International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Image Data Extractor

Dr R Dinesh Kumar[1], P.Harika Sravani[2],J.Mythri[3]

[1]Associate Professor; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India.

[2,3]B.Tech Students ; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India.

Mail Id; mythrijampula@gmail.com[3]

ABSTRACT

This project presents a smart and efficient system for document digitization and data entry automation using a combination of Optical Character Recognition (OCR) and deep learning techniques. The proposed system primarily targets the digitization of loan application forms, which are often received in printed or handwritten formats. Using Tesseract OCR, the system extracts text from scanned images, followed by preprocessing techniques such as grayscale conversion and Otsu thresholding to enhance image clarity. To improve accuracy, transformer-based models like BERT or T5 are incorporated for contextual text understanding and refinement. Key fields such as names, dates, and account numbers are identified and structured using Named Entity Recognition (NER) and regex-based validation. The extracted and cleaned data is then exported into Excel files in a fixed format, making it easy to review and integrate with enterprise applications. This solution significantly reduces human effort, eliminates common manual errors, supports real-time processing, and ensures scalability for large-scale document handling.

Keywords— *Optical Character Recognition (OCR), Document Digitization, Deep Learning, Named Entity Recognition (NER), Intelligent Document Processing, Transformer Models, Data Extraction, Loan Application Automation.*

Introduction

minimizing human intervention in data entry tasks. The project will begin with a literature review to understand existing OCR and deep learning-based document processing techniques. The implementation will involve Teserract for image-based text recognition and Transformer-based models (such as BERT or T5) for refining extracted text and ensuring contextual accuracy. Named Entity Recognition (NER) techniques will also be explored to extract key information such as names, dates, and numerical values.

In today's data-driven world, banks deal with vast amounts of physical and digital loan forms. Traditional data entry methods are labor-intensive, time-consuming, and error-prone. To overcome these challenges, a Deep Learning-Based Smart Data Entry System is proposed for document digitization and processing. By leveraging Optical Character Recognition (OCR), this system automates text extraction, validation, and structuring from various document formats. The implementation of deep learning techniques enhances accuracy and efficiency.

Existing System

Traditional data entry primarily involve manual data entry by human operators, which uses more man-power and it is time-consuming process. Since humans are prone to errors, this method often leads to inaccuracies, inconsistencies, and inefficiencies in data management. Additionally, the need for continuous supervision and verification increases operational costs and slows down workflows. As data volumes grow, manual entry becomes unsustainable, making it difficult for processing loan applications efficiently.

Proposed System

The document digitization system efficiently processes scanned images or documents by extracting key information and organizing it in a structured Excel format. It uses Tesseract OCR for accurate text extraction and Transformer-based models for entity recognition, identifying fields such as name, bank number, and other predefined entries. The Excel sheet has a fixed structure, allowing the system to map extracted data directly to corresponding fields.To enhance usability, the system supports bulk uploads and batch processing, enabling organizations to process large volumes of forms efficiently. It can be integrated with existing HR or finance systems for seamless data transfer. Future updates may include support for multilingual OCR, better handwriting recognition, mobile app access, and advanced security features like encryption and user access control. These improvements would expand the system's functionality, making it suitable for enterprise use across various sectors, including banking, education, and government institutions, where structured data extraction from physical documents is essential.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

REQUIREMENT ANALYSIS

Functional Requirements

These are the requirements that refers to the specific actions, behaviors, or tasks a system or application is designed to perform. Functional requirements describe what the system must do to achieve its objectives and typically outline features, inputs, outputs, and interactions

Non-Functional requirements

These are the requirements that refers to the quality attributes or characteristics of a system that do not directly relate to its specific tasks but focus on how the system performs under certain conditions. These requirements address performance, usability, reliability, scalability, and other operational aspects.

- Scalability : Ability to handle a growing number of users.
- Usability : User-friendly interface that is easy to navigate.
- Reliability : Maintain high system availability to prevent critical

DESIGN

Architecture

Project architecture represents number of components we are using as a part of our project and the flow of request processing i.e. what components in processing the request and in which order. An architecture description is a formal description and representation of a system organized in a way that supports reasoning about the structure of the system. Architecture is of two types. They are:

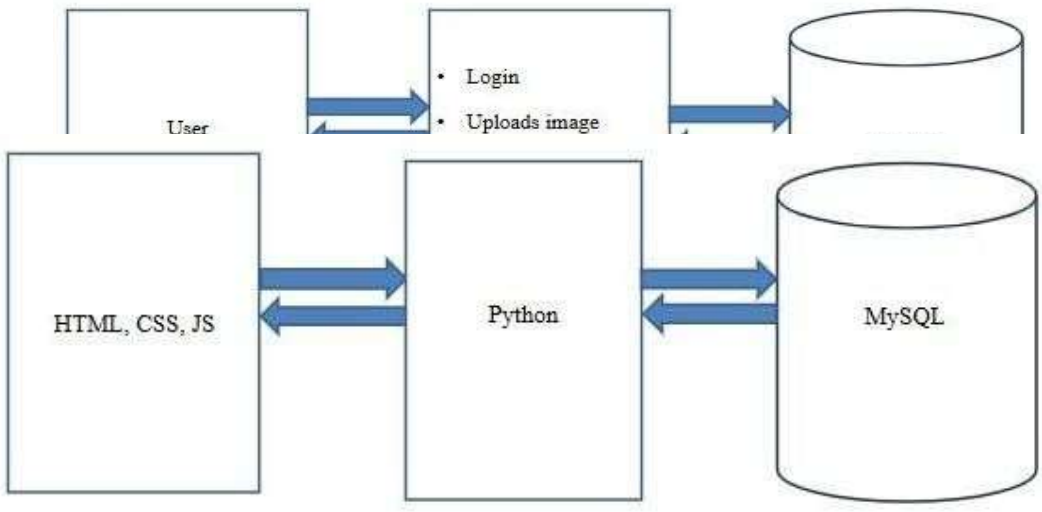
- Software Architecture
- Technical Architecture

Software Architecture

Software architecture design tools help to build software that does not have security issues. This is key because there are software risks in all areas of the software development process.

The fig 1 Software Architecture depicts a user interacting with a system that manages authentication, file uploads, and downloads, with data stored in a MySQL database. User actions—such as login, image uploads, Excel file downloads, and logout—are processed by the system and reflected in the database. The architecture showcases a typical web application flow, ensuring efficient data handling and user management.

Fig. 1 Software Architecture



Technical Architecture

Technical Architecture is a form of IT architecture that is used to design computer systems. It involves the development of a technical blueprint regarding the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

The fig .2 represents the interaction between different technologies in a web application. It showcases three main components: **HTML, CSS, JS** for front-end development, **Python** for back-end processing, and **MySQL** as the database.

Fig 2 Technical Architecture

Software Process Model

We use a software process model to provide a structured approach to software development. It helps plan, design, develop, and maintain software systematically, ensuring quality, efficiency, and timely delivery.

The fig 3 The Waterfall Model is a structured software development process where each phase

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

Feasibility Study, Requirement Analysis, Design, Coding, Testing, and Maintenance

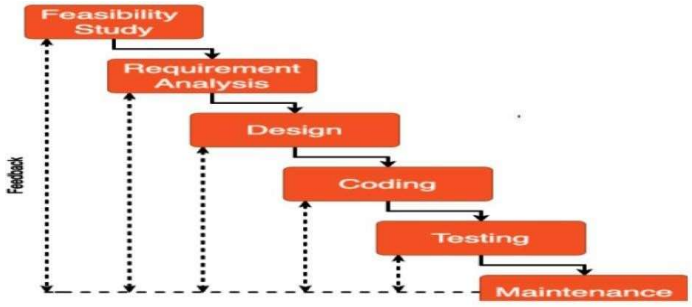


Fig.3Software Process Model

Dataflow Diagram

Data Flow Diagrams (DFDs) are used to visually represent the flow of data within a system.They also improve communication between developers, analysts, and non-technical stakeholders by providing a clear, diagrammatic view of system functionality.

The fig 4 The diagram illustrates the process of converting an uploaded image into an Excel file, involving steps like image processing (grayscale, binarization), text extraction via OCR, and structured data generation using Pandas. It showcases an automated pipeline ensuring accurate text recognition and tabular format conversion.

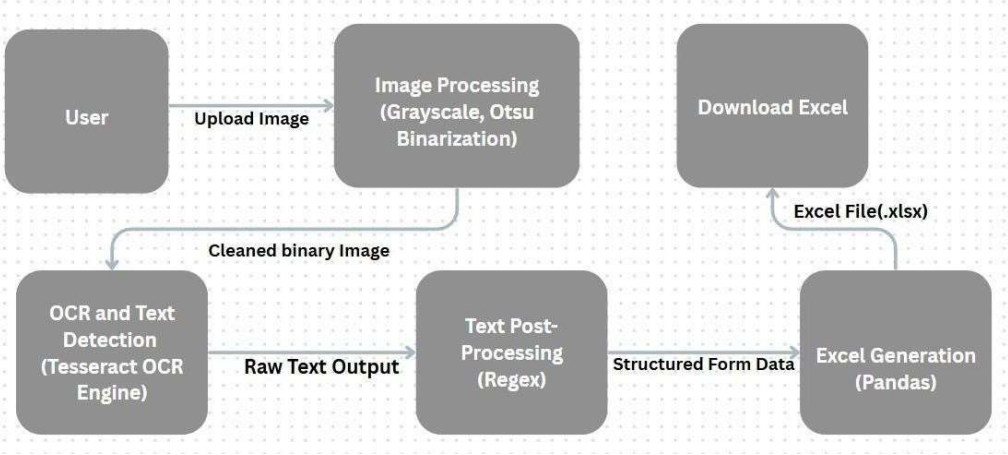


Fig.4 Dataflow Diagram

Algorithm

Grayscale Conversion:

Gray = 0.299 * R + 0.587 * G + 0.114 * B

An image is converted to grayscale so that the image processing will be simple

The fig 5 The image presents three color-coded matrices (red, green, and blue) with numerical values, likely representing a structured 3D data visualization or transformation process.

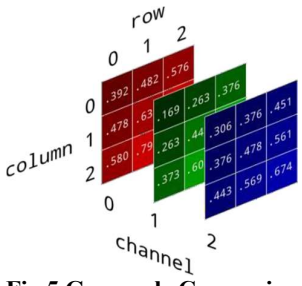


Fig 5 Grayscale Conversion

METHODOLOGY

The Document Digitization project follows a structured development approach starting with requirement analysis and system design. It uses Python, Django, Tesseract OCR, and OpenCV to extract and process text from scanned images. Preprocessing techniques like grayscale conversion and Otsu thresholding enhance OCR accuracy. Extracted data is structured using regex and exported to Excel with Pandas. The system is tested through unit, integration, and user acceptance testing, and deployed as a web app for users to upload documents and download structured Excel data efficiently. It is to accurately extract text from scanned documents using Tesseract OCR and enhance recognition accuracy through image preprocessing techniques like grayscale conversion and Otsu thresholding, followed by structuring the extracted text into key-value pairs for organized storage in Excel format..

Python

Python is one of the most popular programming languages now existing. The main reason for the creation of a programming language like python was to enhance the features to a large extent that were available in the present existing languages. The other reason was to invent a language which can be used easily for the developers who work a lot on media other than texts like speech, images and videos. The other important reason was to increase the built-in functions so as to reduce the number of lines in the codes and implement simplicity. Python is basically created in such a way that the garbage is involuntarily and automatically collected. The Python language can be called as a mixture of all the languages with more features added to it. It is a structured language yet it does not support the use of the semicolons at the end of each operation. Python consists of a very large standard library which consists of a huge number of built-in functions which reduce the developer's load of writing hundreds of lines to perform a single and simple task.

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement. Testing for a Multilevel Data Concealing Technique that integrates Steganography and Visual Cryptography is crucial to ensure its functionality, security, and reliability. The testing process involves several stages, including unit testing, integration testing, and security testing.

Stages of Testing

Unit Testing

During This first round of testing, the program is submitted to assessments that focus on specific units or components of the software to determine whether each one is fully functional. In this phase, a unit can refer to a function, individual program or even a procedure, and White box testing method is usually used to get the job done.One of the biggest benefits of this testing phase is that it can be run every time a piece of code is changed, allowing issues to be resolved as quickly as possible. It quite common for software developers to perform unit tests before delivering software to testers for formal testing.

Integration Testing:

Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is designed to find interface defects between the modules/functions. This is particularly beneficial because it determines how efficiently the units are running together. Keep in mind that no matter how efficiently each unit is running, if they properly integrated, it will affect the functionality of the software program. System Testingtesting is undertaken by independent testers who haven't played a role in developing the program. This testing is performed in an environment that closely mirrors production. System Testing is very important because it verifies that the application meets the technical, functional, and business requirements that were set by the customer.

Acceptance Testing

The final level, Acceptance testing (or User Acceptance Testing), is conducted to determine whether the system is ready for release. During the Software development life cycle, requirements changes can sometimes be misinterpreted in a fashion that does not meet the intended needs of the users. During this final phase, the user will test the system to find out whether the application meets their business needs. Once this process has been completed and the software has passed, the program will then be delivered to production. The extensiveness of these tests is just another reason why bringing software testers in early is important. When a program is more thoroughly tested, a greater number of bugs will be detected; this ultimately results in higher quality software.

Black box testing

It is also called as Behavioral/Specification-Based/Input-Output Testing. Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure. This can be applied to every level of software testing such as Unit, Integration, System and Acceptance Testing.

White box testing

It is also called as Glass Box, Clear Box, Structural

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

Testing. White Box Testing is based on applications internal code structure. In white-box testing an internal perspective of the system, as well as White Box Testing Techniques:

- Statement Coverage

programming skills, are used to design test cases. This testing usually done at the unit level.

- Branch Coverage
- Path Coverage

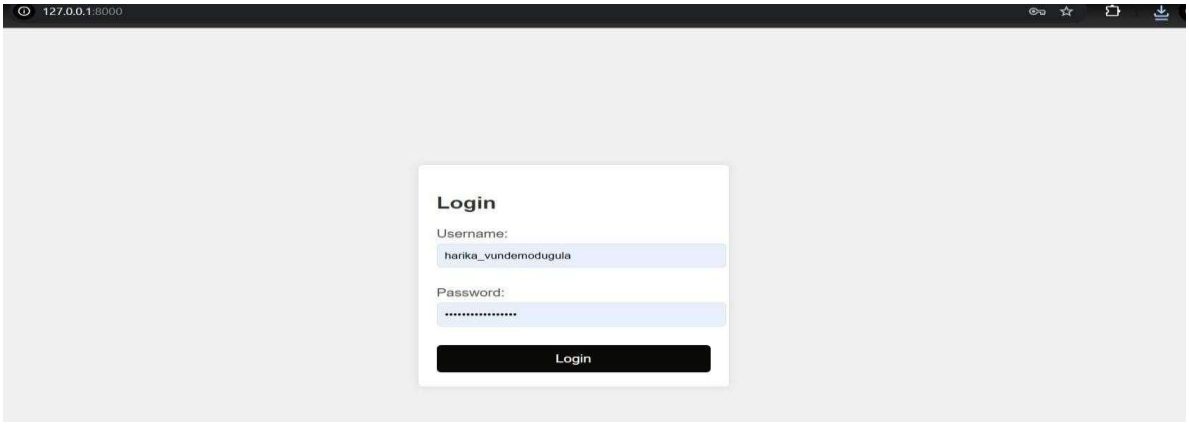
SCREENSHOTS

Run the website

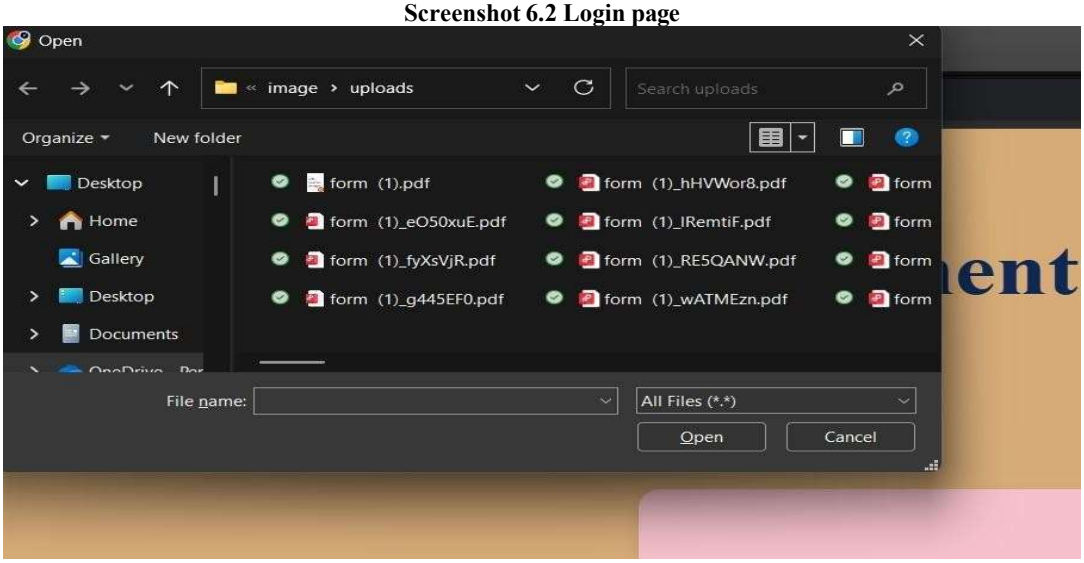
```
C:\Users\harik\OneDrive\Desktop\my_project>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
May 26, 2025 - 21:03:13
Django version 5.1.1, using settings 'my_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Screenshot 6.1 Run the web app from the command prompt



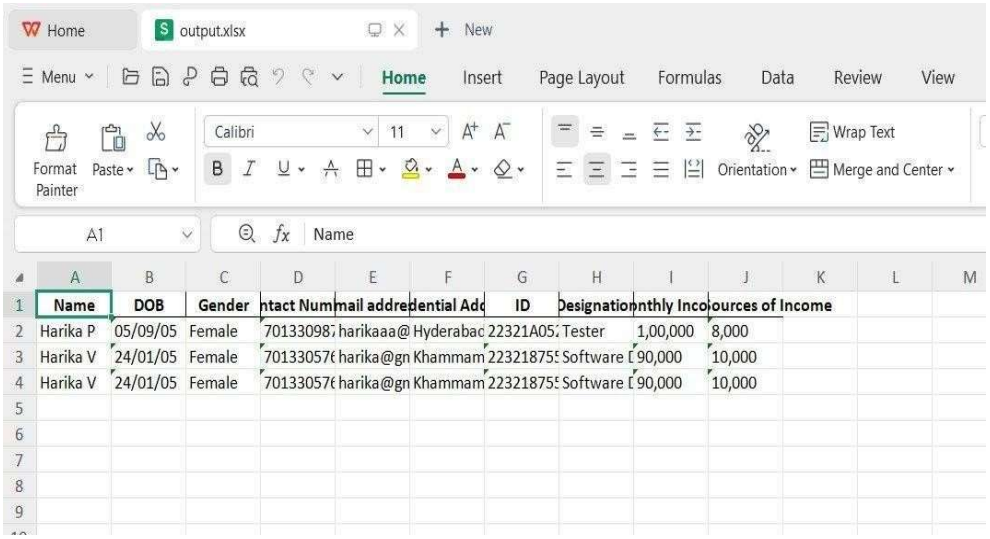
Screenshot 6.2 Login page



Screenshot 6.3 Select file to upload

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

Screenshot 4 Click on Download Excel



Screenshot 5 Image to Excel converted sheet

Conclusion

This project demonstrates an effective and intelligent approach to document digitization using OCR and deep learning techniques. By automating the extraction, validation, and structuring of data from printed and handwritten forms, the system significantly reduces manual workload and human errors. Tools like Tesseract and transformer-based models ensure high accuracy and contextual understanding, while exporting structured data to Excel enhances usability. The solution is scalable, cost-effective, and well-suited for organizations handling large volumes of documents, such as banks. Overall, this system bridges the gap between physical documentation and digital efficiency, improving operational speed and accuracy.

Future Scope

The document digitization system can be enhanced by adding multilingual OCR to support various languages, improving handwriting recognition with deep learning models, and integrating cloud platforms for real-time processing and remote access. Machine learning can enable automatic classification of documents like invoices and forms. A mobile app will allow easy scanning and uploading from smartphones, increasing accessibility. Strengthening security with encryption and access controls will ensure safe handling of sensitive data, making the system suitable for enterprise use. These improvements will broaden usability, boost accuracy, and enhance user

convenience across diverse settings.

REFERENCES

1. **Smith, R. (2022).** An Overview of Tesseract OCR Engine. International Journal of Computer Vision and Image Processing, 14(1), 10-18. Discusses updates and capabilities of Tesseract for image-to-text processing.
2. **Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2023).** BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Journal of Artificial Intelligence Research, 72, 1-18. Explains BERT's use for contextual understanding and Named Entity Recognition (NER).
3. **Vincent, W. S. (2023).** Django for Professionals (4th ed.). WelcomeToCode. A practical guide on building secure, scalable web applications using Django.
4. **Brownlee, J. (2022).** Deep Learning for Computer Vision. Machine Learning Mastery. Covers techniques for image preprocessing and integrating deep learning in OCR tasks.
5. **Chowdhury, S., & Rahman, M. (2022).** Enhancing OCR Accuracy with Image Preprocessing Techniques. Proceedings of the IEEE International Conference on Image Processing (ICIP), 1567–1571.A technical paper evaluating preprocessing methods like Otsu thresholding and grayscale conversion.
6. Sun, Y., Zhou, D., Lin, C., He, C., Ouyang, W., & Zhong, H.-S. (2024). LOCR Location-Guided Transformer for Optical Character Recognition. Findings of ACL: EMNLP 2024, Nov 2024, 5480–5497.
7. Laurent, V. & Lauar, F. (2024). Spanish TrOCR: Leveraging Transfer Learning for Language Adaptation. arXiv preprint arXiv:2407.06950, Jul 2024.
8. Chang, D. & Li, Y. (2024).DLoRA-TrOCR: Mixed Text Mode OCR Based On Transformer. arXiv preprint arXiv:2404.12734, Apr 2024.
9. Khallouli, W., Uddin, M. S., Sousa-Poza, A., Li, J., & Kovacic, S. (2025).Leveraging Transformer-Based OCR Model with Generative Data Augmentation for Engineering Document Recognition.
10. Hamdi, L., Tamasna, A., Boisson, P., & Paquet, T. (2025).VISTA-OCR: Towards Generative and Interactive End-to-End OCR Models.arXiv preprint arXiv:2504.03621, Apr 2025.