# VerbalEx: Automating User Data Extraction from Government- Issued ID Cards Using OCR

**Abstract**

Text extraction from image-based documents is a critical process in enabling access to information stored in non-editable formats, such as scanned records, photographs, and digital documents. With the increasing availability of image text in various forms, converting this information into editable and searchable formats enhances efficiency in tasks like archiving, data retrieval, and content management. However, challenges arise due to variations in text size, orientation, font, and the presence of noise, poor quality, and complex backgrounds in images. This paper presents a solution that combines advanced OCR algorithms with robust image preprocessing methods to improve text extraction performance.The solution is implemented through an API designed for seamless integration into existing workflows, enabling the efficient extraction of text from image documents. The system’s effectiveness is validated through testing on a variety of datasets, demonstrating its ability to handle different document types with high accuracy.

**Keywords—Text Extraction, OCR, Image Processing, Document Digitization, Accessibility**

# Introduction

In the digital age, the ability to extract text from images has become an essential task across various industries. Image text can be found in numerous forms, such as scanned documents, captured photographs, newspapers, posters, advertisements, and more. These images are widely available and serve as critical sources of information, but accessing the valuable content they contain often remains a challenge. The ability to convert this image-based text into editable and searchable

formats significantly enhances the usability of the information, making it more accessible and easier to manage.

Text extraction from images involves converting text embedded in images—whether from scanned documents, photos, or other visual content—into machine-readable text. This process is crucial for many applications in today’s digital world, including document archiving, content indexing, search engine optimization, and automated data entry. By transforming unstructured image-based data into structured, editable text, it becomes easier to store, retrieve, and process the information, greatly improving productivity and efficiency.

**Text extraction technology has vast applications, including:**

One prominent application is **document digitization**, which addresses the growing demand for converting paper-based records, such as government documents, identification cards (e.g., Aadhar, voter IDs, PAN cards), business contracts, and personal records, into digital formats. This transition enables efficient storage, quick access, and streamlined management of critical information.

The technology enhances **searchability and accessibility** by making extracted text searchable. This capability facilitates the swift retrieval of relevant information from large collections of documents, which is especially beneficial in sectors like legal, healthcare, and education, where timely access to paper records is essential.

Through **content indexing**, text extraction enables categorization, archiving, and retrieval of documents by indexing the extracted text. This application is instrumental in creating digital libraries, document management systems, and searchable databases, thus improving information organization.

The integration of text extraction into automated systems fosters **automation and operational efficiency**, eliminating the need for manual data entry. This not only reduces labor costs and human error but also enhances accuracy, particularly in industries like finance and healthcare, where precise data entry is critical.

The technology supports **globalization and multilingual processing** by recognizing and processing text in various scripts and languages. This adaptability is essential for businesses operating in a globalized environment, allowing them to manage and analyze multilingual documents effectively.

Despite its importance, text extraction from images is not without challenges. Factors such as varying text size, font types, background noise, low- quality images, and complex layouts can complicate the process. Traditional OCR systems, which are typically optimized for clear, high-quality printed text, may struggle when faced with degraded or complex documents. Therefore, improving the accuracy and robustness of text extraction technologies is essential for handling a wide range of document types and ensuring high-quality results.

To address these challenges, we propose the development of a robust text extraction API that can be seamlessly integrated into existing systems and workflows. This API will leverage advanced OCR algorithms and image processing techniques to extract text from a variety of document types with high accuracy. It will support multiple document formats and handle complex image conditions such as low resolution, noise, and distortion.

The API will provide an easy-to-use interface for developers and organizations to integrate text extraction capabilities into their platforms, enabling efficient digitization and management of documents. Whether for automating data entry, enhancing document indexing, or improving accessibility, this solution aims to streamline operations and enhance information retrieval across industries.

This paper will explore the methodologies behind developing this text extraction API, the challenges involved, and the expected outcomes of implementing such a solution in real-world applications. Ultimately, the goal is to provide a scalable and reliable tool that can transform image- based content into structured, searchable text, facilitating more efficient document management and access in the digital age.

# Related Work

Many researchers have done their work on extraction of image text and retrieving the information though there are many challenges. These researches are based on different image text detection and extracting techniques which have their own advantages as well as limitations. Review of these literatures is given by the following table in summarized way.

Najwa-Maria Chidiac, Pascal Damien, and Charles Yaacoub [11] developed a method utilizing MSER (Maximally Stable Extremal Regions) and Stroke Width Detectors to detect and extract text from natural scene images, regardless of orientation. Their approach demonstrated enhanced accuracy in handling blurred and noisy images. However, the system faced limitations when dealing with texts that were small in size, had thin strokes, or were affected by shadow effects.

**Rashedul Islam, Md. Rafiqul Islam, and Kamrul Hasan Talukder [12]** introduced a hybrid approach that combines **Edge-Based and Connected Component-Based techniques** to enhance the accuracy of text area detection and extraction. By merging these methods, the algorithm achieved an accuracy improvement of **87.25%.** However, their evaluation was based on just 8 images, and the study did not account for degraded images, small-sized text, or the use of OCR for character recognition. In their future work, they aim to create a database to facilitate training for the system.

**Arvind and Mohamed Rafi [2]** employed a connected component-based approach to enhance the detection and extraction of text from images, while also categorizing the detected text. Their method aimed to improve performance, achieving a precision rate of **65.06%** and a recall rate of 89.25%. They presented their results visually in the form of graphs to highlight the effectiveness of their approach.

**Harpreet Singh and Deepinder Singh [14]** applied mathematical morphology for text extraction from images, achieving improved performance and reduced noise in the process. However, their method struggled with detecting small text against complex backgrounds. The paper suggests that future work will focus on enhancing the system's ability to extract small text and convert it into an editable format.

Niti Syal and Naresh Kumar Garg [7] presented a method that integrates Daubechies Discrete Wavelet Transform (DWT), Gradient Difference, and Support Vector Machines (SVM) to effectively extract text regions from images. Their approach improved the accuracy of text extraction. The authors proposed future work that includes implementing an OCR system for text recognition and exploring more advanced techniques for removing non-text elements from images.

# METHODOLOGY

To achieve the general and specific objectives of this research, various methods and tools will be utilized. These are outlined as follows:

### Literature Review:-

In order to gain a comprehensive understanding of text extraction from images, a thorough review of existing literature on different aspects of this research has been conducted. This includes an exploration of the fundamental concepts, applications, methodologies, and algorithms used in text extraction from image-based documents, as developed and analyzed by various experts and researchers. The review covers the characteristics of text that may influence the extraction process and examines the impact these characteristics have on the effectiveness of text extraction techniques.

Furthermore, an in-depth study of various text extraction methods has been carried out. Among the most widely used techniques are the Connected Component-Based Method, Edge-Based Method, Region-Based Method, Mathematical Morphology- Based Method, and Digital Image Processing. These methods have been explored in detail in several studies [4, 16], and their respective advantages and limitations have been critically analyzed.

**Methods**

The methodology for this dissertation is organized into three main stages, which correspond to different phases of the text extraction process. These stages are: **Pre-processing**, **Processing**, and **Post- processing**. Each stage plays a critical role in ensuring that the text extraction is performed accurately and efficiently.

*A. Pre-processing Stage*

The Pre-processing stage addresses common issues in image documents, such as noise, blurring, and uneven lighting, which can significantly hinder the detection, extraction, and recognition of text. The aim of this stage is to enhance the quality of the image and prepare it for the subsequent processing stages.

In the pre-processing phase, the input image, whether it is scanned, captured from a device, or retrieved from a storage source, undergoes a series of transformations to improve its clarity and remove

any undesirable artifacts. The first step in this process involves removing any noise that may have been introduced during image acquisition or transmission. Noise reduction is essential for minimizing errors during text extraction, as noise can obscure or distort the text content.

Once the image is cleared of noise, the next step is to convert the color (RGB) image into grayscale. This conversion simplifies the image by removing the color dimension, which is not necessary for text extraction. Grayscale images are easier to process and require less computational power, as the focus is solely on the intensity of light rather than color.

After the conversion to grayscale, a thresholding technique is applied to further prepare the image for extraction. Thresholding involves converting the grayscale image into a binary format, where the pixels are classified into two categories— foreground (text) and background—based on a defined threshold. This step simplifies the extraction process by creating a high-contrast image, making the text more distinguishable from the background.

By addressing these issues, the pre-processing stage enhances the quality of the input image, making it more suitable for text detection and extraction. The result is a cleaner and more defined image that facilitates higher accuracy in subsequent processing and recognition stages.



*B.Processing Stage*

The **Processing Stage** is essential for determining whether an image contains text and for identifying the location of text within the image. This stage involves a series of steps aimed at distinguishing the foreground (text) from the background and organizing the text regions for extraction.

The first step in this stage is **Text Detection**, where the enhanced image is analyzed to determine if it contains any text. If text is present, the system identifies the regions of the image where the text is located.

Following detection, **Text Localization** is performed, which involves grouping the detected text regions to form coherent text objects. Tight boundaries are then defined around these text objects to isolate them from the rest of the image. This helps in localizing the text effectively for further processing. The architecture of the processing stage involves closely related modules for text detection, localization, and tracking.

**Text Tracking** is another important step that improves the efficiency of the text extraction process. By tracking the detected text objects, the system can avoid repeatedly applying binarization and recognition steps to each individual text object, thus speeding up the overall extraction process.

The **Text Binarization** step is used to clearly separate the text from the background. It converts the grayscale image into a binary image where the text pixels and the background pixels are represented by distinct binary values—typically, white text on a dark background or vice versa. This step can also be performed before the other stages to enhance the contrast and make text extraction more efficient.

Overall, the processing stage plays a critical role in ensuring that the text is effectively detected, localized, and prepared for accurate extraction in the subsequent phases.

### C.Post-Processing Stage

The **Post-Processing Stage** focuses on the final steps that are applied to the extracted text to enhance its usability. These processes typically include segmentation, character recognition, and text-to- speech conversion.

The first step in this stage is **Segmentation**, which involves dividing the image into meaningful regions to facilitate easier text extraction. Image segmentation is commonly divided into two types: for character segmentation, the image is initially segmented row-wise (line segmentation) to separate the text into lines. Following that, the individual lines are segmented column-wise (word segmentation) to identify distinct words within each line. This step is critical for breaking down the text into manageable units for recognition [6].

Once the image has been segmented, the next step is **Character Recognition**, where the binary text object is processed to convert the segmented text into machine-readable characters. This is typically achieved using an OCR (Optical Character Recognition) tool, which translates the detected text into ASCII characters. As shown in Figure 4, the OCR system is capable of recognizing words, such as "PARUL UNIVERSITY," from the image and converting them into a text format.

Finally, the **Text to Speech (TTS)** process is applied to convert the recognized text into audible speech. This allows the system to output the recognized characters or words as spoken language, enhancing accessibility for users who may prefer or require audio output.

In summary, the post-processing stage focuses on refining the extracted text, converting it into usable formats, and adding functionality such as speech output to make the text more accessible and usable.

## System Architecture for VerbalEx

The architecture of **VerbalEx** is designed to address the complexities involved in automating the extraction of user data from government-issued ID cards. The system employs a modular, layered architecture that leverages modern technologies such as Optical Character Recognition (OCR) and API integration, ensuring a scalable, secure, and efficient solution. The following sections provide a detailed overview of the architecture, structured according to the phases of the Software Development Life Cycle (SDLC).



*1. Requirement Analysis*

The **Requirement Analysis** phase is pivotal in defining the core objectives and functionality of the **VerbalEx** system. In this phase, key stakeholders, including project managers, business analysts, and domain experts, collaborate to gather detailed requirements. The main goal is to understand the needs of the end-users and identify the specific features required for the successful operation of the system.

For **VerbalEx**, the primary functionality includes the extraction of essential user data (such as name, address, and date of birth) from government-issued ID cards using OCR. Additionally, the system must provide an API to facilitate integration with other applications, such as automated form filling, e- signature services, and multilingual support. Other key considerations during this phase include security and privacy, given the sensitive nature of the personal data being processed. Risks, such as OCR inaccuracies and handling diverse card formats, are also identified and mitigated during this stage.

Furthermore, quality assurance criteria and potential risks are outlined to ensure that the system’s performance aligns with the user’s needs. This foundational understanding sets the stage for the design and development phases that follow.

*2..System Design*

The **System Design** phase translates the requirements gathered in the previous phase into a detailed architecture and design specification. The **VerbalEx** system is built on a combination of frontend and backend technologies, as well as machine learning (ML) for OCR processing.

The **frontend** is developed using **React.js**, chosen for its ability to create dynamic, user-friendly interfaces. React.js efficiently handles complex user interactions and real-time updates, enabling users to upload ID card images and view the extracted data effortlessly. Its flexibility ensures a smooth and intuitive user experience.

The **backend** leverages **Node.js** and **Express.js** to deliver high performance and scalability. These technologies manage HTTP requests, user sessions, and database interactions, acting as a bridge between the frontend, OCR service, and database. The backend ensures a seamless flow of data across the system, maintaining stability and responsiveness.

A critical element of the system is **OCR integration**. Using **Tesseract OCR** via Python, the system converts text from images into machine-readable formats. This machine-learning-based tool processes ID card images, extracts text, and returns structured data, enabling efficient downstream use.

To store the extracted data, the system employs **MongoDB** as its database. MongoDB’s capability to handle unstructured and semi-structured data makes it an ideal choice for managing text extracted from various ID card formats. Its scalability ensures robust performance even with large volumes of data.

Finally, the system incorporates **RESTful API design** to expose backend functionalities. These APIs allow third-party applications, such as automated form-filling tools and e-signature services, to access the extracted data. This design not only ensures smooth integration with external systems but also facilitates extensibility for future enhancements.

This phase also defines the system’s scalability requirements and outlines the cloud infrastructure to be used for deployment. The system is designed to accommodate potential future growth in data volume and user load.

*3..Implementation*

The **Implementation** phase involves the actual development of the **VerbalEx** system, wherein the design specifications are translated into code. During this phase, the frontend and backend components are developed concurrently.

The **frontend development** employs **React.js** to create a user-friendly and responsive interface. This interface allows users to upload images of ID cards effortlessly. Designed to be intuitive and compatible across devices, the frontend enables users to view extracted data and interact with the backend through API calls, ensuring a seamless user experience.

The **backend development** utilizes **Node.js** and **Express.js**, providing robust infrastructure for handling HTTP requests, managing user authentication, and integrating the OCR service. The backend also processes the extracted data and facilitates communication with the database, ensuring efficient data flow and system reliability.

**OCR service integration** is achieved through the **Tesseract OCR library**, implemented using Python. This component is fine-tuned to handle the unique challenges posed by ID cards, such as varying fonts, image quality, and complex backgrounds. This optimization ensures high accuracy in text extraction, even under challenging conditions.

For **database integration**, **MongoDB** is used to store the extracted data in a structured format. The database is optimized for scalability, capable of handling large datasets with fast read and write operations. This ensures reliable and efficient data retrieval and storage, even as the system scales.

Throughout this phase, continuous testing is performed to ensure that the system components are functioning correctly and that the overall architecture meets the defined requirements.

*4..Testing*

The **Testing** phase ensures that the **VerbalEx** system operates as intended and meets the requirements defined in the earlier stages. The system undergoes various levels of testing to identify and resolve any issues before deployment.

**Unit Testing**: Individual modules, including the frontend, backend, and OCR service, are tested in isolation to verify that each component functions correctly.

**Integration Testing**: The system components are tested together to ensure smooth communication and data flow between the frontend, backend, OCR service, and database.

**System Testing**: The entire system is tested to ensure that all parts work together seamlessly. The system is subjected to different use cases, such as uploading different types of ID card images, to verify its robustness and accuracy.

**User Acceptance Testing (UAT)**: End- users conduct testing to ensure that the system meets their needs and expectations. Feedback from this phase is used to make any final adjustments before deployment.

## Summary

The architecture of **VerbalEx** is designed with a focus on scalability, efficiency, and security. By adhering to a structured SDLC approach, from **Requirement Analysis** to **Maintenance**, the system provides a robust solution for automating the extraction of user data from ID cards. The modular architecture, which integrates OCR for text extraction, a flexible backend for API access, and a secure database for storing user information, ensures that **VerbalEx** meets both current and future needs. The architecture’s flexibility also allows for easy enhancements, such as adding support for additional languages or integrating with other systems in the future.

**References**

1.International Journal of Computer Applications Technology and Research Volume 3– Issue 4, 239 - 243, 2014, ISSN: 2319–8656

https://ijcat.com/archives/volume3/issue4/ijcatr03041009.pdf

2.Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Suresh, Tao Wang, David J. Wu, Andrew Y. Ng Computer Science Department Stanford University 353 Serra Mall Stanford, CA 94305 USA.

https://www.cs.utexas.edu/~dwu4/papers/TextRecog.pdf