background contents. The system proceeds to perform connected component analysis to detect character outlines that it organizes into text lines. Counted words undergo word recognition through a two-step approach where trained classifiers identify them in the initial step, and the engine follows with runtime pattern adaptations in the second step. Language independence stands as a major benefit of Tesseract since it accepts training data for Latin-based alphabets and Arabic and Chinese, and Hindi through script-specific labeling procedures. Users can enhance the recognition quality of Tesseract by creating customized models that specialize in unique fonts or document domains [3].

The OCR engine demonstrates solid performance but struggles with suboptimal image quality and textual content written by hand as well as intricate document formats that require supplementary image cleanup tools such as distortion adjustment and picture quality. The accuracy of

OCR can be dependent on preprocessing techniques and segmentation algorithms [3].

II. LITERATURE REVIEW

Optical Character Recognition (OCR) has been a research field that has been active for many decades with uses ranging from document scanning to text reading from natural photographs. The prominent use of OCR is the identification of text in images to machine-readable format to be processed and analyzed further. The accuracy of OCR systems is highly dependent upon the quality of the input image and the methods of preprocessing adopted before text recognition. This section addresses the current state of the art in OCR with a focus on preprocessing techniques, text extraction, and accuracy measures, as in recent research.

OCR technology has advanced a great deal. Earlier OCR software, the IBM 1287, could recognize only handwritten numbers and worked at a slow pace [4]. Tesseract is more precise and can recognize various fonts and languages. Tesseract is an open-source OCR engine developed by HP and later supported by Google, which has gained immense popularity because of its capability and versatility [5]. It employs a multi-stage approach, adaptive thresholding, connected component analysis, and feature extraction to extract text from an image [3]. Nevertheless, OCR software still has limitations in handling noisy, skewed, or low-quality images, for which effective preprocessing techniques are needed.

Preprocessing is crucial in OCR systems because it enhances image quality and text recognition accuracy. Several preprocessing techniques have been proposed in the literature, most of which are similar to those used in our application.

Image Rotation and Skew Correction: The skewed images can greatly affect the OCR accuracy. The Hough Transform is among the methods that have gained vast usage in identifying and correcting the skew [6]. The Hough Transform is demonstrated to correct angles smaller than 90 degrees to orient the lines of text horizontally to facilitate better recognition.

Color Space Transformations: Conversion of the image to grayscale or other color spaces like HSI (Hue, Saturation, and Intensity) can reduce the complexity of the image and improve text extraction. The importance of color space transformations for image simplification and contrast improvement, which is vital for accurate OCR. [7]

Image Inversion and Binarization: Binarization, the process of transforming images into black and white images, is a regular preprocessing operation. The thresholding technique for binarization is used to segment out text from the background. Inversion, which swaps the colors of the text and background, can also improve the accuracy of OCR, especially in cases where the text color is lighter than that of the background. [7]

Resizing and DPI Adjustment: Resizing images based on DPI resolution ensures that text is resized properly for OCR emphasizing resized images to maintain clear text, particularly for characters with a size of under 20 pixels, which are typically overlooked by OCR engines like Tesseract. Resizing images based on DPI resolution ensures that text is resized properly for OCR. Brisinello et al. [5] aimed at image resizing to maintain sharp text, particularly for text of a size smaller than 20 pixels, which is typically ignored by Tesseract and other OCR engines [7].

Noise Removal and Morphological Operations: Noise in images like speckles or uneven illumination can disrupt text recognition. Low-pass filtering and morphological operations like erosion and dilation had already been proposed to remove noise and enhance the text areas. These are particularly good techniques of improving OCR accuracy in low-quality images Noise in images, such as speckles or uneven lighting, can interfere with text recognition. The use of low-pass filters and morphological processes like erosion and dilation to remove noise and raise the text-containing areas. These are particularly good techniques of improving OCR accuracy in low-quality images [7].

Text extraction is performed after preprocessing by OCR engines like Tesseract. Tesseract employs a two-pass recognition strategy, in which words are recognized in the first pass and results are adjusted in the second pass with the assistance of an adaptive classifier [3]. The approach is precise, especially for machine-printed text. However, handwritten text remains challenging due to variability in writing style and character connectivity [8].

Later, post-processing techniques are implemented to make the accuracy more precise. These techniques are spelling correction, lexicon-based post-processing, and context analysis. Google's web-based spelling suggestion as a post-processing technique to efficiently enhance the word error rate in OCR output. These procedures provide qualitative assessments of the accuracy of OCR and aid in finding optimal preprocessing for specific applications.

Future research will continue to refine the accuracy of OCR with advanced machine learning techniques such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks that have been effective in dealing with complex text recognition processes.

III. METHODOLOGY

Our project follows a systematic approach for text extraction from images, analysis, and evaluation, presented in Figure 1. The approach has various steps: image preprocessing, Optical Character Recognition (OCR), text embedding, similarity computation, performance measurement, and final selection of the best preprocessing technique. Each step is formulated to optimize OCR accuracy, select the best preprocessing technique, and ensure strong text extraction.

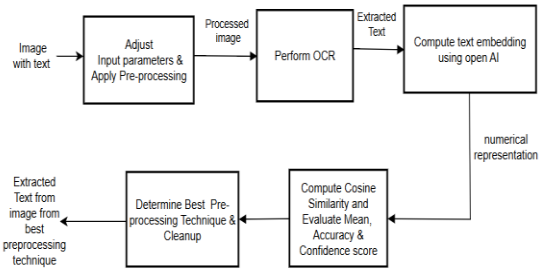


Image with text
Adjust Input parameters & Apply Pre-processing
Processed image
Perform OCR
Extracted Text
Compute text embedding using open AI
numerical representation
Compute Cosine Similarity and Evaluate Mean, Accuracy & Confidence score
Determine Best Pre-processing Technique & Cleanup
Extracted Text from image from best preprocessing technique

Figure 1: Block diagram representing the workflow of OCR

A. *Image Preprocessing*

To improve the accuracy of text extraction, preprocessing of the input image is the first step in the pipeline. Noise, distortions, and misalignment are common problems of raw images that can hinder the performance of OCR. We use a variety of preprocessing techniques systematically to rectify these problems. These methods fall into seven various approaches, one of which targets a specific problem of image quality.

Resize and Rotate: Skewed text can cause OCR recognition errors. To add precision, we rotate the image to align the text. For angle-scan or angle-photographed documents, this step is very crucial. To keep text proportions and avoid distortion, the image is resized by DPI (dots per inch) resolution after rotation. Particularly for images with very small or very large text, resizing ensures that text is of a reasonable size for OCR processing.

Canny and Invert: The Canny edge detection algorithm is used to reduce background interference and highlight text outlines in order to improve OCR accuracy. This method works especially well for photos with a lot of noise or poor contrast. To increase contrast and legibility, the colours are inverted—that is, the background is switched—after edge detection. When working with light-colored text on a darker background, this inversion step is essential.

Chain Filter: To improve OCR performance, the chain filter first uses canny edge detection to reduce noise and sharpen text boundaries. This works particularly well for photos with cluttered backgrounds or low contrast. The foreground and background colours are then reversed in an inversion step to improve text clarity. When processing images where the text appears brighter than its surroundings, this transformation is essential.

Grayscale and Binarization: Binarization further isolates text by turning the image into black and white, while grayscale conversion streamlines the image by reducing it to a single channel. This combination reduces background artifact interference and improves text clarity, making it especially helpful for images with a lot of noise or uneven lighting.

Invert: If text in an image is darker than the background, contrast is enhanced by applying inversion in this step independently. Here, for images with inverted text, like for photographs taken from shiny surfaces, this step proves to be very helpful. Inversion improves the readability of text and aids character recognition by the OCR engine by reversing the foreground and background color.

HIS Adjustment: Text visibility in images with low contrast can be improved by adjusting the image's color balance using the Hue, Saturation, and Intensity (HSI) color space transformation. This method improves the distinction between text and non-text areas, making it particularly useful for images with colored text or backgrounds.

Mirror: To provide appropriate alignment for OCR processing, images of flipped text are rectified through horizontal mirroring. The method is highly effective in the case of images from mirrored surfaces or documents that were scanned inappropriately. When the image is horizontally flipped, the text becomes aligned correctly, and the OCR becomes more accurate.

Following the application of these preprocessing techniques, the processed image is sent to the OCR engine for text extraction. Each preprocessing method is assessed separately to determine its effect on OCR accuracy, ensuring the best choice for the final text extraction.

B. *Optical Character Recognition (OCR)*

After preprocessing, Tesseract OCR SDK applies optical character recognition. There are three primary steps for the OCR process: segmentation, confidence scoring, and character recognition. First, Tesseract separates the image into characters and words using pattern recognition algorithms. Detection of text regions and their separation from non-text material, like images or noise, is the task of this step. Later, a language model is used to identify and translate each element of the text. Tesseract is flexible for multiple text recognition operations because it can recognize many languages and scripts. Lastly, each word that Tesseract recognizes gets a confidence measure that reflects the OCR engine's level of confidence in the accuracy of the text. The reliability of extracted text is approximated based on such confidence measures.

C. *Text Embedding and Similarity Computation*

We used the text-embedding-ada-002 model from OpenAI, a cutting-edge transformer-based neural network, to assess the quality of the extracted text. This model transforms text into high-dimensional numerical representations called embeddings that capture the text's semantic meaning. By allowing precise comparisons between various OCR outputs, these embeddings guarantee that the extracted text is both accurate and semantically aligned. The OpenAI API was accessed using the following configurations shown in Table 1.

Table 1: Set of configurations for Embedding

Configurations	
Key	Values
API_URL	https://api.openai.com/v1/embeddings
EMBEDDING_MODEL	text-embedding-ada-002
MEDIA_TYPE	application/json

Each text segment extracted is run through the OpenAI model, which transforms the text into a dense vector representation. The cosine similarity metric is utilized to measure the similarity of the extracted text among different preprocessing approaches. The higher the similarity metric, the more semantically similar the OCR output is to the reference text, confirming the success of the preprocessing approach.

In addition to semantic similarity, dictionary accuracy is computed, which is the ratio of words recognized that occur within a given set dictionary. It is useful in testing OCR performance by filtering out words recognized wrongly, thus offering linguistic accuracy. The mean confidence score is given by calculating the confidence level reported by the OCR engine to every recognized word and averaging it. A high confidence score indicates that the OCR engine is more confident in recognizing text, reflecting the reliability of the preprocessing technique.

By incorporating cosine similarity metrics, dictionary accuracy, and mean confidence score, the system provides a comprehensive evaluation of OCR performance. The multimodal analysis guarantees the detection of the most efficient preprocessing technique so that high-quality, reliable text extraction is achievable.

IV. IMPLEMENTATION

A. Pre-processing Configuration and Parameters

This section delves into the details of the parameters and configurations shown in Table 2 of the various preprocessing techniques mentioned in the methodology, explaining their roles, values, and how they collectively contribute to the system's performance and accuracy in our project.

Table 2: Set of preprocessing Configurations and its parameters

Preprocessing configurations	
Key	Values
InputImage	image.jpg
PreprocessingSettings.RotateAngles	[-90.0, -60.0, -45.0, -30.0, -20.0, -10.0, 10.0, 20.0, 30.0, 45.0, 60.0, 90.0, 180.0]
PreprocessingSettings.Thresholds	[50, 100, 200, 300]
PreprocessingSettings.TargetDPIs	[20.0, 10, 5.0, 0.5, 0.2]
PreprocessingSettings.SatFactors	[20.0, 10, 5.0, 0.5, 0.2]
PreprocessingSettings.IntensityFactors	20.48 (0.02 *1024)
Max(adjustable)	10

The input image file, called image.jpg, acts as the base for all text extraction and preprocessing tasks. Several preprocessing parameters are implemented to improve OCR accuracy. The preprocessing rotation angles are utilized in values spanning 90° to 180°, which ensures that skewed text is properly aligned for best OCR performance. Threshold values 50, 150, 200, and 250 are applied for image binarization in separating text from background. The image is resized as well, with target DPI values of 50, 100, 200, and 300 for better text clarity.

Preprocessing also involves saturation correction using intensity factors of 20.0, 10, 5.0, 0.5, and 0.2 that enhance or reduce the intensity of color for better readability of text, especially in low-contrast images. Similarly, intensity factors of the same values assist in brightness and contrast correction for proper lighting for text extraction. The number of preprocessing methods to be chosen is limited to N, which is taken from the maximum mean cosine similarity scores, with only effective methods being taken.

Thus, such settings were employed to test various preprocessing methods systematically and ensure that the most effective solution was obtained. Systematic testing of sets of techniques was enabled by the Preprocessing Factory and therefore ensured that the most effective solution was obtained.

B. Performance Metrics Calculation

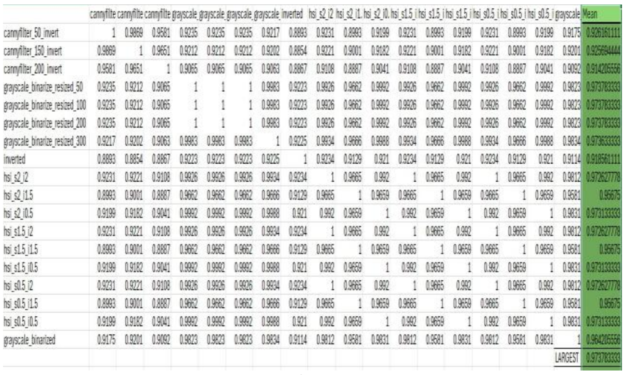
The performance of different preprocessing techniques is compared through the evaluation of OCR output obtained through every technique. This comparison process has a significant function in finding the most appropriate preprocessing technique for quality text extraction from images. The system uses three key parameters for the measurement of OCR performance, including cosine similarity, dictionary accuracy, and mean confidence score. We select the top N best preprocessing techniques based on the highest mean cosine similarity scores. These steps provide a clear idea of how effectively each preprocessing technique operates, enabling the system to select the best technique for text extraction, as shown in Figure 1.

Dictionary Accuracy: Dictionary accuracy is a metric that determines the percentage of words that are recognized and found in a particular dictionary. This metric is employed to compute OCR performance by removing misrecognized words so that the resulting text is linguistically accurate. The system reads an English dictionary from a file and verifies each word recognized against this dictionary. If a word is found in the dictionary, it is a correct recognition. The dictionary accuracy is calculated as the ratio of the number of correct words to the number of words identified. This metric is particularly helpful for finding preprocessing techniques that produce linguistically accurate results.

Mean Confidence Score: The OCR engine assigns a confidence score to each word and also calculates the mean for the whole page, indicating how sure it is that the recognition is correct. A high mean confidence score indicates that the OCR engine has higher confidence in its text recognition, meaning that the preprocessing technique used is effective. This metric is stored along with dictionary accuracy for later analysis.

C. *Performance Evaluation*

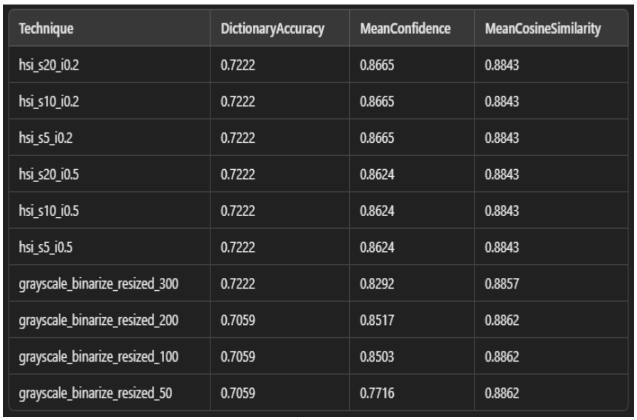
The mean cosine similarity for each technique is calculated by averaging the similarity scores compared with other preprocessing techniques, as shown in Figure 2.



	cannyfilte	cannyfilte	cannyfilte	grayscale_	grayscale_	grayscale_	grayscale_	inverted	hsi_s2_i2	hsi_s2_i1.	hsi_s2_i0.	hsi_s1.5_i	hsi_s1.5_i	hsi_s1.5_i	hsi_s0.5_i	hsi_s0.5_i	hsi_s0.5_i	grayscale	Mean
cannyfilter_50_invert	1	0.9869	0.9581	0.9235	0.9235	0.9235	0.9217	0.8893	0.9231	0.8993	0.9199	0.9231	0.8993	0.9199	0.9231	0.8993	0.9199	0.9175	0.926161111
cannyfilter_150_invert	0.9869	1	0.9651	0.9212	0.9212	0.9212	0.9202	0.8854	0.9221	0.9001	0.9182	0.9221	0.9001	0.9182	0.9221	0.9001	0.9182	0.9201	0.925694444
cannyfilter_200_invert	0.9581	0.9651	1	0.9065	0.9065	0.9065	0.9063	0.8867	0.9108	0.8887	0.9041	0.9108	0.8887	0.9041	0.9108	0.8887	0.9041	0.9092	0.914205556
grayscale_binarize_resized_50	0.9235	0.9212	0.9065	1	1	1	0.9983	0.9223	0.9926	0.9662	0.9992	0.9926	0.9662	0.9992	0.9926	0.9662	0.9992	0.9823	0.973783333
grayscale_binarize_resized_100	0.9235	0.9212	0.9065	1	1	1	0.9983	0.9223	0.9926	0.9662	0.9992	0.9926	0.9662	0.9992	0.9926	0.9662	0.9992	0.9823	0.973783333
grayscale_binarize_resized_200	0.9235	0.9212	0.9065	1	1	1	0.9983	0.9223	0.9926	0.9662	0.9992	0.9926	0.9662	0.9992	0.9926	0.9662	0.9992	0.9823	0.973783333
grayscale_binarize_resized_300	0.9217	0.9202	0.9063	0.9983	0.9983	0.9983	1	0.9225	0.9934	0.9666	0.9988	0.9934	0.9666	0.9988	0.9934	0.9666	0.9988	0.9834	0.973633333
inverted	0.8893	0.8854	0.8867	0.9223	0.9223	0.9223	0.9225	1	0.9234	0.9129	0.921	0.9234	0.9129	0.921	0.9234	0.9129	0.921	0.9114	0.918561111
hsi_s2_i2	0.9231	0.9221	0.9108	0.9926	0.9926	0.9926	0.9934	0.9234	1	0.9665	0.992	1	0.9665	0.992	1	0.9665	0.992	0.9812	0.972627778
hsi_s2_i1.5	0.8993	0.9001	0.8887	0.9662	0.9662	0.9662	0.9666	0.9129	0.9665	1	0.9659	0.9665	1	0.9659	0.9665	1	0.9659	0.9581	0.95675
hsi_s2_i0.5	0.9199	0.9182	0.9041	0.9992	0.9992	0.9992	0.9988	0.921	0.992	0.9659	1	0.992	0.9659	1	0.992	0.9659	1	0.9831	0.973133333
hsi_s1.5_i2	0.9231	0.9221	0.9108	0.9926	0.9926	0.9926	0.9934	0.9234	1	0.9665	0.992	1	0.9665	0.992	1	0.9665	0.992	0.9812	0.972627778
hsi_s1.5_i1.5	0.8993	0.9001	0.8887	0.9662	0.9662	0.9662	0.9666	0.9129	0.9665	1	0.9659	0.9665	1	0.9659	0.9665	1	0.9659	0.9581	0.95675
hsi_s1.5_i0.5	0.9199	0.9182	0.9041	0.9992	0.9992	0.9992	0.9988	0.921	0.992	0.9659	1	0.992	0.9659	1	0.992	0.9659	1	0.9831	0.973133333
hsi_s0.5_i2	0.9231	0.9221	0.9108	0.9926	0.9926	0.9926	0.9934	0.9234	1	0.9665	0.992	1	0.9665	0.992	1	0.9665	0.992	0.9812	0.972627778
hsi_s0.5_i1.5	0.8993	0.9001	0.8887	0.9662	0.9662	0.9662	0.9666	0.9129	0.9665	1	0.9659	0.9665	1	0.9659	0.9665	1	0.9659	0.9581	0.95675
hsi_s0.5_i0.5	0.9199	0.9182	0.9041	0.9992	0.9992	0.9992	0.9988	0.921	0.992	0.9659	1	0.992	0.9659	1	0.992	0.9659	1	0.9831	0.973133333
grayscale_binarized	0.9175	0.9201	0.9092	0.9823	0.9823	0.9823	0.9834	0.9114	0.9812	0.9581	0.9831	0.9812	0.9581	0.9831	0.9812	0.9581	0.9831	1	0.964205556
																		LARGEST	0.973783333

Figure 2: Mean Cosine Similarity for each Preprocessing Technique

The system then calculates the dictionary accuracy and mean confidence score for each preprocessing technique as shown in Figure 3.



Technique	DictionaryAccuracy	MeanConfidence	MeanCosineSimilarity
hsi_s20_i0.2	0.7222	0.8665	0.8843
hsi_s10_i0.2	0.7222	0.8665	0.8843
hsi_s5_i0.2	0.7222	0.8665	0.8843
hsi_s20_i0.5	0.7222	0.8624	0.8843
hsi_s10_i0.5	0.7222	0.8624	0.8843
hsi_s5_i0.5	0.7222	0.8624	0.8843
grayscale_binarize_resized_300	0.7222	0.8292	0.8857
grayscale_binarize_resized_200	0.7059	0.8517	0.8862
grayscale_binarize_resized_100	0.7059	0.8503	0.8862
grayscale_binarize_resized_50	0.7059	0.7716	0.8862

Figure 3: Calculation of the best preprocessing technique

D. *Final Selection Process*

The top N techniques are chosen based on the highest mean cosine similarity values, where N is determined by the input parameter Max. Once the top methods are identified, they are merged with their corresponding dictionary accuracy and mean confidence scores. After merging, the techniques are sorted in descending order, first by dictionary accuracy and then by mean confidence score. Suppose the selected technique has a dictionary accuracy of zero. In that case, the system re-evaluates the results and selects the top N techniques solely based on dictionary accuracy and mean confidence score, without considering cosine similarity. The system eliminates false positives and re-evaluates the results to ensure that only meaningful and accurate techniques are considered. So the best technique is chosen based on the first one from the top N techniques.

V. RESULTS

The text extraction in the project was tested on various types of documents, such as supermarket receipts, Car license plates, rotated images, mirror images, and passport ids. All of the document types present different challenges. For example, supermarket receipts contain noisy, unstructured text, whereas passports contain structured data (such as name, nationality, and passport number). Rotated and mirrored text tests the system's robustness to geometric distortion. Three key measures—Mean Confidence Score, Dictionary Accuracy, and Mean Cosine Similarity—were used to evaluate the performance of the system and identify the best preprocessing method for each case. The thorough analysis provides accurate and dependable text extraction across various real-world scenarios. We determine the best preprocessing strategy for the given cases listed below:

A. *Rotate Image*

The original input image, shown in Figure 4, contains the text that was processed by the system:

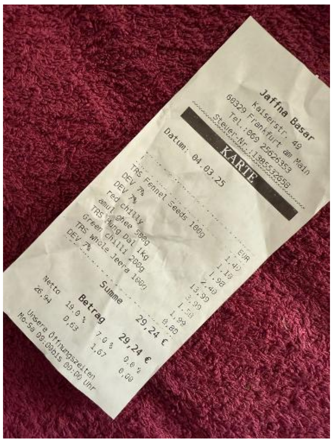
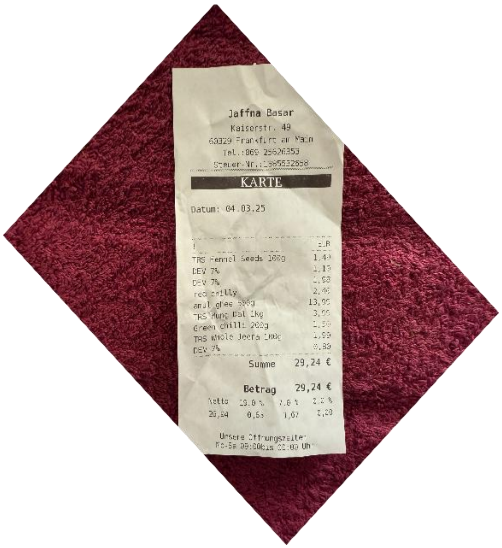


Figure 4: Original input image (rotated image)



Jaffna Basar
Kaiserstr. 49
60329 Frankfurt am Main
Tel.:069 25626353
Steuer-Nr.:1385532658

KARTE

Datum: 04.03.25

	EUR
TRS Fennel Seeds 100g	1,49
DEV 7%	1,19
DEV 7%	1,98
red chilly	2,49
amul ghee 500g	13,99
TRS Mung Dal 1kg	3,99
Green chilli 200g	1,50
TRS Whole Jeera 100g	1,99
DEV 7%	0,89
Summe	29,24 €
Betrag	29,24 €

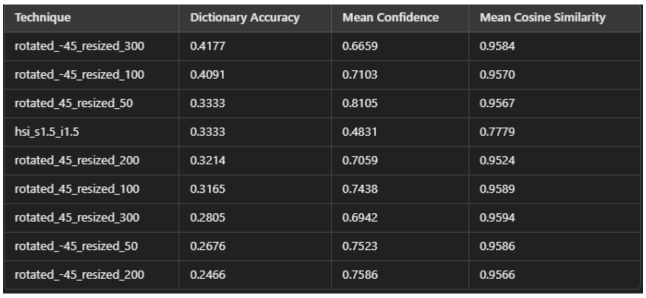
Netto	19.0 %	7.0 %	0.0 %
26,94	0,63	1,67	0,00

Unsere Öffnungszeiten
Mo-Sa 09:00bis 20:00 Uhr

Figure 5: Processed Image

Performance of Preprocessing Techniques- Figure 5 is the processed image, which refers to the modified version of the original input image after applying preprocessing techniques.

Figure 6, the table shows the comparison of top approaches as defined in Implementation Section IV [D]. As one can see, rotated 45-degree images are always higher on all measures than other transformations. For instance, rotated_45_resized_300 achieves a Mean Cosine Similarity and Dictionary Accuracy higher than other techniques like hs1_s1.5_i1. Interestingly, differences in resizing at 45-degree rotation (i.e., 50/100/200/300) reveal a trade-off: while rotated_45_resized_50 performs the optimal Mean Confidence (0.8105), its Dictionary Accuracy falls to 0.3333. This demonstrates the critical balance between rotation angle and resizing values for optimal OCR performance.



Technique	Dictionary Accuracy	Mean Confidence	Mean Cosine Similarity
rotated_-45_resized_300	0.4177	0.6659	0.9584
rotated_-45_resized_100	0.4091	0.7103	0.9570
rotated_45_resized_50	0.3333	0.8105	0.9567
hsi_s1.5_i1.5	0.3333	0.4831	0.7779
rotated_45_resized_200	0.3214	0.7059	0.9524
rotated_45_resized_100	0.3165	0.7438	0.9589
rotated_45_resized_300	0.2805	0.6942	0.9594
rotated_-45_resized_50	0.2676	0.7523	0.9586
rotated_-45_resized_200	0.2466	0.7586	0.9566

Figure 6: Comparison of the top techniques

The graph in Figure 7 shows a comparison of the top preprocessing techniques. Where the top N=10 techniques are selected based on Mean cosine similarity, they are sorted first by dictionary accuracy and later by mean confidence as mentioned in Implementation Section IV [D].

The technique rotated_45_resized_300 achieves the highest overall performance. This illustrates that rotating the image by 45 degrees and resizing it to 300 DPI produces the most accurate and reliable text extraction results.

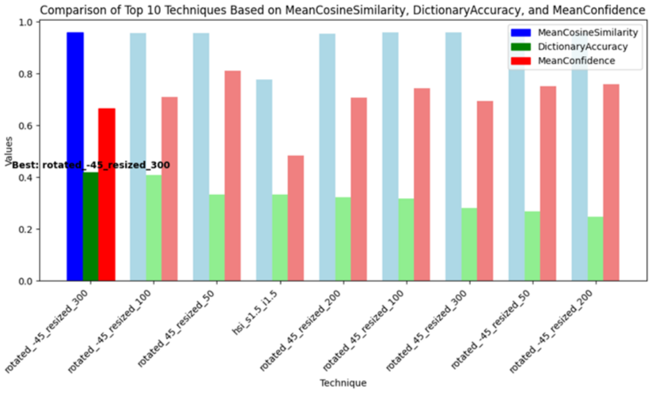
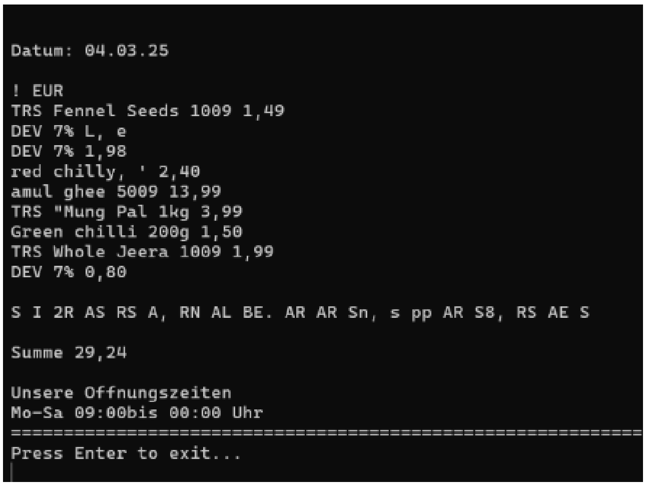


Figure 7: Visual representation of the Performance of preprocessing techniques

The console output in Figure 8 displays the final results of the best-performing technique (rotated_45_resized_300). The extracted text is shown in Figure 23



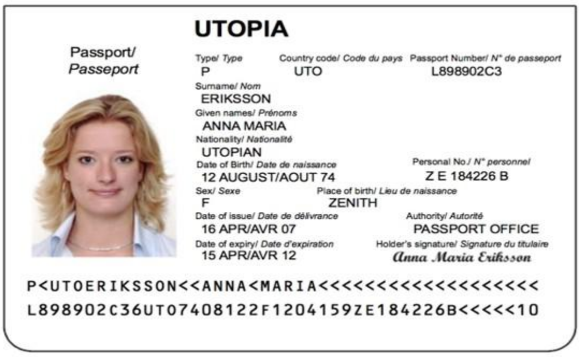
```
Datum: 04.03.25

! EUR
TRS Fennel Seeds 1009 1,49
DEV 7% L, e
DEV 7% 1,98
red chilly, ' 2,40
amul ghee 5009 13,99
TRS "Mung Pal 1kg 3,99
Green chilli 200g 1,50
TRS Whole Jeera 1009 1,99
DEV 7% 0,80

S I 2R AS RS A, RN AL BE. AR AR Sn, s pp AR S8, RS AE S

Summe 29,24

Unsere Offnungszeiten
Mo-Sa 09:00bis 00:00 Uhr
============================================================
Press Enter to exit...
```

Figure 8: Console Output (Extracted Text)

The output indicates that the system successfully extracts the text with high accuracy. This confirms that the selected preprocessing technique successfully enhances OCR performance.

B. *Passport Document*

Input Image - The original input image, shown in Figure 9, is the passport document that was processed by the system.



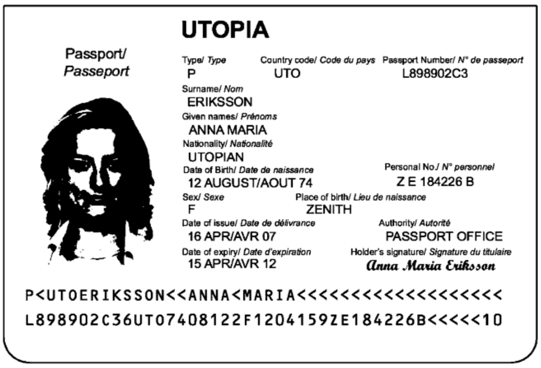
UTOPIA

Passport/
Passeport

Type/ *Type*: P
Country code/ *Code du pays*: UTO
Passport Number/ *N° de passeport*: L898902C3
Surname/ *Nom*: ERIKSSON
Given names/ *Prénoms*: ANNA MARIA
Nationality/ *Nationalité*: UTOPIAN
Date of Birth/ *Date de naissance*: 12 AUGUST/AOUT 74
Personal No./ *N° personnel*: Z E 184226 B
Sex/ *Sexe*: F
Place of birth/ *Lieu de naissance*: ZENITH
Date of issue/ *Date de délivrance*: 16 APR/AVR 07
Authority/ *Autorité*: PASSPORT OFFICE
Date of expiry/ *Date d'expiration*: 15 APR/AVR 12
Holder's signature/ *Signature du titulaire*: Anna Maria Eriksson

P<UTOERIKSSON<<ANNA<MARIA<<<<<<<<<<<<<<<<<<<
L898902C36UTO7408122F1204159ZE184226B<<<<<10

Figure 9: Original Input Image (Passport)



UTOPIA

Passport/
Passeport

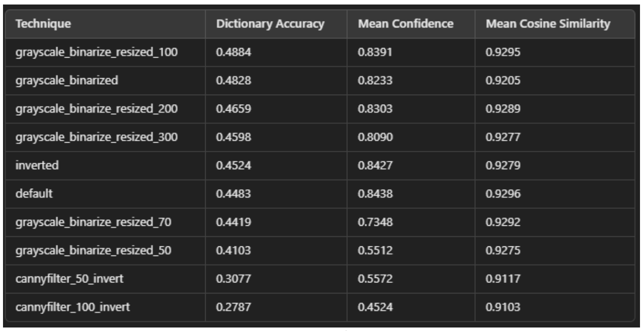
Type/ *Type*: P
Country code/ *Code du pays*: UTO
Passport Number/ *N° de passeport*: L898902C3
Surname/ *Nom*: ERIKSSON
Given names/ *Prénoms*: ANNA MARIA
Nationality/ *Nationalité*: UTOPIAN
Date of Birth/ *Date de naissance*: 12 AUGUST/AOUT 74
Personal No./ *N° personnel*: Z E 184226 B
Sex/ *Sexe*: F
Place of birth/ *Lieu de naissance*: ZENITH
Date of issue/ *Date de délivrance*: 16 APR/AVR 07
Authority/ *Autorité*: PASSPORT OFFICE
Date of expiry/ *Date d'expiration*: 15 APR/AVR 12
Holder's signature/ *Signature du titulaire*: Anna Maria Eriksson

P<UTOERIKSSON<<ANNA<MARIA<<<<<<<<<<<<<<<<<<<
L898902C36UTO7408122F1204159ZE184226B<<<<<10

Figure 10: Processed image

Performance of Preprocessing Techniques- Figure 10 refers to the modified version of the original input image after applying preprocessing techniques.

The table in Figure 11 provides a detailed comparison of the top techniques. Where the top N=10 techniques are selected as mentioned in Implementation Section IV [D]. The approach of grayscale_binarize_resized_100 is the most effective in terms of all characteristics, with a Mean Cosine Similarity of 0.9295, a Dictionary Accuracy of 0.4884, and a Mean Confidence Score of 0.8391. This means grayscale_binarize_resizing_100 gives the most accurate text extraction results for this input.



Technique	Dictionary Accuracy	Mean Confidence	Mean Cosine Similarity
grayscale_binarize_resized_100	0.4884	0.8391	0.9295
grayscale_binarized	0.4828	0.8233	0.9205
grayscale_binarize_resized_200	0.4659	0.8303	0.9289
grayscale_binarize_resized_300	0.4598	0.8090	0.9277
inverted	0.4524	0.8427	0.9279
default	0.4483	0.8438	0.9296
grayscale_binarize_resized_70	0.4419	0.7348	0.9292
grayscale_binarize_resized_50	0.4103	0.5512	0.9275
cannyfilter_50_invert	0.3077	0.5572	0.9117
cannyfilter_100_invert	0.2787	0.4524	0.9103

Figure 11: Comparison of the top techniques

The graph in Figure 12 comparatively plots the top 10 techniques with respect to their Mean Cosine Similarity, Dictionary Accuracy, and Mean Confidence Score. Grayscale_binarize_resized_100 method stands out as having the highest Mean Cosine Similarity and Mean Confidence Score, indicating its superiority in enhancing OCR for structured documents like passports.

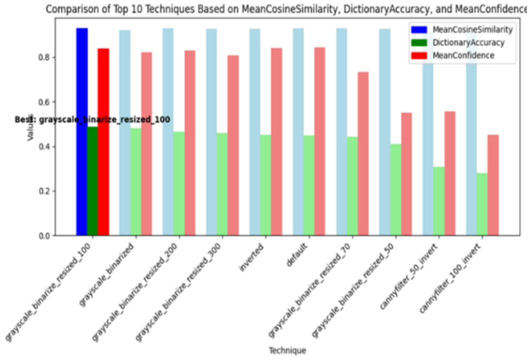
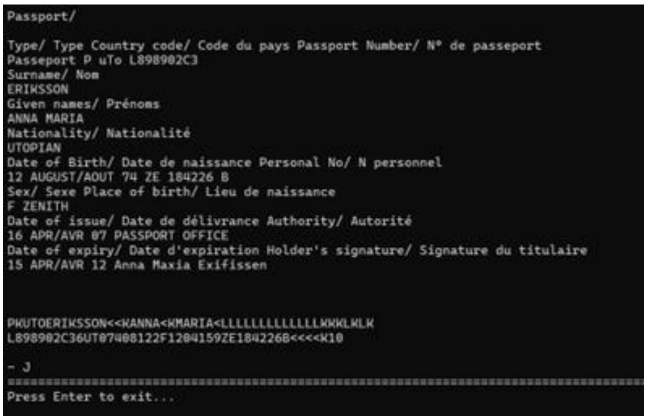


Figure 12: Visual representation of the Performance of preprocessing techniques

Final Output- The console output in Figure 13 displays the results of the best-performing technique as grayscale_binarize_resized_100.The output indicates that the system can successfully extract the text with high accuracy, as the Mean Cosine Similarity is 0.9295, Dictionary Accuracy is 0.4884, and the Mean Confidence Score is 0.8391. This shows that the grayscale_binarize_resized_100 preprocessing technique can enhance OCR performance on structured documents like passports.



```
Passport/

Type/ Type Country code/ Code du pays Passport Number/ N° de passeport
Passeport P uTo L898902C3
Surname/ Nom
ERIKSSON
Given names/ Prénoms
ANNA MARIA
Nationality/ Nationalité
UTOPIAN
Date of Birth/ Date de naissance Personal No/ N personnel
12 AUGUST/AOUT 74 ZE 184226 B
Sex/ Sexe Place of birth/ Lieu de naissance
F ZENITH
Date of issue/ Date de délivrance Authority/ Autorité
16 APR/AVR 07 PASSPORT OFFICE
Date of expiry/ Date d'expiration Holder's signature/ Signature du titulaire
15 APR/AVR 12 Anna Maxia Exifissen



PKUTOERIKSSON<<KANNA<KMARIA<LLLLLLLLLLLLLWKKLKLK
L898902C36UT07408122F1204159ZE184226B<<<<K10

- J
================================================================================
Press Enter to exit...
```

Figure 13: Console Output (Extracted Text)

C. Car License Plate

Input Image- The original input image, shown in Figure 14, contains the license plate of a car that the system processed.

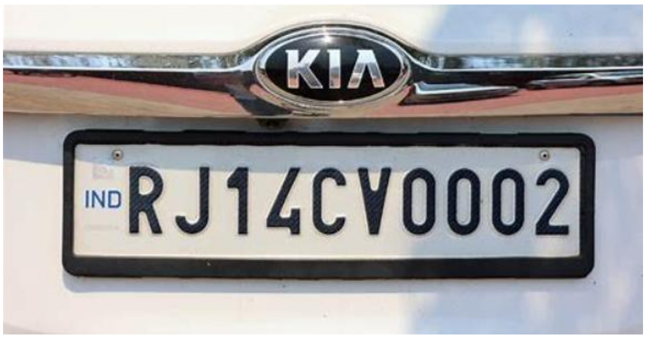


Figure 14: Original input image (License plate)

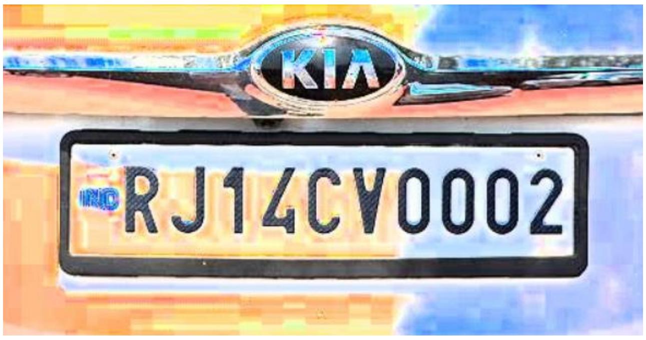
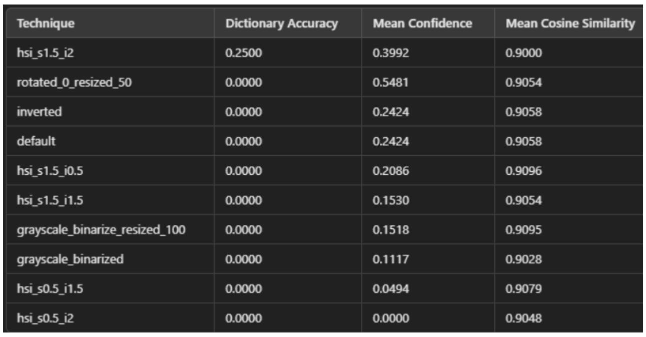


Figure 15: Preprocessed Image

Performance of Preprocessing Techniques- Figure 15 refers to the modified version of the original input image after applying preprocessing techniques.

The table in Figure 16 provides a detailed comparison of the top techniques as mentioned in the Implementation Section IV [D]. The approach of hsi_s2.5_i2 is the most effective in terms of all characteristics, with a Mean Cosine Similarity of 0.9000, a Dictionary Accuracy of 0.2500, and a Mean Confidence Score of 0.3992. This means HIS method with g 1.5 saturation and 2.0 intensity (hsi_s2.5_i2) gives the most accurate text extraction results for this input.



Technique	Dictionary Accuracy	Mean Confidence	Mean Cosine Similarity
hsi_s1.5_i2	0.2500	0.3992	0.9000
rotated_0_resized_50	0.0000	0.5481	0.9054
inverted	0.0000	0.2424	0.9058
default	0.0000	0.2424	0.9058
hsi_s1.5_i0.5	0.0000	0.2086	0.9096
hsi_s1.5_i1.5	0.0000	0.1530	0.9054
grayscale_binarize_resized_100	0.0000	0.1518	0.9095
grayscale_binarized	0.0000	0.1117	0.9028
hsi_s0.5_i1.5	0.0000	0.0494	0.9079
hsi_s0.5_i2	0.0000	0.0000	0.9048

Figure 16: Comparison of the top techniques

The graph in Figure 17 compares the top techniques against their Mean Cosine Similarity, Dictionary Accuracy, and Mean Confidence Score. The hsi_s2.5_i2 method has a Mean Cosine Similarity of 0.9571, Dictionary Accuracy of 0.3056, and Mean Confidence Score of 0.8408, illustrating how well it performs for extracting text from the license plate.

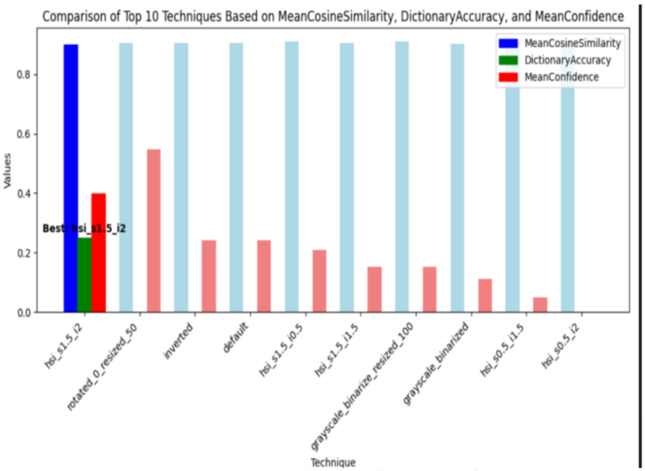
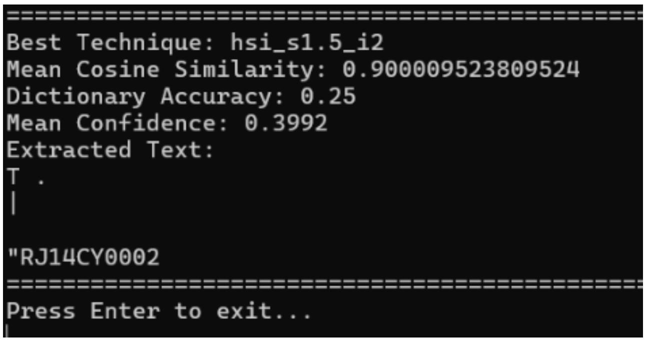


Figure 17: Visual representation of the Performance of preprocessing techniques

Final Output- The console output in Figure 18 shows that the best method is hsi_s2.5_i2.The output confirms that the system extracts text correctly with high accuracy. This confirms that the preprocessing technique of hsi_s2.5_i2 effectively enhances OCR performance.

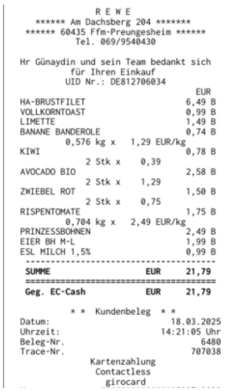


```
==============================================
Best Technique: hsi_s1.5_i2
Mean Cosine Similarity: 0.900009523809524
Dictionary Accuracy: 0.25
Mean Confidence: 0.3992
Extracted Text:
T .
|

"RJ14CY0002
==============================================
Press Enter to exit...
```

Figure 18: Console Output (Extracted Text)

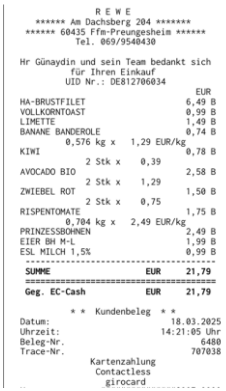
D. *Supermarket Bill*

Input Image- The original input image, shown in Figure 19, contains the supermarket bill that was processed by the system.



```
                  R E W E
     ****** Am Dachsberg 204 *******
   ****** 60435 Ffm-Preungesheim ******
             Tel. 069/9540430

Hr Günaydin und sein Team bedankt sich
            für Ihren Einkauf
          UID Nr.: DE812706034
                                    EUR
HA-BRUSTFILET                      6,49 B
VOLLKORNTOAST                      0,99 B
LIMETTE                            1,49 B
BANANE BANDEROLE                   0,74 B
         0,576 kg x   1,29 EUR/kg
KIWI                               0,78 B
             2 Stk x    0,39
AVOCADO BIO                        2,58 B
             2 Stk x    1,29
ZWIEBEL ROT                        1,50 B
             2 Stk x    0,75
RISPENTOMATE                       1,75 B
         0,704 kg x   2,49 EUR/kg
PRINZESSBOHNEN                     2,49 B
EIER BH M-L                        1,99 B
ESL MILCH 1,5%                     0,99 B
 ---------------------------------------
 SUMME                    EUR     21,79
 =======================================
 Geg. EC-Cash             EUR     21,79

            * *  Kundenbeleg  * *
Datum:                           18.03.2025
Uhrzeit:                       14:21:05 Uhr
Beleg-Nr.                              6480
Trace-Nr.                            707038
              Kartenzahlung
               Contactless
                girocard
```

Figure 19: Original input image (Supermarket Bill)

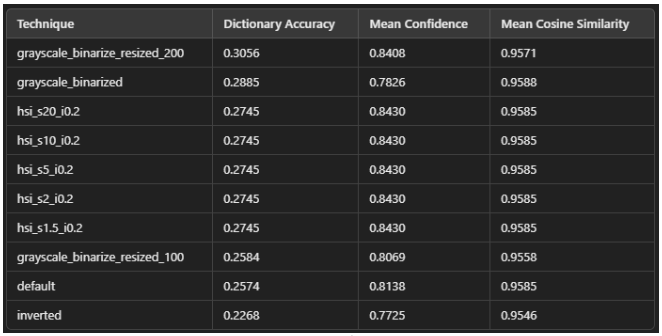


```
                  R E W E
     ****** Am Dachsberg 204 *******
   ****** 60435 Ffm-Preungesheim ******
             Tel. 069/9540430

Hr Günaydin und sein Team bedankt sich
            für Ihren Einkauf
          UID Nr.: DE812706034
                                    EUR
HA-BRUSTFILET                      6,49 B
VOLLKORNTOAST                      0,99 B
LIMETTE                            1,49 B
BANANE BANDEROLE                   0,74 B
         0,576 kg x   1,29 EUR/kg
KIWI                               0,78 B
             2 Stk x    0,39
AVOCADO BIO                        2,58 B
             2 Stk x    1,29
ZWIEBEL ROT                        1,50 B
             2 Stk x    0,75
RISPENTOMATE                       1,75 B
         0,704 kg x   2,49 EUR/kg
PRINZESSBOHNEN                     2,49 B
EIER BH M-L                        1,99 B
ESL MILCH 1,5%                     0,99 B
 ---------------------------------------
 SUMME                    EUR     21,79
 =======================================
 Geg. EC-Cash             EUR     21,79

            * *  Kundenbeleg  * *
Datum:                           18.03.2025
Uhrzeit:                       14:21:05 Uhr
Beleg-Nr.                              6480
Trace-Nr.                            707038
              Kartenzahlung
               Contactless
                girocard
```

Figure 20: Processed Image

Performance of Preprocessing Techniques- Figure 20 refers to the modified version of the original input image after applying preprocessing techniques.

The table in Figure 21 presents a detailed comparison of the top methods as described in Implementation Section IV [D]. The grayscale_binarize_resized_200 method demonstrates the best overall performance, achieving a Mean Cosine Similarity of 0.9571, a Dictionary Accuracy of 0.3056, and a Mean Confidence Score of 0.8408. This suggests that converting the image to grayscale, binarizing, and resizing to 200 DPI yields the most accurate text extraction for the input image.



Technique	Dictionary Accuracy	Mean Confidence	Mean Cosine Similarity
grayscale_binarize_resized_200	0.3056	0.8408	0.9571
grayscale_binarized	0.2885	0.7826	0.9588
hsi_s20_i0.2	0.2745	0.8430	0.9585
hsi_s10_i0.2	0.2745	0.8430	0.9585
hsi_s5_i0.2	0.2745	0.8430	0.9585
hsi_s2_i0.2	0.2745	0.8430	0.9585
hsi_s1.5_i0.2	0.2745	0.8430	0.9585
grayscale_binarize_resized_100	0.2584	0.8069	0.9558
default	0.2574	0.8138	0.9585
inverted	0.2268	0.7725	0.9546

Figure 21: Comparison of the top techniques

The graph in Figure 22 compares the top techniques against their Mean Cosine Similarity, Dictionary Accuracy, and Mean Confidence Score. The grayscale_binarize_resized_200 method has the highest Mean Cosine Similarity and Mean Confidence Score, illustrating how well it performs for extracting text from bills.

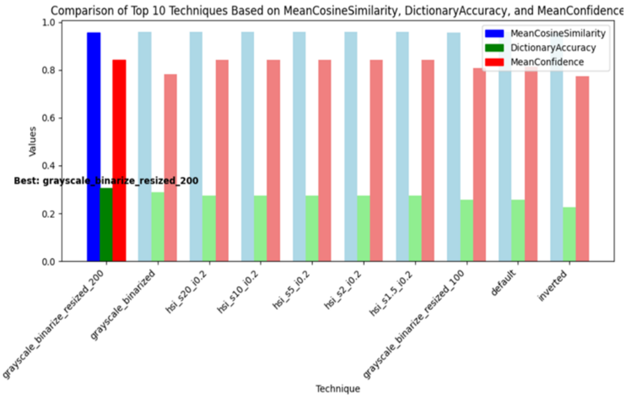
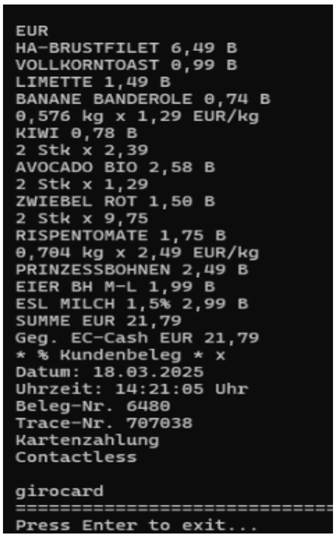


Figure 22: Visual representation of the Performance of preprocessing techniques

Final Output- The console output in Figure 23 shows that the best method is grayscale_binarize_resized_200.The output confirms that the system extracts text correctly with high accuracy, as indicated by the Mean Cosine Similarity of 0.9571, Dictionary Accuracy of 0.3056, and Mean Confidence Score of 0.8408. This confirms that the preprocessing technique of grayscale_binarize_resized_200 effectively enhances OCR performance on structured documents like bills.



```
EUR
HA-BRUSTFILET 6,49 B
VOLLKORNTOAST 0,99 B
LIMETTE 1,49 B
BANANE BANDEROLE 0,74 B
0,576 kg x 1,29 EUR/kg
KIWI 0,78 B
2 Stk x 2,39
AVOCADO BIO 2,58 B
2 Stk x 1,29
ZWIEBEL ROT 1,50 B
2 Stk x 9,75
RISPENTOMATE 1,75 B
0,704 kg x 2,49 EUR/kg
PRINZESSBOHNEN 2,49 B
EIER BH M-L 1,99 B
ESL MILCH 1,5% 2,99 B
SUMME EUR 21,79
Geg. EC-Cash EUR 21,79
* % Kundenbeleg * x
Datum: 18.03.2025
Uhrzeit: 14:21:05 Uhr
Beleg-Nr. 6480
Trace-Nr. 707038
Kartenzahlung
Contactless

girocard
==============================
Press Enter to exit...
```

Figure 23: Console Output (Extracted Text)

E. Histogram

Input Image- The original input image, shown in Figure 24, contains the text that was processed by the system.

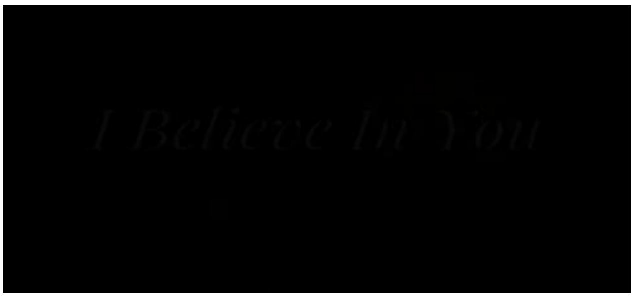


Figure 24: Original Input Image (Histogram)

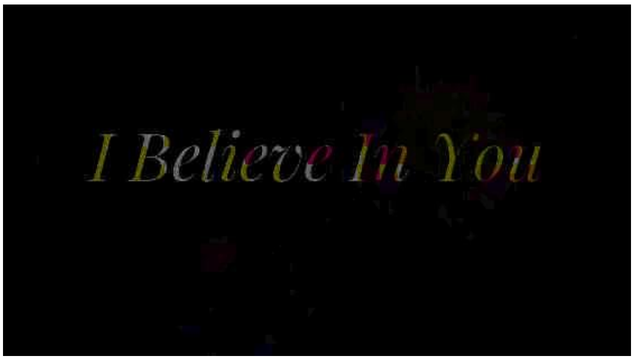
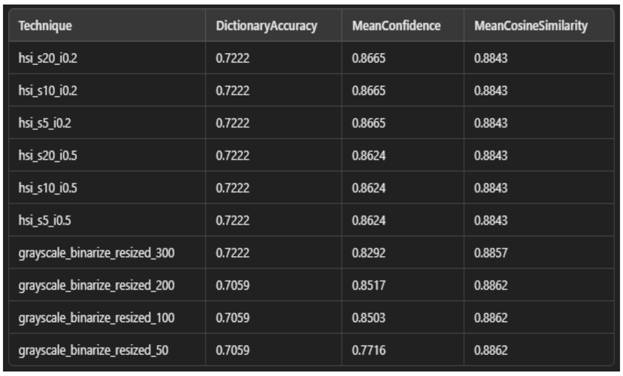


Figure 25: Processed image

Performance of Preprocessing Techniques- Figure 25 refers to the modified version of the original input image after applying preprocessing techniques.

The table in Figure 26 depicts a comparative analysis of the top preprocessing techniques as explained in the Implementation Section IV [D]. The hsi_s20_i0.2 technique has the best overall performance, with the Dictionary Accuracy being 0.7222, Mean Confidence being 0.8665, and Mean Cosine Similarity being 0.8843. This indicates that HSI-based preprocessing offers the most accurate and consistent text extraction, and extreme resizing reduces the mean confidence.



Technique	DictionaryAccuracy	MeanConfidence	MeanCosineSimilarity
hsi_s20_i0.2	0.7222	0.8665	0.8843
hsi_s10_i0.2	0.7222	0.8665	0.8843
hsi_s5_i0.2	0.7222	0.8665	0.8843
hsi_s20_i0.5	0.7222	0.8624	0.8843
hsi_s10_i0.5	0.7222	0.8624	0.8843
hsi_s5_i0.5	0.7222	0.8624	0.8843
grayscale_binarize_resized_300	0.7222	0.8292	0.8857
grayscale_binarize_resized_200	0.7059	0.8517	0.8862
grayscale_binarize_resized_100	0.7059	0.8503	0.8862
grayscale_binarize_resized_50	0.7059	0.7716	0.8862

Figure 26**:** Comparison of the top techniques

The best 10 preprocessing techniques are compared using the Mean Cosine Similarity, Dictionary Accuracy, and Mean Confidence Score in the graph in Figure 27. Hsi_s20_i10 is the highest-performing technique, with the highest value on all three scores, making it the most efficient to utilize to improve OCR accuracy. HSI-based techniques are always excellent, having high similarity and accuracy, and the technique grayscale_binarized shows the least confidence, indicating the lower reliability of this technique for OCR processes. The above discussion confirms that preprocessing HSI techniques are best for extracting correct and high-confidence text from images.

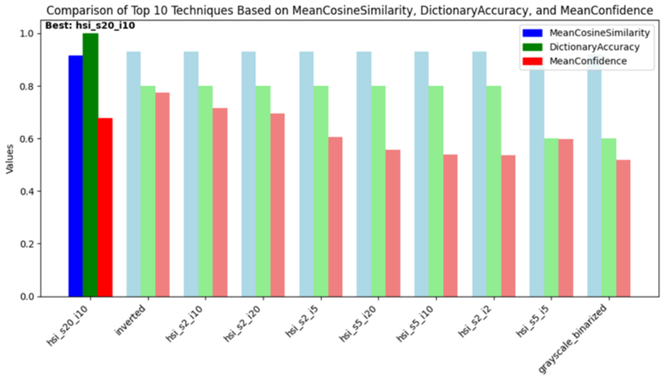
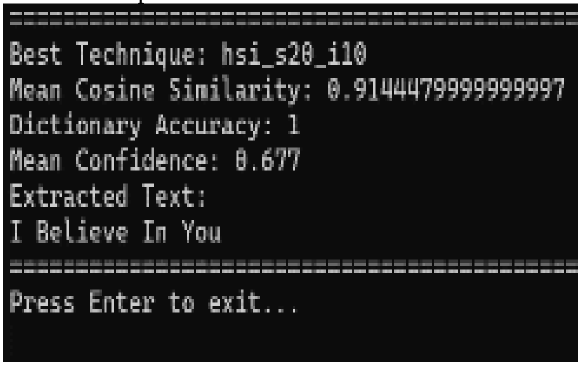


Figure 27: Visual representation of the Performance of preprocessing techniques

Final Output- The console output of Figure 28 displays the final results of the best-performing approach (hsi_s20_110). The output confirms that the system is able to accurately extract the text with high precision, evidenced by the Mean Cosine Similarity of 0.9146, Dictionary Accuracy of 1.0000, and Mean Confidence Score of 0.6770. This confirms that the HSI adjustment preprocessing approach effectively enhances OCR performance.



```
==================================================
Best Technique: hsi_s20_i10
Mean Cosine Similarity: 0.9144479999999997
Dictionary Accuracy: 1
Mean Confidence: 0.677
Extracted Text:
I Believe In You
==================================================
Press Enter to exit...
```

Figure 28: Console Output (Extracted Text)

F. Mirror Image

The original input image, shown in Figure 29, contains the text that was processed by the system:

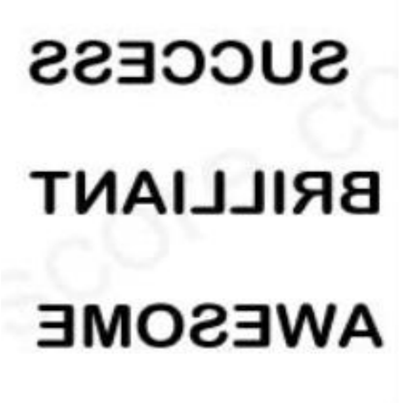


Figure 29: Original Input Image (Horizontal mirror image)

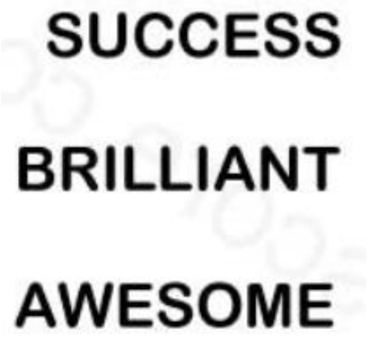
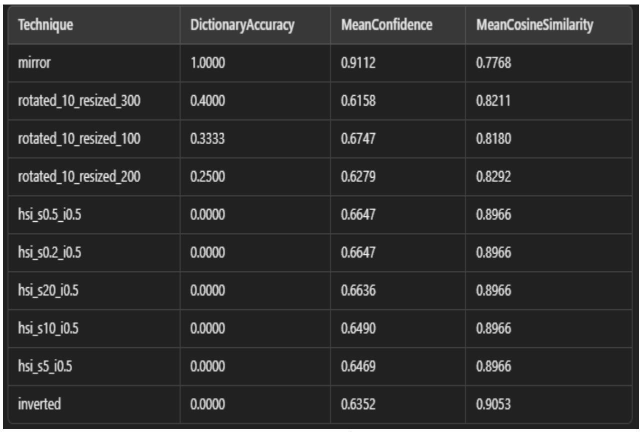


Figure 30: Processed Image

Performance of Preprocessing Techniques- Figure 30 refers to the modified version of the original input image after applying preprocessing techniques.

The comparison of the best methods can be found in the table in Figure 31. Where the top N=10 techniques are selected based on Mean cosine similarity, then they are sorted first by dictionary accuracy and later by mean confidence. Hence, with a Dictionary Accuracy of 1.0000, Mean Confidence Score of 0.9112, and Mean Cosine Similarity of 0.7768, the mirror technique performs best overall.

This suggests that for this input, the mirror image yields the most accurate text extraction results.



Technique	DictionaryAccuracy	MeanConfidence	MeanCosineSimilarity
mirror	1.0000	0.9112	0.7768
rotated_10_resized_300	0.4000	0.6158	0.8211
rotated_10_resized_100	0.3333	0.6747	0.8180
rotated_10_resized_200	0.2500	0.6279	0.8292
hsi_s0.5_i0.5	0.0000	0.6647	0.8966
hsi_s0.2_i0.5	0.0000	0.6647	0.8966
hsi_s20_i0.5	0.0000	0.6636	0.8966
hsi_s10_i0.5	0.0000	0.6490	0.8966
hsi_s5_i0.5	0.0000	0.6469	0.8966
inverted	0.0000	0.6352	0.9053

Figure 31: Comparison of the top techniques

The top ten methods are compared in the graph in Figure 32 according to their Mean Confidence Score, Dictionary Accuracy, and Mean Cosine Similarity. The mirror technique is the most successful in improving OCR performance, as evidenced by its highest Dictionary Accuracy and Mean Confidence Score.

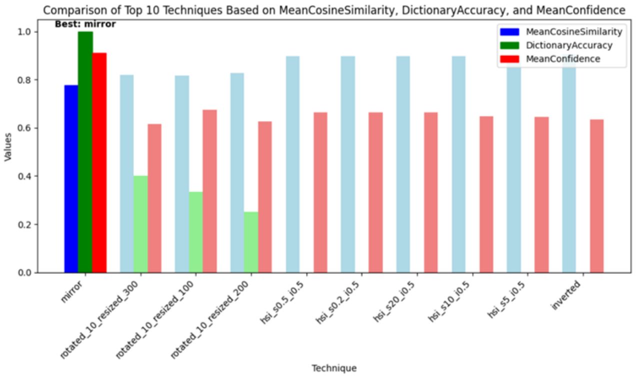
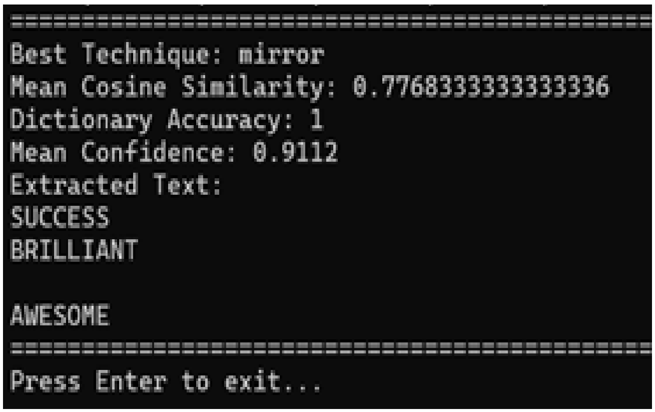


Figure 32: Visual representation of the Performance of preprocessing techniques

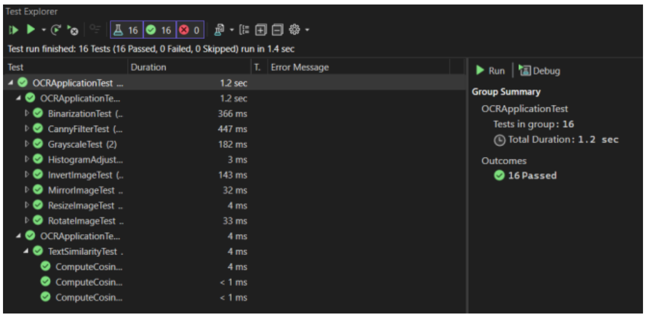
The final output of the best-performing method mirror is shown in the console output in Figure 33. With a Dictionary Accuracy of 1.0000, a Mean Confidence Score of 0.9112, and a Mean Cosine Similarity of 0.7768, the output shows that the system successfully extracts the text accurately. This demonstrates that the mirror preprocessing method effectively enhances OCR performance for this input image.



```
==============================================
Best Technique: mirror
Mean Cosine Similarity: 0.7768333333333336
Dictionary Accuracy: 1
Mean Confidence: 0.9112
Extracted Text:
SUCCESS
BRILLIANT

AWESOME
==============================================
Press Enter to exit...
```

Figure 33: Console Output (Extracted Text)

Unit Testing: The unit tests validate the functionality of the preprocessing and OCR components under various conditions. For the preprocessing techniques, tests cover cases such as image rotation, resizing, grayscale conversion, binarization, and mirroring. Similarly, for the OCR and text similarity modules, tests take into account various text alignments, noise, and image resolutions. The tests ensure that each component is working as expected, producing correct and consistent results on diverse input data. Figure 34 shows the unit testing results, where all 19 tests were executed successfully, demonstrating the system's strength and correctness.



```
Test Explorer
16  16  0
Test run finished: 16 Tests (16 Passed, 0 Failed, 0 Skipped) run in 1.4 sec

Test                          Duration   T.  Error Message
◢ ✓ OCRApplicationTest ...     1.2 sec
  ◢ ✓ OCRApplicationTe...      1.2 sec
    ▷ ✓ BinarizationTest (..    366 ms
    ▷ ✓ CannyFilterTest (...    447 ms
    ▷ ✓ GrayscaleTest (2)       182 ms
    ▷ ✓ HistogramAdjust...        3 ms
    ▷ ✓ InvertImageTest (..     143 ms
    ▷ ✓ MirrorImageTest ..       32 ms
    ▷ ✓ ResizeImageTest ..        4 ms
    ▷ ✓ RotateImageTest ..       33 ms
  ◢ ✓ OCRApplicationTe...        4 ms
    ◢ ✓ TextSimilarityTest .      4 ms
        ✓ ComputeCosin...         4 ms
        ✓ ComputeCosin...       < 1 ms
        ✓ ComputeCosin...       < 1 ms

Group Summary
  OCRApplicationTest
    Tests in group: 16
    Total Duration: 1.2 sec
  Outcomes
    ✓ 16 Passed
```

Figure 34: Unit Testing

VI. CONCLUSION

The text extraction system is a significant improvement on traditional OCR approaches by the addition of advanced preprocessing techniques, strong OCR capabilities, and cosine similarity evaluation. The transition from simple preprocessing techniques to systematic incorporation of techniques such as rotation, resizing, HSI correction, and mirroring considerably enhanced the precision and reliability of text extraction. The application of cosine similarity and dictionary precision metrics ensures that the extracted text is accurate. The system's ability to dynamically select the best preprocessing process based on measures of performance enhances computing capacity and maximizes efficiency. Unit tests assure robustness in the system by testing all the components to perform exactly as expected under different conditions. Overall, this system is significantly improved for text extraction accuracy, efficiency, and reliability, and proves to be a useful tool for high-quality OCR performance applications.

In the future, it is possible to extend this methodology to handwritten text recognition, where there are further challenges due to differences in writing styles, distortions, and inconsistencies in letter formation. Furthermore, extending the OCR capability beyond English to support several languages, including those with complex scripts, will significantly increase its usability. This can be achieved by training the OCR model with multilingual datasets and including language-dependent preprocessing techniques. These advances will make the system more versatile, covering an even broader range of real-world OCR applications.

VII. REFERENCES

[1] S. L. a. H. T. R. Avyodri, "Optical Character Recognition (OCR) for Text Recognition and its Post-Processing Method: A Literature Review," in *1st International Conference on Technology Innovation and Its Applications (ICTIIA), 2022*, Tangerang, Indonesia, 2022.

[2] R. M. a. A. Garg, "Text extraction using OCR: A Systematic Review,," in *Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2020.

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

[4] N. &. I. Z. &. N. N. Islam, "A Survey on Optical Character Recognition System," *ITB Journal of Information and Communication Technology,* 2016.

[5] C. &. P. A. &. P. D. Patel, "Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study," *International Journal of Computer Applications*, 2012.

[6] G. K. K. Bhavesh Kumar Shukla, "An approach for Skew Detection using Hough Transform," in *International Journal of Computer Applications (0975 – 8887)*, Uttar Pradesh , India, 2016.

[7] M. &. G. R. &. P. M. A. T. Brisinello, "Improving optical character recognition performance for low quality images," in *2021 28th Conference of Open Innovations Association (FRUCT)*, Italy, 2017.

[8] T. M. Breuel, "High Performance Text Recognition Using a Hybrid Convolutional-LSTM Implementation," in *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, 2017.