



**TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS**

**A Project Report
On
Marksheet Digitization using OCR and Report Generation for QAA**

Submitted By:

| | |
|-----------------|----------------|
| Anish Timsina | (THA077BEI007) |
| Bishal Giri | (THA077BEI014) |
| Sugam Khatiwada | (THA077BEI046) |

Submitted To:

Department of Electronics and Computer Engineering,
Thapathali Campus
Kathmandu, Nepal

March 2025



**TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS**

A Project Report

On

Marksheet Digitization using OCR and Report Generation for QAA

Submitted By:

| | |
|-----------------|----------------|
| Anish Timsina | (THA077BEI007) |
| Bishal Giri | (THA077BEI014) |
| Sugam Khatiwada | (THA077BEI046) |

Submitted To:

Department of Electronics and Computer Engineering,
Thapathali Campus
Kathmandu, Nepal

In Partial fulfillment for the award of the Bachelor's Degree in Electronics,
Communication and Information Engineering.

Under the Supervision of

Associate Professor Er. Shanta Maharjan

March 2025

DECLARATION

We hereby declare that the report of the project entitled “**Marksheet Digitization using OCR and Report Generation for QAA**” which is being submitted to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, in the partial fulfillment of the requirements for the award of the **Degree of Bachelor of Engineering in Electronics and Communication Engineering**, is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work.

Anish Timsina (THA077BEI007) _____

Bishal Giri (THA077BEI014) _____

Sugam Khatiwada (THA077BEI046) _____

Date: March 2025

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to **the Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, a minor project work entitled “**Marksheet Digitization using OCR and Report Generation for QAA**” submitted by **Anish Timsina, Bishal Giri and Sugam Khatiwada** in partial fulfillment for the award of Bachelor’s Degree in Electronics and Communication Engineering. The Project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of Bachelor’s degree of Electronics and Communication Engineering.

Project Supervisor

Associate Professor Er. Shanta Maharjan
EMIS Head, Thapathali Campus

External Examiner

Project Co-ordinator

Mr. Sudip Rana
Department of Electronics and Computer Engineering, Thapathali Campus

Mr. Umesh Kanta Ghimire

Head of the Department,
Department of Electronics and Computer Engineering, Thapathali Campus

March, 2025

COPYRIGHT

The author has agreed that the library, Department of Electronics and Computer Engineering, Thapathali Campus, may make this report freely available for inspection. Moreover, the author has agreed that the permission for extensive copying of this project work for scholarly purpose may be granted by the professor/lecturer, who supervised the project work recorded herein or, in their absence, by the head of the department. It is understood that the recognition will be given to the author of this report and to the Department of Electronics and Computer Engineering, IOE, Thapathali Campus in any use of the material of this report. Copying of publication or other use of this report for financial gain without approval of the Department of Electronics and Computer Engineering, IOE, Thapathali Campus and author's written permission is prohibited. Request for permission to copy or to make any use of the material in this project in whole or part should be addressed to department of Electronics and Computer Engineering, IOE, Thapathali Campus.

ACKNOWLEDGEMENT

We would like to thank the **Institute of Engineering, Tribhuvan University** for providing us with such a wonderful opportunity to learn and implement our knowledge in the form of a major project.

We appreciate the dedication of the **Department of Electronics and Computer Engineering, Thapathali Campus** and express our gratitude for providing us with an excellent platform and the opportunity to apply and enhance our skills. We are also very thankful to our supervisor **Associate Professor Er. Shanta Maharjan** for his support and guidance.

In addition, we would like to thank all teaching as well as non-teaching staff and classmates for their useful advice and support.

Sincerely,

Anish Timsina (THA077BEI007)

Bishal Giri (THA077BEI014)

Sugam Khatiwada (THA077BEI046)

ABSTRACT

“Marksheet Digitization using OCR and Report Generation for QAA” focuses on conversion of physical exam marksheets into digital format with the help of Optical Character Recognition (OCR) technology along with Quality Assurance and Accreditation (QAA) for enhanced report generation. The main aim of this project is to convert scanned or physical academic records into an organized digital format ensuring enhanced accuracy, accessibility and data management. This project begins with high-resolution scanning of physical marksheets, followed by preprocessing steps to refine image clarity. OCR technology then extracts the text data from the marksheet which is further validated and structured into a database and integrated with reporting tools to analyze and visualize the digitized data. To further enhance accessibility and usability, a dedicated website is developed, providing secure, scalable, and convenient access to digitized marksheets and reports for students, teachers, and administrators. The system not only reduces manual effort and errors but also provides a robust platform for managing large volumes of student data efficiently.

Keywords: Digitization, Marksheets, OCR, QAA, Report, Secure

Table of Contents

| | |
|--------------------------------------|-------------|
| DECLARATION..... | i |
| CERTIFICATE OF APPROVAL | ii |
| COPYRIGHT..... | iii |
| ACKNOWLEDGEMENT..... | iv |
| ABSTRACT | v |
| List of Tables | ix |
| List of Equations | xii |
| List of Abbreviations | xiii |
| 1. INTRODUCTION | 1 |
| 1.1 Background Information..... | 1 |
| 1.2 Motivation..... | 2 |
| 1.3 Problem Statement | 2 |
| 1.4 Objectives | 3 |
| 1.5 Scope and applications..... | 3 |
| 1.6 Project Limitation | 3 |
| 1.7 Report Organization..... | 4 |
| 2. LITERATURE REVIEW | 5 |
| 3. REQUIREMENT ANALYSIS | 9 |
| 3.1 Project Requirements | 9 |
| 3.1.1 Scanner..... | 9 |
| 3.1.2 Jupyter Notebook..... | 9 |
| 3.1.3 Google Collaboratory..... | 9 |
| 3.1.4 Visual Studio Code | 9 |
| 3.1.5 HTML/CSS | 10 |
| 3.1.6 Library Packages..... | 10 |
| 3.1.7 SQLite..... | 12 |

| | | |
|-----------|--|-----------|
| 3.2 | Feasibility Analysis..... | 13 |
| 3.2.1 | Technical Feasibility | 13 |
| 3.2.2 | Economic Feasibility | 13 |
| 3.2.3 | Organizational Feasibility | 14 |
| 3.2.4 | Operational Feasibility | 14 |
| 4. | SYSTEM ARCHITECTURE AND METHODOLOGY | 15 |
| 4.1 | System Block Diagram | 15 |
| 4.2 | Flowchart | 17 |
| 4.3 | Image Preprocessing | 18 |
| 4.3.1 | Grayscale Conversion | 19 |
| 4.3.2 | Canny Edge Detection | 20 |
| 4.3.3 | Hough Line Transform | 21 |
| 4.3.4 | Deskew image..... | 23 |
| 4.4 | Text detection and Extraction using OCR | 25 |
| 4.4.1 | Table detection with YOLOv8..... | 27 |
| 4.4.2 | Paddle OCR to extract table data | 31 |
| 4.4.3 | Tesseract for outside table data extraction..... | 33 |
| 4.5 | SQLite to Store Result in Database | 37 |
| 4.6 | User Interface..... | 39 |
| 4.6.1 | Home Page | 39 |
| 4.6.2 | Admin Dashboard | 39 |
| 4.6.3 | Student Dashboard | 43 |
| 5. | IMPLEMENTATION DETAILS | 44 |
| 5.1 | Integrating various image preprocessing steps | 44 |
| 5.2 | Integrating Yolov8s for table detection | 44 |
| 5.3 | Integrating tesseract-OCR and paddle-OCR..... | 44 |
| 5.4 | Integrating obtained excel file into the database..... | 44 |

| | | |
|-----------|--|-----------|
| 5.5 | Integration in user interface | 45 |
| 5.6 | Database Design..... | 45 |
| 5.6.1 | Table Information | 46 |
| 5.6.2 | Explanation of Relationship between Tables..... | 47 |
| 5.6.3 | Entity Relationship Diagram..... | 47 |
| 5.7 | Use Case Diagram..... | 48 |
| 5.8 | Sequence Diagram | 50 |
| 5.9 | DFD Diagram..... | 53 |
| 5.10 | Activity Diagram | 56 |
| 6. | RESUTLS AND ANALYSIS | 59 |
| 6.1 | User interfaces | 59 |
| 6.1.1 | Student Dashboard..... | 60 |
| 6.1.2 | Admin Dashboard | 63 |
| 6.2 | Table and text extraction..... | 75 |
| 6.3 | Database Schema and Instances..... | 79 |
| 6.4 | Result Analysis (Tesseract VS Paddle)..... | 81 |
| 6.5 | Character Accuracy Rate | 82 |
| 7. | FUTURE ENHANCEMENTS | 83 |
| 7.1 | Integration of Student Records for Enhanced Analysis in QAA Reports..... | 83 |
| 7.2 | Enhancing Accuracy with Paid OCR Integration | 83 |
| 8. | CONCLUSION | 84 |
| 9. | APPENDICES | 85 |
| | Appendix A: Project Schedule | 85 |
| | Appendix B: Project Budget..... | 86 |
| | Appendix C: Code Snippets | 87 |
| | References | 88 |

List of Figures

| | |
|---|----|
| Figure 4-1: System Block Diagram | 15 |
| Figure 4-2: Flowchart of the System | 17 |
| Figure 4-3: Image Preprocessing | 19 |
| Figure 4-4: Flowchart of Text Detection and Extraction..... | 25 |
| Figure 4-5: Architecture of YOLOv8 | 27 |
| Figure 4-6: YOLOv8 Table Detection | 28 |
| Figure 4-7: PaddleOCR Working Procedure/ Architecture | 31 |
| Figure 4-8: Tesseract OCR Working Procedure/ Architecture..... | 34 |
| Figure 4-9: Flowchart for Admin Dashboard | 40 |
| Figure 4-10: Flowchart for Student Dashboard | 43 |
| Figure 5-1: ER Diagram..... | 48 |
| Figure 5-2: Use Case Diagram..... | 48 |
| Figure 5-3: Sequence Diagram | 51 |
| Figure 5-4: Data Flow Diagram..... | 53 |
| Figure 5-5: Activity Diagram..... | 57 |
| Figure 6-1: Home Page | 59 |
| Figure 6-2: Student Login..... | 60 |
| Figure 6-3: Student Dashboard | 60 |
| Figure 6-4: Generated Result | 61 |
| Figure 6-5: Result in PDF | 61 |
| Figure 6-6: Student Progression..... | 62 |
| Figure 6-7: Admin Login | 63 |
| Figure 6-8: Admin Dashboard | 63 |
| Figure 6-9: Upload Marksheet | 64 |
| Figure 6-10: Marksheet Uploaded | 64 |
| Figure 6-11: Result Analysis Webpage | 65 |
| Figure 6-12: Analysis Type Selection..... | 65 |
| Figure 6-13: Programme wise Pass/Fail Bar Graph | 66 |
| Figure 6-14: Subject wise Pass/Fail Bar Graph | 67 |
| Figure 6-15: Semester wise Pass/Fail Percentage Pie-Chart | 68 |
| Figure 6-16: Rank of Students | 69 |
| Figure 6-17: Pass Percentage Trend | 70 |

| | |
|--|----|
| Figure 6-18: Subject Average Marks | 71 |
| Figure 6-19: Grade Distribution Analysis..... | 72 |
| Figure 6-20: Assessment Vs Final Comparison..... | 73 |
| Figure 6-21: Subject Improvement | 74 |
| Figure 6-22: Table Detection | 76 |
| Figure 6-23: Table Extraction | 76 |
| Figure 6-24: Extracted text in a .xlsx file | 77 |
| Figure 6-25: Masking of Table | 78 |
| Figure 6-26: Contextual text extracted | 78 |
| Figure 6-27: Extracted contextual text in a .xlsx file..... | 78 |
| Figure 6-28: Merged excel file | 79 |
| Figure 6-29: Student Information Table | 79 |
| Figure 6-30: Marks Obtained Table..... | 80 |
| Figure 6-31: Subject Information Table | 80 |
| Figure 6-32: Table obtained by using paddle OCR | 81 |
| Figure 6-33: Table obtained by using paddle OCR | 81 |
| Figure 9-1: Gantt Chart..... | 85 |
| Figure 9-2: Code for Extraction from Tesseract OCR..... | 87 |
| Figure 9-3: Code for Extraction from Paddle OCR and Structured Table | 87 |

List of Tables

| | |
|---------------------------------------|----|
| Table 5-1: Student Table..... | 46 |
| Table 5-2: Subject Table..... | 46 |
| Table 5-3: Marks Obtained Table | 47 |
| Table 9-1: Project Budget | 86 |

List of Equations

| | |
|------------|----|
| 4-1 | 19 |
| 4-2 | 20 |
| 4-3 | 20 |
| 4-4 | 20 |
| 4-5 | 21 |
| 4-6 | 22 |
| 4-7 | 22 |
| 4-8 | 23 |
| 4-9 | 23 |
| 4-10 | 24 |
| 4-11 | 29 |
| 4-12 | 30 |
| 4-13 | 30 |
| 4-14 | 32 |
| 4-15 | 33 |
| 4-16 | 35 |
| 4-17 | 35 |
| 4-18 | 35 |
| 4-19 | 36 |
| 6-1 | 82 |

List of Abbreviations

| | |
|-----------|---|
| ACID | Atomicity, Consistency, Isolation, Durability |
| CAR | Character Accuracy Rate |
| CNNs | Convolutional Neural Network |
| CNTK | Cognitive Toolkit |
| CSS | Cascading Style Sheets |
| DBNet | Direct Broadcast Network |
| DeepDeSRT | Deep Learning for Detection and Structure Recognition of Tables |
| EAST | Efficient and Accurate Scene Text |
| EMIS | Educational Management Information System |
| FAIR | Facebook's AI Research lab |
| FDWs | Foreign Data Wrappers |
| HTML | Hypertext Markup Language |
| ICDAR | International Conference on Document Analysis and Recognition |
| ID | Identification |
| IOE | Institute of Engineering |
| LBP | Local Binary Patterns |
| LSTM | Long Short-Term Memory |
| OCR | Optical Character Recognition |
| OpenCV | Open-Source Computer Vision Library |
| ORM | Object-relational Mapper/Mapping |

| | |
|--------|---|
| PIL | Python Imaging Library |
| QAA | Quality Assurance and Accreditation |
| R-CNNs | Region-Based Convolutional Neural Network |
| RDBMS | Relational Database Management System |
| ResNet | Residual Networks |
| SATRN | Self-Attention Text Recognition Network |
| SDMG-R | Spatial Dual-Modality Graph Reasoning |
| SPAs | Single Page Applications |
| SQL | Structured Query Language |
| SQLite | Structured Query Language Lite |
| SSDs | Single Shot Detector |
| SVM | Support Vector Machine |
| UI | User Interface |
| UX | User Experience |
| YOLO | You Only Look Once |

1. INTRODUCTION

This project aims to develop an automated system for extracting and managing marksheet data using Optical Character Recognition (OCR) technology. The system streamlines the process of digitizing academic records, ensuring accurate data extraction, secure storage, and easy access through a user-friendly web interface. By leveraging modern technologies like OpenCV and Tesseract OCR, this project seeks to improve the efficiency and reliability of academic record management in Nepal.

1.1 Background Information

The digitization of marksheets is a vital initiative aimed at improving the management and accessibility of academic records within educational institutions. In Nepal, traditional record-keeping methods rely heavily on manual, paper-based systems, which are not only labor-intensive but also prone to errors. These systems require significant physical storage space, making it challenging to maintain and retrieve records efficiently, especially as the number of students and the volume of records grow. Physical marksheets are also susceptible to environmental damage, loss, and unauthorized access, raising concerns about the security and integrity of these records.

The Institute of Engineering at Tribhuvan University, founded in 2029 B.S., offers diverse engineering programs with an annual enrollment of 14,664 students. The institution gathers substantial data from stakeholders, necessitating thorough analysis and processing for future planning. To manage student and faculty information, syllabi, and course records efficiently, educational institutions utilize an Educational Management Information System (EMIS). By analyzing this data, institutions can evaluate performance metrics and make informed decisions, such as identifying subjects where students encounter challenges and allocating resources accordingly. This initiative aims to enhance educational management by automating the calculation of subject pass rates and facilitating the generation of Quality Assurance and Accreditation (QAA) reports as needed. Furthermore, the project intends to digitize student mark sheets, enabling online analysis of class and subject conditions to support ongoing institutional development.

1.2 Motivation

Since the introduction to image processing and digitization, various projects have been using these technologies to find their applications in different fields. This project is also motivated with a view to helping the areas where there is paper-based and labor-intensive storage of documents like marksheets which is quite troublesome and comes with greater risk of damage and loss. It has also been difficult for students as well as administrators to get easy access to the required academic records. With this project, these tasks can be simplified to a greater extent as it enhances the efficiency and effectiveness of educational record-keeping, benefiting both institutions and students.

1.3 Problem Statement

Every year, approximately 430 students enroll in various engineering programs across each constituent campus of the Institute of Engineering (IOE). Throughout a student's undergraduate journey, extensive data such as semester marks, attendance records, and ID information is generated. Effectively managing this data is crucial for evaluating individual student performance, tracking campus-wide enrollment, and efficiently retrieving specific records as required.

To facilitate these needs, many campuses have adopted an Educational Management Information System (EMIS). However, these systems are often traditional in nature and do not harness the capabilities of modern web technologies. They frequently require manual intervention from system users.

Efforts are underway to enhance these systems, aiming to leverage modern web technologies. This improvement seeks to streamline data management processes, automate tasks, and improve accessibility to information. By upgrading EMIS capabilities, campuses aim to enhance overall operational efficiency and provide more effective support for student management and administrative tasks. Although softcopy of marksheet is required by campuses for record keeping, analysis of results and many more but until now IOE doesn't provide softcopy, it provides only hardcopy to the campuses.

Hence, this project aims to develop a comprehensive Marksheet Digitization system for

IOE constituent campuses to facilitate efficient and accurate storage, update, and retrieval of marksheet of students and easy analysis of data in various fields.

1.4 Objectives

The main objectives of the project are:

- To digitize hardcopy of marksheet using OCR and provide secure access through user-friendly web platforms.
- To develop progress report as per need of QAA.

1.5 Scope and applications

The scope of this project encompasses the development of a comprehensive system for automated extraction, storage, and management of marksheet data using OCR technology. The primary application of this system is within educational institutions, where it streamlines administrative processes by reducing manual data entry and minimizing errors. The system will be usable to many colleges/schools with no availability of digital records for marksheets of students. Teachers can easily access and analyze grades of students and performance data, with the help of this system. Additionally, this system can be adapted for use in other sectors that require efficient management of large volumes of printed data, such as government agencies and private organizations. This system can also be adapted for managing handwritten/digital bills for shops, stores in Nepal and can also be employed in healthcare facilities to digitize patient records, in banking to process financial documents, and in government offices to manage civil records.

1.6 Project Limitation

The system is designed to handle specific types of marksheets, such as those issued by IOE, but it has limitations when applied to marksheets from other institutions or those with variations in design and structure. These limitations include:

Limited Database Schema Flexibility: The existing database schema is tailored to IOE marksheets, making it challenging to extract and store data from marksheets of other institutions with different formats.

Difficulty Handling Noisy Images: The system struggles to accurately extract text from noisy marksheet images, such as those with college stamps, logos, or other extraneous elements that interfere with text recognition.

Addressing these limitations would improve the system's adaptability and accuracy for a wider range of applications.

1.7 Report Organization

This report is divided into ten chapters for clear and concise presentation of the project. Chapter one is the Introduction which covers the background, motivation, problem definition, objective, scope, and application of the project. Second chapter, which is Literature Review, provides an overview of the relevant literature and works related to the project. Chapter three, Requirement Analysis, analyzes the software as well as library packages required for the project. Chapter four, The System Architecture and Methodology, provides a detailed description of the system architecture, flow of the project, image preprocessing steps, text detection and extraction using OCR along with database used and user interface of the project. Chapter five, Implementation Details, describes how the software and libraries have been implemented for the project along with database design, ER diagram, use cases, sequence and DFD diagrams. Chapter Six, Results and Analysis, includes the outcome and analysis of the project. Chapter Seven, Future Enhancement, provides insight about features that can be added in the future to improve the project. Chapter Eight, Conclusion, concludes the report summarizing the main findings. Lastly, Chapter Nine, Appendices, includes the project schedule, budgets, and code snippets.

2. LITERATURE REVIEW

The extraction and manipulation of marksheet data involves several critical tasks, including image preprocessing, text extraction, data structuring, and user interaction through a graphical interface. This literature review examines the existing research and methodologies that underpin the development of this field. Most of these have reported results on table detection and data extraction separately.

Before deep learning, most table detection methods used simple rules or additional data about the tables. “TINTIN: A System for Retrieval in Text Tables” used structural information to detect tables and their individual fields [1]. T. Kasar et al. [2] used intersecting horizontal and vertical lines along with low-level features to identify table regions. They employed an SVM classifier for this classification task and used probabilistic graphical models to detect tables. Silva et al. [3] developed a model using the joint probability distribution over sequential observations of visual page elements and the hidden state of a line (HMM). This approach effectively merged potential table lines into complete tables, achieving a high degree of completeness.

The paper by T. Ojala et al. introduces Local Binary Patterns (LBP) [4] as a method for texture classification that is invariant to grayscale variations and rotations. The authors demonstrate the effectiveness of LBP in capturing local texture patterns by encoding the relationship between a pixel and its neighbors using binary patterns. This technique allows for robust texture classification across different illumination conditions and orientations. The paper discusses various applications of LBP in image analysis and computer vision, highlighting its versatility and efficiency compared to traditional texture descriptors.

In the studies by Dos Santos et al. [5] and Likforman-Sulem et al. [6], text extraction is seen as a vital part of analyzing document images, but there is no one-size-fits-all solution. To effectively separate text from a document page, it's important to find all possible text areas. Text image segmentation involves finding objects and boundaries like lines and curves in images. This process labels parts of the image, like pixels or connected components, based on similar visual features, creating segments that cover the entire image or outlining the shapes. Each segment has pixels with similar

characteristics, such as color, brightness, or texture, while adjacent segments show noticeable differences in these features.

In their paper presented at the 2011 International Conference on Computer Vision, Wang et al. proposed a trainable neural network designed for scene text recognition. [7] This method was capable of recognizing text directly from images without requiring separate preprocessing steps like character segmentation. The paper's significant contributions include the development of a unified framework that combines text detection and recognition into one trainable system and the use of convolutional neural networks (CNNs) for both feature extraction and sequence modeling.

In another approach, D. N. Tran et al. proposed “Table Detection from Document Image using Vertical Arrangement of Text Blocks” [8] which located text components and extracted text blocks. The height of each text block was then compared to the average height, and if it met a series of criteria, the region of interest was classified as a table.

Zhao et al.'s review article offers an extensive examination of the current state of object detection techniques within the realm of deep learning. [9] The authors meticulously explore the evolution of deep learning architectures, including CNNs, R-CNNs, SSDs, and YOLO networks. They cover a wide spectrum of methodologies and innovations, emphasizing the strengths and weaknesses of each approach. The review underscores the significance of evaluation criteria and benchmark datasets in measuring the performance of these models, providing invaluable insights into advanced methods for tackling challenges such as scale variation and occlusion. This study delivers a brief yet comprehensive overview of the latest trends and advancements in deep learning-based object detection, making it an essential resource for researchers and practitioners.

Another paper by X. Zhao [10] utilized grid text in Convolutional Neural Networks, embedding text into the document as a feature that includes both semantic meaning and spatial distribution. The study of this paper reveals that there is a need to enhance semantic feature extraction while also considering image-level features. R. Palm describes a system that employs RNNs and LSTM to capture the contextual information of data within a document. [11] Based on the context of the words, the model attempts to generalize to unseen templates. However, the model does not account for the spatial

layout of the document and extracts key-value pairs in a left-to-right sequence.

DeepDeSRT was proposed as a method that employs deep learning for both table detection and table structure recognition, including identifying rows, columns, and cell positions within detected tables [12]. The paper by Kavasidis et al. integrated deep convolutional neural networks, graphical models, and saliency concepts to localize tables and charts in documents [13]. This technique was tested on an extended version of the ICDAR 2013 table competition dataset and outperformed existing models.

V. Sunder et al. described two techniques for converting document images into structured formats in his paper [14]. The first technique involves neural learning, which uses a pre-trained deep learning model to read and transform document images into structured data by incorporating a predefined database schema. The second technique involves reusable logic programs that extract entities from a document. These programs use the entities and primitive links identified through neural learning to synthesize information. This approach includes a template-free solution that learns to detect both pre-printed and handwritten text, as well as predicting pairwise relationships. A convolutional network method is used to detect pre-printed and handwritten text lines, and the functionality from the detection network is integrated to semantically classify potential connections.

S. Paliwal et al. describe deep learning end-to-end models for both table detection and structure recognition in their paper. [15] These models leverage the interdependence between the tasks of detecting tables and recognizing their structure, focusing on dividing table and column regions. After identifying table subregions, semantic rule-based methods are used to extract rows. This approach is content-based, resistant to noise, and generalized for handling various unseen document formats. It effectively tackles the challenge of extracting key fields from a diverse range of document formats that may not have been previously encountered.

In early 2022, Ambroise Berthe conducted a study titled "Text extraction from ID cards using deep learning". [16] The research involved a studious process to gather data for training a model to identify and extract information from ID documents in images. The images were carefully cropped and straightened to create a suitable database. Berthe

opted for specific deep learning models in Python, preferring YoloV7 for document segmentation and ResNet over OpenCV for image straightening. DBNet was selected for text detection, and SATRN was chosen for text recognition, both noted for their speed and accuracy. For information extraction, SDMG-R was used, leveraging its effective use of spatial relationships and features in the text regions.

3. REQUIREMENT ANALYSIS

3.1 Project Requirements

For the project, following hardware, libraries, frameworks, and tools are used:

3.1.1 Scanner

A scanner is a device which captures the image from physical documents like marksheets. It is used to capture clear and detailed images with high resolution which is effective for effective image processing and character recognition.

3.1.2 Jupyter Notebook

Jupyter Notebook is an open-source interactive web application that facilitates the creation and sharing of documents containing live code, equations, visualizations, and narrative text. It supports Python and various other programming languages as well. Users can produce dynamic reports, conduct data analysis, and create machine learning models within a single, user-friendly interface.

3.1.3 Google Collaboratory

Google Collab is a cloud-based service offered by Google that facilitates writing, executing, and collaborating on Python code within an interactive and convenient environment. It provides numerous features and advantages, making it a preferred choice for data scientists, researchers, and developers. The platform supports seamless collaboration, access to powerful computational resources, and easy integration with various data science libraries and tools. This combination of functionality and ease of use has established Google Collab as a valuable tool for a wide range of coding and data analysis tasks.

3.1.4 Visual Studio Code

Visual Studio Code is open and versatile source code editor from Microsoft which stands out for its speed and broad language support. Visual Studio Code offers a highly customizable experience through extensions, packed with features like intelligent code completion, debugging tools, an integrated terminal, and Git integration. Its cross-

platform compatibility has led to widespread adoption among developers.

3.1.5 HTML/CSS

HTML is the standard markup language used to create and design documents on the web. It structures the content on the web and consists of a series of elements or tags that tell the browser how to display the content.

CSS is a style sheet language used to describe the presentation and formatting of a document written in HTML or XML. It controls the layout, colors, fonts, and overall visual appearance of web pages. When combined, HTML provides the structure, and CSS provides the styling, allowing developers to create visually appealing and well-structured web pages.

3.1.6 Library Packages

3.1.6.1 Keras

Keras is a highly popular open-source deep learning library that provides a high-level, user-friendly interface for creating and training neural networks. Developed with simplicity and ease of use in mind, Keras allows for rapid prototyping and experimentation. It supports multiple backend engines, such as TensorFlow, Theano, and Microsoft Cognitive Toolkit (CNTK), making it versatile and adaptable. Its straightforward and intuitive design has made it a preferred choice for both novices and experienced researchers in the machine learning field, enabling them to efficiently develop and deploy deep learning models.

3.1.6.2 Scikit-Learn

Scikit-learn is a highly utilized open-source machine learning library offering an extensive suite of tools for data preprocessing, feature selection, model training, and evaluation. Its wide array of algorithms and user-friendly interface have made scikit-learn a favored choice among machine learning practitioners.

3.1.6.3 NumPy

NumPy is an essential open-source library for numerical computing in Python. It features a powerful array object and a suite of functions for efficient manipulation of

large multidimensional arrays and matrices. As the backbone of many scientific and data-focused Python packages, NumPy is a crucial tool for numerical computations.

3.1.6.4 Pandas

Pandas is a robust open-source Python library designed for high-performance data manipulation and analysis. Built on top of NumPy, it provides a flexible and intuitive interface for handling structured data, making it an essential tool for data scientists and analysts.

3.1.6.5 Matplotlib

Matplotlib is a widely used data visualization library in Python, offering a comprehensive suite of tools for creating static, animated, and interactive plots. Its flexible and customizable interface allows for the generation of a diverse array of visualizations, making it a favored choice among data scientists and researchers.

3.1.6.6 OpenCV

OpenCV is a popular open-source library designed for computer vision and image processing tasks. It offers a rich set of functions and algorithms that enable users to carry out a wide variety of image and video analysis operations. OpenCV also provides extensive support for image and video input/output operations, allowing users to read and write images and videos in multiple formats.

3.1.6.7 Pytesseract

Pytesseract is a Python wrapper for Google's Tesseract-OCR Engine. It allows developers to use Tesseract-OCR functionality within Python scripts to perform optical character recognition (OCR) on images. Pytesseract simplifies the process of extracting text from images and is widely used for tasks such as extracting text from scanned documents, receipts, and images containing textual content.

3.1.6.8 Django

Django is a high-level Python web framework known for its simplicity, versatility, and rapid development capabilities. It provides a comprehensive set of tools and features including an ORM for database interactions, a built-in admin interface, robust security

features, and a powerful template engine for generating dynamic web content. Django follows a "batteries-included" philosophy, offering everything needed to build web applications efficiently while promoting clean and maintainable code. Its scalability, extensive documentation, and active community make it a preferred choice for developers aiming to create secure, scalable, and feature-rich web applications.

3.1.6.9 Django ORM

The Django ORM (Object-Relational Mapping) is a powerful feature of the Django web framework that allows developers to interact with the database using Python code instead of raw SQL queries. It provides a high-level abstraction for database operations, making it easier to create, retrieve, update, and delete records in the database.

3.1.6.10 PyTorch

PyTorch, created by Facebook's AI Research lab (FAIR), is an open-source machine learning library renowned for its flexibility, dynamic computation graphs, and ease of use. It provides a robust platform for building and training neural networks, prioritizing a more intuitive and Pythonic interface, which makes it particularly appealing to developers.

3.1.6.11 PP structure

Paddle-OCR pipeline (PP) Structure involves a sequence of steps from preprocessing images to detecting text, recognizing it, and then post-processing the results. This structure ensures efficient and accurate OCR performance. Paddle-OCR has a series of models called PP-OCR, which are optimized for different use cases, such as lightweight models for mobile devices and more robust models for complex scenarios.

3.1.7 SQLite

SQLite is a lightweight, self-contained, serverless database engine that is widely recognized for its simplicity and ease of use. It operates as a complete database engine within a single library file, requiring no external dependencies or configuration. Unlike other relational database management systems (RDBMSs), SQLite is serverless, as it functions directly from the application process, storing data in a single file on disk. Its compact library size, often less than 1 MB, makes it ideal for embedded systems and

small applications. SQLite is platform-independent, running seamlessly on operating systems such as Windows, macOS, and Linux. It ensures data reliability by adhering to ACID (Atomicity, Consistency, Isolation, Durability) principles, and as open-source software, it is free to use and modify.

The portability and simplicity of SQLite have made it a popular choice across various use cases. It is extensively used in embedded systems, such as mobile devices (e.g., Android and iOS), IoT devices, and hardware products. Desktop applications often leverage SQLite for local storage in programs like web browsers and media players. Additionally, it is a preferred tool for testing, prototyping, and lightweight data analysis due to its ease of implementation and minimal resource requirements. While SQLite excels in smaller-scale applications, it has some limitations, such as limited support for high-concurrency environments, advanced features like stored procedures, and robust user management. Despite these constraints, SQLite remains an excellent choice for projects where simplicity, portability, and reliability are key priorities.

3.2 Feasibility Analysis

3.2.1 Technical Feasibility

The technical feasibility of this project looks promising. By using Tesseract OCR, we can effectively extract text from marksheet images, and with image preprocessing techniques, we can further improve accuracy. Databases like SQLite or MySQL handles data storage and retrieval efficiently. The backend is built with Django and the frontend with React, both of which are modern and robust frameworks. Hosting the system on a database server ensures the scalability, reliability, and accessibility

3.2.2 Economic Feasibility

Economically, this project is viable. The initial costs for hardware and software development are manageable, especially with the use of free, open-source software. Ongoing costs, like server maintenance and cloud services, are relatively low compared to the savings from reduced manual labor and faster processing times. Automation leads to significant long-term savings. Additionally, there are opportunities for funding from government programs, NGOs, and international grants aimed at promoting digital education in Nepal.

3.2.3 Organizational Feasibility

The project exhibits strong organizational feasibility with its user-friendly design, incorporating a straightforward and adaptable user interface (UI). The simplicity makes it easy for individuals with basic English language skills to operate the software seamlessly, irrespective of their location. This accessibility is a significant organizational advantage, reducing the necessity for extensive training. Being a web-based application, it allows operation through mobile phones.

3.2.4 Operational Feasibility

The system required to execute this project includes a basic setup with an Intel i3 processor, 4 GB of RAM, and a 500 GB hard drive, running on a standard Windows 7 operating system. Additionally, the project can be accessed and operated via a straightforward mobile website interface. This setup ensures that the project is both feasible and operationally efficient, as it leverages readily available, cost-effective hardware and software configurations. The compatibility with a mobile website interface further enhances its accessibility and usability, making it practical for a wide range of users.

4. SYSTEM ARCHITECTURE AND METHODOLOGY

4.1 System Block Diagram

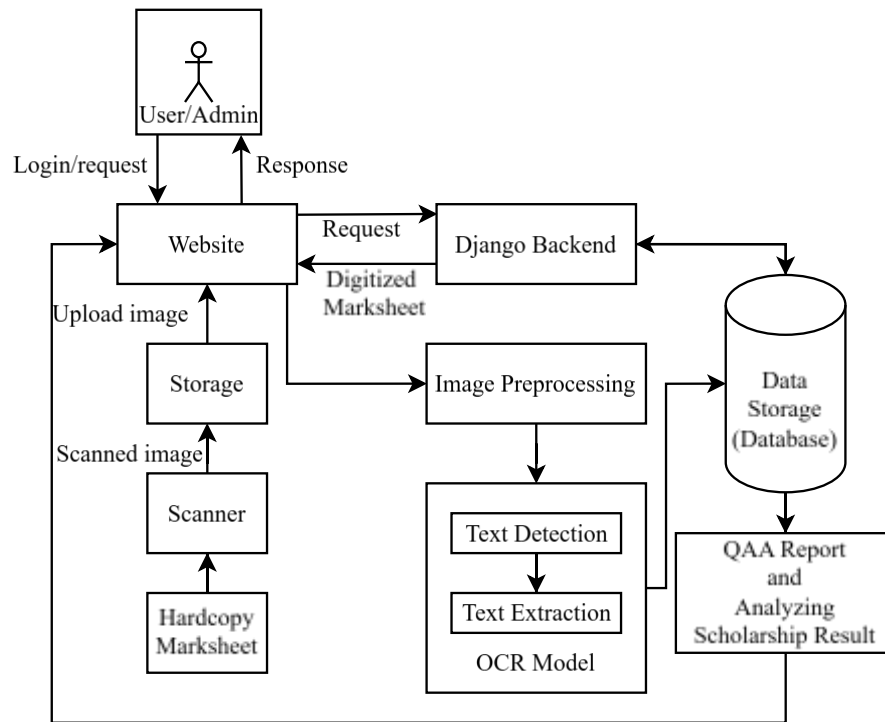


Figure 4-1: System Block Diagram

The system begins with a simple and user-friendly web interface, where both students and administrators can log in to access features tailored to their needs. Students use the platform to view their digitized marksheets and download them if necessary, providing a straightforward and convenient way to access their academic records. Administrators, however, have additional responsibilities. They log in to upload scanned marksheets, which triggers a series of automated steps to process and digitize the data.

When an administrator uploads a scanned marksheet through the web interface, the image is first saved in temporary storage. This ensures the system has access to the original file for the next stages of processing. The Django backend, which acts as the central hub for all system operations, retrieves the image from storage and sends it to the Image Preprocessing module. Here, the system uses OpenCV to clean up and optimize the image. This involves reducing noise to remove any distortions, enhancing contrast to make the text more legible, and correcting any skew to properly align the image. These adjustments are essential for preparing the image for the next step: text

extraction.

Once preprocessing is complete, the enhanced image is passed to the OCR (Optical Character Recognition) module. This is where the system extracts meaningful data from the image. The OCR first identifies areas containing text and then converts that text into a machine-readable format. The processed data is sent back to the Django backend, which stores it securely in the SQLite database. This database acts as the system's memory, keeping all digitized marksheet data organized and ready for use.

Students interact with the database indirectly through the web interface. When a student logs in and requests their marksheet, the backend retrieves the relevant data from the database and displays it on the web interface. Students can then download their records if they wish. Administrators, on the other hand, use the stored data for deeper insights. They can generate reports, such as Quality Assurance (QAA) assessments and scholarship analyses, based on predefined criteria. The backend processes these requests, pulls the required data from the database, and presents the results to administrators in the web interface.

Every part of the system is closely connected and works together seamlessly. The web interface provides users with a clear and accessible way to interact with the system. The backend ensures that all components—temporary storage, preprocessing, OCR, and the database—are coordinated efficiently. This interconnected workflow ensures that scanned images are processed accurately, data is stored securely, and both students and administrators can get exactly what they need. The system is designed to be intuitive, reliable, and easy to use, giving students quick access to their academic records while helping administrators manage and analyze data effortlessly.

4.2 Flowchart

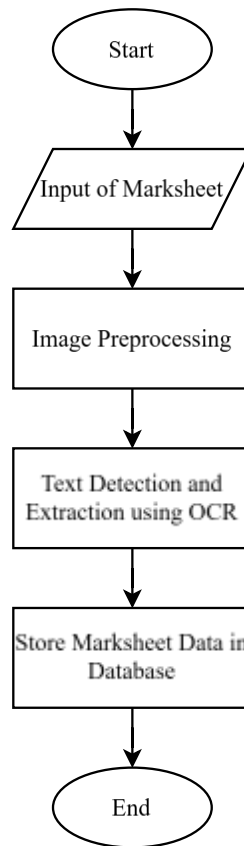


Figure 4-2: Flowchart of the System

The flowchart outlines a detailed process for digitizing marksheets through text detection and extraction. It begins with the input of a marksheet image, which undergoes several preprocessing steps. The image is first converted to grayscale, followed by edge detection using the Canny edge detection technique, and then line detection with the Hough Line Transform. If the image is tilted, it is deskewed to ensure proper alignment. The next phase involves text detection and extraction, where a YOLOv8 (You Only Look Once) model is employed to detect the table portion of the marksheet. This detected table is cropped, and text within it is extracted and formatted into a table structure. The formatted table is then converted into an Excel (.xlsx) file. Simultaneously, the table in the original marksheet image is masked to facilitate the extraction of remaining text, which is also converted into an Excel (.xlsx) file. Finally, the two Excel files (one containing the table data and the other containing the remaining text) are merged into a single file, ensuring that all relevant information from the marksheet is digitized accurately.

This comprehensive approach allows for precise extraction of structured and unstructured data, ensuring that no important details are missed. The use of advanced techniques like the YOLOv8 model for table detection highlights the integration of modern machine learning methods in document processing. Grayscale conversion and edge detection simplify the image, making it easier to identify and extract text accurately. Deskewing ensures that even poorly scanned documents are correctly processed. Converting extracted data into Excel format ensures compatibility with widely used data analysis and management tools. Overall, this process not only automates the digitization of marksheets but also enhances the accuracy and efficiency of data extraction, making it a valuable tool for educational institutions and other organizations that handle large volumes of documents.

4.3 Image Preprocessing

Image preprocessing plays a crucial role in preparing input data for OCR systems, ensuring that extracted text is accurate and reliable. By applying these preprocessing techniques, OCR performance can be significantly enhanced, leading to more effective digitization of marksheets.

For Image preprocessing, OpenCV is used. These libraries are used for Image Enhancement. OpenCV is used for more sophisticated tasks such as noise reduction, contrast adjustment, thresholding, and geometric transformations. Various Image preprocessing inference includes:

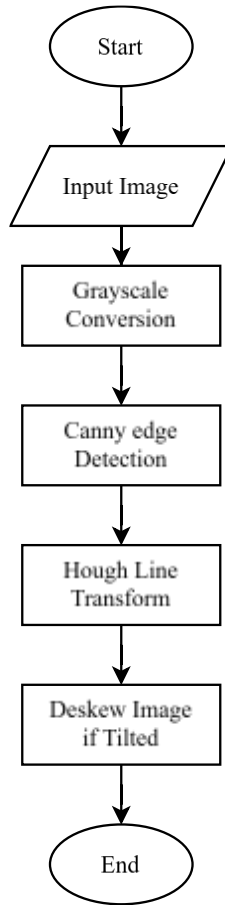


Figure 4-3: Image Preprocessing

4.3.1 Grayscale Conversion

This process transforms a color image into a grayscale by calculating the weighted average of its red, green, and blue channels. Grayscale simplifies the image, facilitating subsequent processing steps. This process converts images from other color spaces like RGB, HSV, etc into shades of grey. It varies from complete black to complete white. Grayscale conversion calculates the brightness of each pixel using a weighted sum of the Red, Green, and Blue channels. The formula used is:

$$Gray = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad 4-1$$

These weights are based on human perception, where green contributes the most to brightness, followed by red, then blue.

4.3.2 Canny Edge Detection

Canny edge detection starts with a Gaussian blur to reduce noise. It then calculates the image gradient to identify areas of high-intensity change, classifies them as edges, and applies thinning operations for simplification. The outcome is a binary image marking object edges. Canny edge detection involves the following steps:

4.3.2.1 Noise Reduction

The image is first smoothed using a Gaussian filter to reduce noise, which helps prevent false edge detection.

4.3.2.2 Gradient Calculation

The image gradients are calculated using Sobel filters to find the intensity and direction of edges. These gradients help determine the strength (magnitude) and orientation of edges. The matrix of Sobel filter is:

Sobel X (for detecting vertical edges)

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad 4-2$$

Sobel Y (for detecting horizontal edges)

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad 4-3$$

Once the gradients G_x and G_y are computed, the gradient magnitude is given by:

$$G = \sqrt{G_x * G_x + G_y * G_y} \quad 4-4$$

And the gradient direction (edge orientation) is:

$$\tan \theta = G_y/G_x \quad 4-5$$

These filters highlight intensity changes in images, making them useful for edge detection.

4.3.2.3 Non-Maximum Suppression

This step thins out the edges by retaining only local maxima in the direction of the gradient. It ensures that edges are only marked at the sharpest points. For each pixel, compare its gradient magnitude with its two neighboring pixels in the gradient direction. If the pixel's magnitude is not the highest, suppress it (set it to zero).

4.3.2.4 Double Thresholding

Two threshold values (lower and upper) are used to classify edge pixels:

Strong edges: Pixels with gradients above the upper threshold.

Weak edges: Pixels with gradients between the lower and upper thresholds.

Non-edges: Pixels with gradients below the lower threshold are discarded.

4.3.2.5 Edge Tracking by Hysteresis

Weak edges connected to strong edges are retained, while unconnected weak edges are discarded. This ensures continuity of edges.

4.3.3 Hough Line Transform

Following edge detection, the Hough Line Transform is used to detect straight lines in the image. This method is particularly useful for identifying the rows and columns of tables, as well as any other linear elements. Detecting these lines helps in accurately locating and isolating tables from the rest of the text.

The basic idea behind the Hough Line Transform is to represent a line in a different way using polar coordinates instead of the traditional Cartesian equation. The steps involved in hough line transform are listed below.

1. Convert the image to grayscale.
2. Apply edge detection.
3. Transform edge points into Hough space and accumulate votes.
4. Detect peaks in the accumulator, which correspond to the strongest lines.
5. Convert detected lines back to Cartesian coordinates and overlay them on the image.

4.3.3.1 Line Representation

A line in Cartesian form is represented as:

$$y = mx + c \quad 4-6$$

However, this representation fails for vertical lines where the slope m is undefined. Instead, the polar form is used:

$$\rho = x \cos\theta + y \sin\theta \quad 4-7$$

where:

- ρ is the perpendicular distance from the origin to the line.
- θ is the angle of the normal to the line.

4.3.3.2 Hough Space (Accumulator Matrix)

The accumulator matrix (also called Hough Space) is a key component of the Hough Line Transform. It is a 2D array used to store votes for potential lines in an image. Each point in the image votes for all possible lines passing through it by computing values of ρ for different θ values. These votes are stored in an accumulator matrix (Hough space). The peaks in this accumulator correspond to the most likely line candidates. The accumulator size can be large for high-resolution images, so techniques like

probabilistic Hough Transform (PHT) help reduce computational cost by randomly sampling edge points instead of processing all of them.

4.3.4 Deskew image

To ensure the marksheet is correctly aligned, the system checks for any tilt in the image. If the image is tilted, it is rotated to align it properly. Proper alignment is essential for accurate text detection and extraction. Deskewing an image using OpenCV involves detecting the skew angle and rotating the image to correct it. The most common approach is to use edge detection and contours to determine the text or object orientation. The steps involved is given below.

1. Convert the image to grayscale.
2. Apply thresholding to create a binary image.
3. Find contours and determine the minimum bounding box angle.
4. Rotate the image to correct the skew.

4.3.4.1 Finding the Skew Angle

Using OpenCV's `cv2.minAreaRect()`, we determine the smallest bounding box around the text or object. The function returns an angle θ (theta), which is used to correct the skew. If $\theta < -45^\circ$, the angle is adjusted using:

$$\theta_{corrected} = -(90 + \theta) \quad 4-8$$

Otherwise,

$$\theta_{corrected} = -\theta \quad 4-9$$

4.3.4.2 Rotation Matrix Calculation

To rotate the image by angle θ , we use the 2D affine transformation matrix. This matrix is computed in OpenCV using:

`M=cv2.getRotationMatrix2D((xc,yc), θ ,1.0)`

where:

- (x_c, y_c) is the center of rotation (usually the image center),
- θ is the corrected skew angle.

4.3.4.3 Applying Rotation to the Image

Once we have the rotation matrix M , we use it to warp the image using:

$$I' = M \cdot I \quad 4-10$$

where:

- I is the original image pixel coordinates,
- I' is the new (deskewed) image coordinates.

This ensures that the image is properly deskewed and aligned.

4.4 Text detection and Extraction using OCR

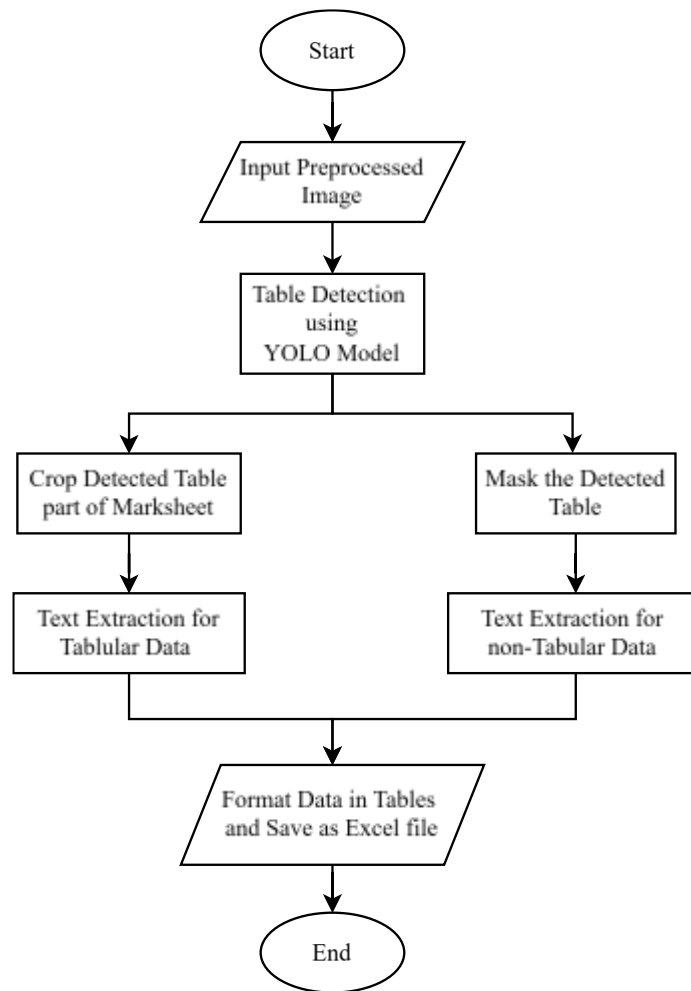


Figure 4-4: Flowchart of Text Detection and Extraction

With the image prepared, the YOLOv8 model is used to detect tables within the marksheet. Once a table is identified, it is cropped out, isolating it for detailed processing. For the rest of the marksheet (non-tabular content), the system uses a masking technique. This involves covering the detected table region with a solid color (i.e. black), effectively removing it from the image. This ensures that the remaining text extraction focuses only on non-tabular information without interference.

For the tabular data, the cropped table is processed using PaddleOCR, which extracts both the structure of the table and the data within it. This is especially useful for handling complex tables with rows and columns. The extracted table data is then neatly formatted into an Excel file. For the non-tabular parts of the marksheet, the masked image is processed with Tesseract OCR. Before text extraction, additional

enhancements like contrast adjustments and noise reduction are applied to improve the accuracy of the results.

Once all text is extracted, the data is cleaned and organized. Specific fields, such as student names, roll numbers etc. are identified and paired using predefined patterns or regular expressions. This ensures that the information is properly structured and ready for use. Both the tabular and non-tabular data are saved into separate Excel files for easy access and management.

At the end of the process, all the individual Excel files are combined into a single, comprehensive file. This step simplifies data organization and makes it easier to analyze or share. Temporary files created during the workflow are cleaned up to keep the workspace tidy and efficient.

4.4.1 Table detection with YOLOv8

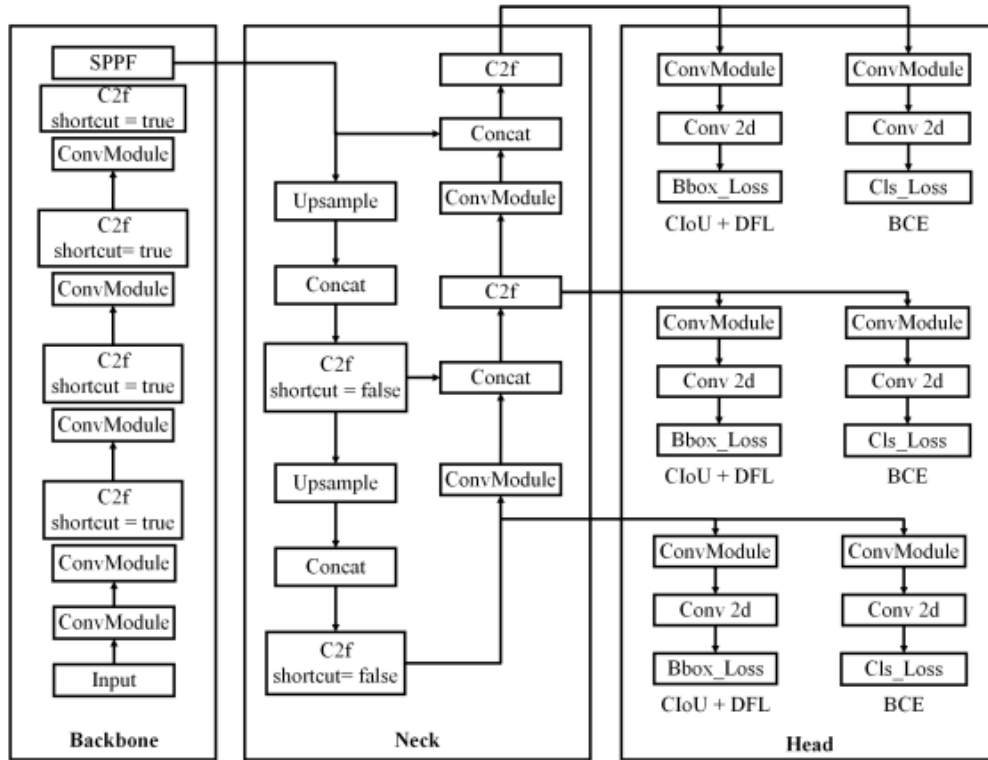


Figure 4-5: Architecture of YOLOv8

The YOLOv8 object detection system, like its predecessors is a single-stage detector that processes and input image to predict bounding boxes and class probabilities for objects all in a single go. Here is a look at how YOLOv8 operates, broken down into several key stages. The architecture consists of a Backbone Neck and Head. The Backbone is a convolutional neural network (CNN) that is primarily responsible for extracting feature maps from the input image. It processes the image through multiple layers of convolutions, capturing various levels of abstraction and important spatial details. The Neck component is responsible for aggregating the features extracted by the Backbone. This is typically achieved using path aggregation blocks like the Feature Pyramid Network (FPN), which combines feature maps from different scales to create a rich, multi-scale feature representation. Then it passes them onto head, predicting the final bounding boxes and class probabilities for detected objects.

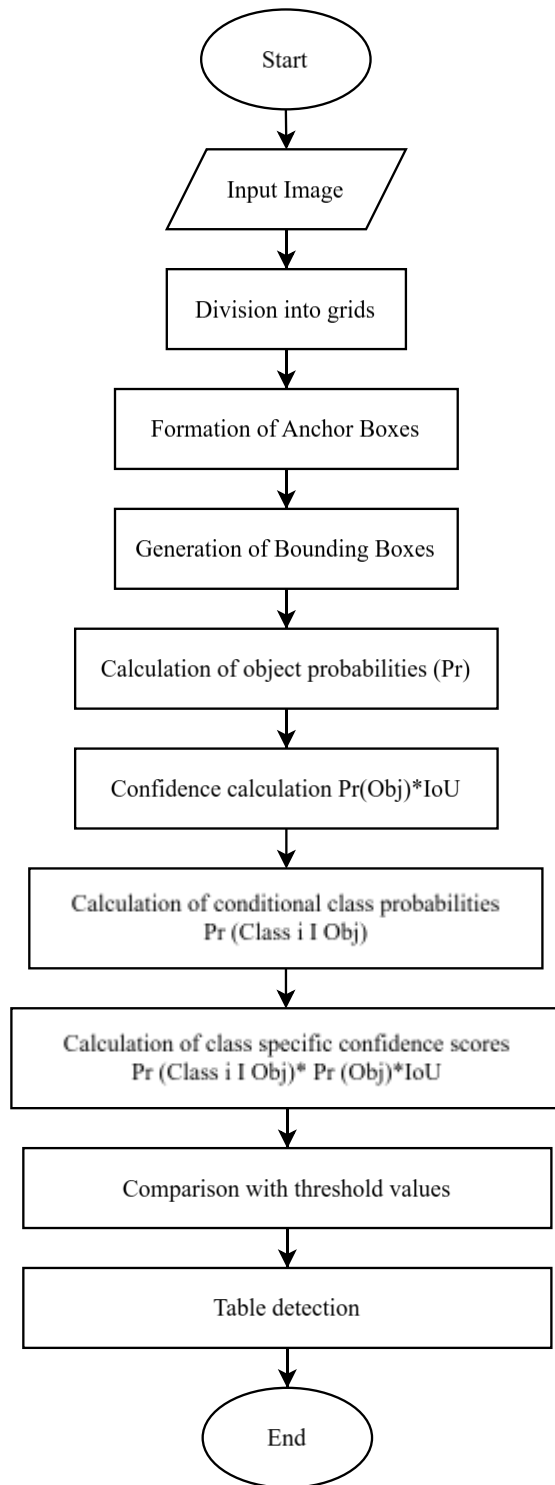


Figure 4-6: YOLOv8 Table Detection

The YOLOv8s Table Detection model is an advanced object detection model built on the YOLO (You Only Look Once) framework. It specializes in identifying tables within images, regardless of whether they have visible borders or are borderless. This model has been meticulously fine-tuned using an extensive dataset, enabling it to achieve

impressive accuracy in detecting tables and differentiating between bordered and borderless types. Here's how the YOLO process works:

Input Image: The YOLO model takes the entire image as input. This image is resized to a fixed size (e.g., 416x416 pixels) to maintain consistency during processing.

Division into Grids: The image is divided into a grid of $S \times S$ cells. Each cell is responsible for predicting objects whose center lies within it. The model processes the image through multiple stages, generating feature maps at different scales to detect objects of various sizes. Specifically, YOLOv8s produces feature maps with the following grid sizes:

80x80 grid captures fine details, aiding in the detection of small objects in the image.

40x40 grid balances between detail and context, suitable for medium-sized objects.

20x20 grids focuses on larger objects, capturing broader spatial information.

Formation of Anchor Boxes: Predefined anchor boxes with different sizes and aspect ratios are assigned to each grid cell. These boxes act as templates for predicting bounding boxes for objects of varying shapes and sizes.

Generation of Bounding Boxes: The model predicts the bounding box coordinates (center x, y, width, and height), refining them relative to the anchor boxes.

Calculation of Object Probabilities (Pr): YOLO predicts the confidence score $Pr(Object)$, which indicates whether an object exists in a bounding box.

Confidence Calculation:

$$Pr(Object) \times IoU. \quad 4-11$$

The confidence score is further refined using the Intersection over Union (IoU), which measures the overlap between the predicted bounding box and the ground truth box.

Calculation of Conditional Class Probabilities:

$$Pr(Class | Object) \quad 4-12$$

For each bounding box, YOLO predicts the probabilities of the object belonging to various predefined classes. These probabilities are conditional on the presence of an object.

Calculation of Class-Specific Confidence Scores: The final confidence score for each class is calculated as:

$$Pr(Class | Object) \times Pr(Object) \times IoU \quad 4-13$$

This score combines the likelihood of an object existing, the IoU, and the probability of the object belonging to a specific class.

Comparison with Threshold Values: YOLO filters out low-confidence predictions by applying a threshold to the confidence scores. Predictions below this threshold are discarded.

Table Detection: After filtering, YOLO outputs the final bounding boxes and class labels (e.g., "table") for detected objects. Non-Maximum Suppression (NMS) is applied to remove duplicate detections.

In this project pre-trained YOLO (You Only Look Once) model is employed for detecting tables within marksheet images. This state-of-the-art, real-time object detection system is pre-trained on a dataset that includes table structures. The process involves loading the YOLOv8 model and configuring it with specific parameters such as confidence threshold, IoU (Intersection over Union) threshold and maximum detections. The image is then passed through the YOLOv8 model, which outputs bounding boxes around detected tables along with confidence scores. These bounding boxes are used to crop the table region from the original image, isolating the table for further processing.

4.4.2 Paddle OCR to extract table data

The architecture of PaddlePaddle (PARallel Distributed Deep LEarning) is designed to be flexible, scalable, and efficient, supporting a wide range of deep learning tasks. It offers both dynamic and static computational graph modes, making it adaptable for research and production. PaddlePaddle supports distributed training with data and model parallelism, optimized communication strategies, and mixed-precision training for high efficiency. Its modular design includes specialized libraries for tasks like recommendation systems (PaddleRec), OCR (PaddleOCR), NLP (PaddleNLP), and GANs (PaddleGAN). It is optimized for various hardware platforms, including CPUs, GPUs, and accelerators, and provides a lightweight, high-performance inference engine for deployment on cloud, mobile, and edge devices. PaddlePaddle also includes automated tools for hyperparameter tuning, pre-trained models, and seamless deployment options. With user-friendly APIs, visualization tools, and robust debugging support, PaddlePaddle is a powerful framework for both beginners and experts in AI development.

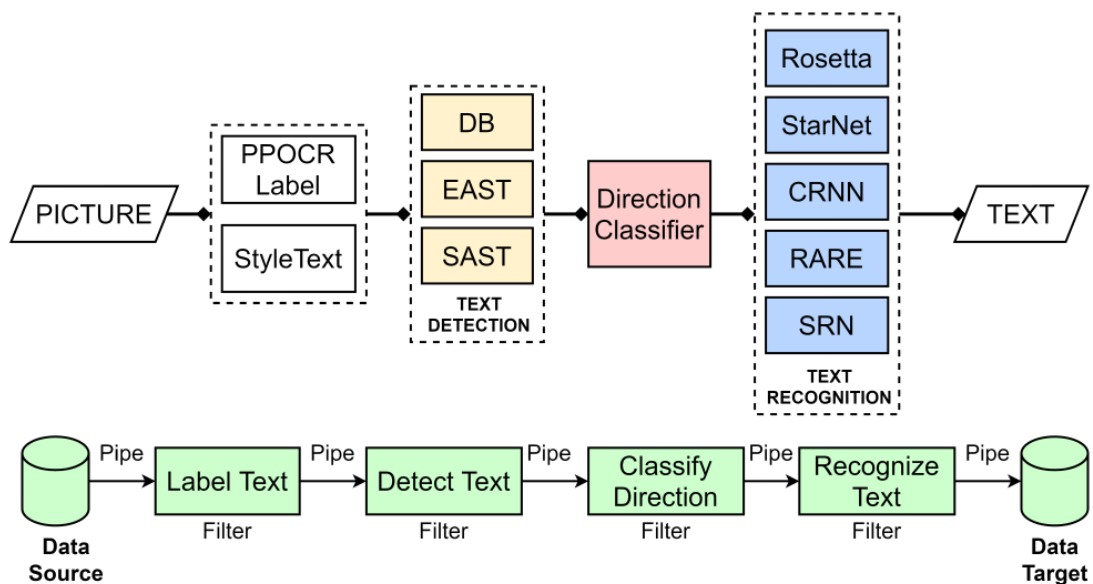


Figure 4-7: PaddleOCR Working Procedure/ Architecture

Below is a step-by-step explanation with relevant equations and description of the working procedure of Paddle OCR :

Input (Picture): The pipeline begins with an input image that contains text, such as scanned documents, photos, or screenshots.

Labeling Text (PPOCR Label) Purpose: Annotate the text regions within the image for supervised training. PPOCRLabel is a graphical tool for labeling text regions and assigning text content to them. It generates labeled data used to train text detection and recognition models.

Text Detection: The labeled image is passed through one or more text detection algorithms to locate text regions. The output obtained is bounding boxes for detected text regions. Some of the algorithms that are used in paddle OCR for text detection are described below:

- i. **DB (Differentiable Binarization):** Detects text regions by binarizing feature maps.

$$Loss = BCE(\hat{P}, P) + \lambda \cdot Dice(\hat{P}, P) \quad 4-14$$

Here:

- \hat{P} : Predicted probability map
 - P : Ground truth binary map
 - λ : Balancing coefficient
- ii. **EAST (Efficient and Accurate Scene Text Detector):** Detects text regions with quadrilateral bounding boxes.
 - iii. **SAST (Segmentation-Based Scene Text Detector):** Improves detection of irregular text shapes. Output: Bounding boxes for detected text regions.

Direction Classifier: It Classifies the orientation of detected text (e.g., 0°, 90°, 180°, or 270°). A lightweight neural network predicts the orientation of text to normalize it for recognition. It gives rotated or aligned text regions.

Text Recognition: Once text regions are localized and oriented, they are passed through a text recognition model. Some of the algorithms used are:

- i. **Rosetta:** Recognizes text using convolutional and recurrent networks.
- ii. **StarNet:** Corrects geometric distortions in text regions for better recognition.

- iii. **CRNN (Convolutional Recurrent Neural Network):** Combines CNNs and RNNs for feature extraction and sequential modeling.

$$y = \text{Softmax}(W_z + b) \quad 4-15$$

- W, b : Models weights.
 - x : Input feature vector.
- iv. **RARE:** Addresses irregular text recognition using attention mechanisms.
- v. **SRN (Sequence Recognition Network):** Enhances text recognition accuracy through self-attention.

In this process, recognized text content from each text region is obtained.

Output (Text): The final output is structured text extracted from the image, which can be stored or processed further.

This modular design makes PaddleOCR flexible for handling different scenarios like multi-language recognition, bordered or borderless text, and complex layouts. In this project, after the table part from the marksheet image is cropped, PaddleOCR's PP-Structure is used to extract the data from them. PaddleOCR is a powerful tool for text detection and recognition in images, and PP-Structure is specifically designed for handling complex layouts like tables. PaddleOCR is set up with the right settings, like enabling angle classification and specifying that the text is in English. The cropped table image is then fed into PP-Structure, which analyzes the table structure, identifying rows and columns. The extracted table data is then saved as an XLSX file, preserving the original table format from the image.

4.4.3 Tesseract for outside table data extraction

The architecture of Tesseract OCR is designed as a modular pipeline that processes images to extract text accurately. It begins with input image acquisition, where the system accepts various formats like JPEG, PNG, BMP, and TIFF. The image undergoes preprocessing steps such as binarization, noise removal, and deskewing to enhance quality and prepare for text extraction. Tesseract then performs page segmentation,

dividing the image into regions containing text, graphics, and other elements, and further identifies structures like paragraphs, columns, and tables. The core OCR process involves two main steps: character segmentation and character recognition. The image is divided into words and then individual characters, with overlapping or touching characters separated. Tesseract uses an adaptive recognition engine that combines pattern matching with a neural network-based approach, specifically a Long Short-Term Memory (LSTM) network, to improve contextual understanding and accuracy. Additionally, it integrates language models and dictionaries to correct recognition errors and interpret ambiguous text. Finally, the recognized text is output in formats such as plain text or searchable PDFs. Tesseract's architecture is further enhanced by its support for training custom models, making it adaptable for various languages and fonts. With its open-source, flexible design, and the introduction of neural network capabilities in version 4.0, Tesseract remains one of the most robust and widely used OCR engines.

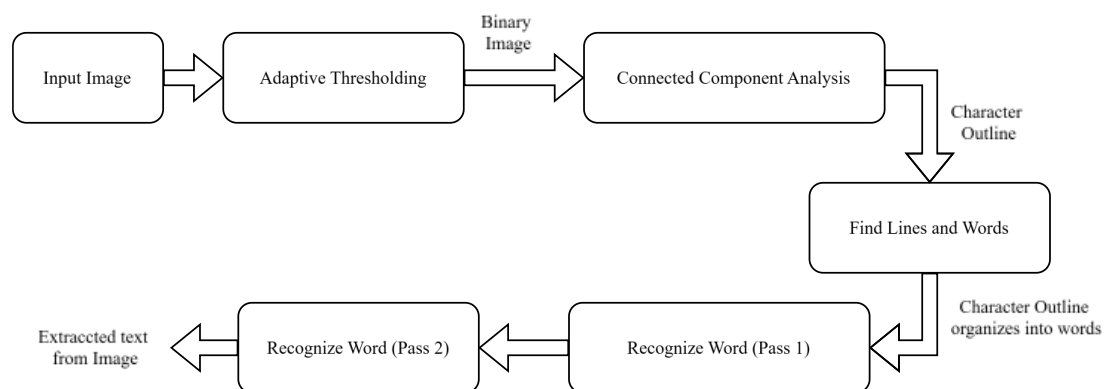


Figure 4-8: Tesseract OCR Working Procedure/ Architecture

This diagram represents the Tesseract OCR Working Procedure/Architecture, which outlines the steps involved in extracting text from an image using the Tesseract OCR engine.

Input Image: The process starts with an image containing text. This image can be a scanned document, photo, or any digital image.

Adaptive Thresholding: Adaptive thresholding involves converting a grayscale image into a binary image. The local threshold $T(x,y)$ is calculated for each pixel based on the

mean or weighted mean of the surrounding neighborhood.

$$I_{binary}(x, y) = \begin{cases} 0, & \text{if } I_{grayscale}(x, y) < T(x, y) \\ 255, & \text{if } I_{grayscale}(x, y) \geq T(x, y) \end{cases} \quad 4-16$$

Here:

- $I_{binary}(x, y)$: Binary pixel value.
- $I_{grayscale}(x, y)$: Grayscale pixel value.
- $T(x, y)$: Threshold for pixel (x, y) , often the mean of surrounding pixels:

$$T(x, y) = \frac{1}{N} \sum_{i=1}^N I_{grayscale}(x_i, y_i) \quad 4-17$$

- N : Number of pixels in the local neighborhood

A binary image where text pixels are typically white, and background pixels are black is obtained from this process.

Connected Component Analysis (CCA): It identifies connected regions in the binary image, which are potential text components (e.g., characters or parts of characters). It labels connected regions and group pixels belonging to the same character outline. A set of character outlines representing potential text are obtained.

Find Lines and Words: It organizes the detected character outlines into meaningful structures, such as lines and words. It aligns character outlines horizontally to form text lines and group characters within a line based on spacing to form words. A organized text region containing lines and words is obtained.

Horizontal alignment of bounding boxes B_i is determined:

$$B_{i+1}(x_1) - B_i(x_2) < threshold \quad 4-18$$

- $B_i(x_1)$ and $B_i(x_2)$: Start and end coordinates of bounding boxes B_i
- The threshold depends on the spacing between characters.

$$B_i(y_1) - B_{i+1}(y_1) < \epsilon \quad 4-19$$

- ϵ : Small allowable misalignment.

Recognize Words (Pass 1): In this step, the first pass of text recognition is done. It uses a pre-trained model to match character outlines to known patterns. It recognizes words based on character sequences. An initial recognized text with potential errors is obtained.

Recognize Words (Pass 2): The process refines the results from the first pass to improve accuracy. It recheck ambiguous words using language models and dictionaries and apply contextual analysis to refine word recognition. The Corrected and finalized recognized text is obtained.

Extracted Text: The final output is the extracted text from the image, ready for further use or processing.

This architecture highlights Tesseract's focus on efficiency and accuracy through a combination of image preprocessing, character segmentation, and contextual recognition. Tesseract is used to extract text that is outside the detected table regions. Before using Tesseract, the image undergoes a masking process to exclude the table regions, ensuring that only the text outside the tables is considered. The masking process involves creating a mask with the same dimensions as the image, initialized to true. For each detected bounding box of the table, the mask is updated to exclude the table region. The masked image is then processed further. Several preprocessing steps are performed on the masked image: converting it to grayscale, adjusting contrast and brightness, applying histogram equalization, median blurring, binarization using Otsu's method, and morphological operations to separate characters and reduce noise. Tesseract is configured with custom parameters to enhance text recognition accuracy, specifying the language and character set to recognize. The preprocessed and masked image is then passed through Tesseract to extract text data. The extracted text data is further processed to organize it into an xlsx file. Then, the xlsx file from Paddle OCR and Tesseract are merged into a single xlsx file.

4.5 SQLite to Store Result in Database

SQLite is a lightweight, serverless, and self-contained relational database management system (RDBMS) that stores data in a single file. Its simplicity and efficiency make it ideal for embedded systems, mobile apps, and small-scale projects. SQLite follows a relational database model, which organizes data into tables, rows, and columns. SQLite's operation revolves around three primary components:

- i. SQL Parser: Interprets and validates SQL commands.
- ii. Virtual Database Engine (VDBE): Executes SQL commands after parsing.
- iii. B-Tree Storage Engine: Manages tables and indexes, ensuring data is efficiently stored and retrieved.

SQLite guarantees the ACID property. SQLite executes SQL queries in the following stages:

- i. Parsing: The SQL parser verifies the syntax and checks for errors in the query. It generates an Abstract Syntax Tree (AST) to represent the query.
- ii. Query Optimization: The query planner optimizes the execution plan by determining the best way to retrieve or manipulate data. For example: Use indexes for faster lookups. Avoid full table scans where unnecessary.
- iii. Execution: The Virtual Database Engine (VDBE) executes the query plan. For SELECT queries, SQLite retrieves data from tables or indexes and filters it based on conditions. For INSERT/UPDATE/DELETE queries, SQLite modifies the database file.

SQLite's simplicity and efficiency make it an excellent choice for local databases in mobile and embedded systems. Its working revolves around straightforward file-based storage, query execution, and robust ACID compliance, ensuring reliability. To store the results of OCR in a database, Django ORM is used in conjunction with a SQLite database, providing an efficient and structured approach to managing the extracted data. This process involves several key steps, each of which is crucial to ensure the smooth handling and storage of OCR outputs. First, a Django project is set up, where the framework provides the foundation for building the application. During the setup, the database configuration is updated to use SQLite, a lightweight and easy-to-use database

that integrates seamlessly with Django. The database settings in the *settings.py* file are configured accordingly, specifying SQLite as the database engine. Next, Django models are defined to represent the structure of the OCR results we intend to store. These models serve as a blueprint for the database tables, specifying the fields and data types required. Each field corresponds to a specific piece of information extracted from the marksheet images, such as student names, roll numbers, subjects, and marks. Once the models are designed, the necessary migrations are created and applied to ensure the database schema aligns with the defined models.

Following the database setup, text extraction from the marksheet images is performed using OCR technology. This involves processing the images to recognize and extract textual data, which is often unstructured at this stage. The extracted text is then processed further to ensure accuracy and consistency. It is converted into structured data that matches the fields specified in the Django models. For instance, raw OCR output might need cleaning, formatting, or validation to align with the database schema. Finally, the processed data is saved into the SQLite database using Django ORM. The ORM abstracts the database interactions, allowing use of Python code instead of writing raw SQL queries. This makes the process more intuitive and reduces the likelihood of errors. By running the necessary migrations beforehand, the database is prepared to store the structured OCR results effectively. This end-to-end process facilitates the seamless integration of OCR data with a database, enabling efficient storage, retrieval, and analysis of the extracted information.

4.6 User Interface

The user interface (UI) of the mark sheet digitization system is designed to make the process smooth and user-friendly. It focuses on essential features to help users upload marksheets, extract data, and generate reports with ease. The system includes secure login and role-based access to ensure only authorized users can access specific features, with permissions customized for admins and regular users. It simplifies the handling of marksheet images by automatically extracting data and displaying it in a clear format. To maintain accuracy, the UI allows users to review the extracted data, fix any errors, and validate information before saving. It also includes tools for creating different types of reports as needed by campuses. By combining simplicity with functionality, the system ensures users can complete their tasks efficiently and accurately.

User Interaction contains various modules as required which are listed below.

4.6.1 Home Page

It contains Student/Admin Login Page. This page is used to perform the login authentication process by the administrator and students. After completion of successful login, administrator's main activity form is loaded for administrator and for students. result query activity form is displayed.

4.6.2 Admin Dashboard

The Admin Dashboard for the Django-based marksheet digitization project serves as the central hub for administrators to manage and analyze student data. The dashboard features three main functionalities: uploading marksheets and performing result analysis. Each functionality is accessible via dedicated buttons that navigate to respective pages.

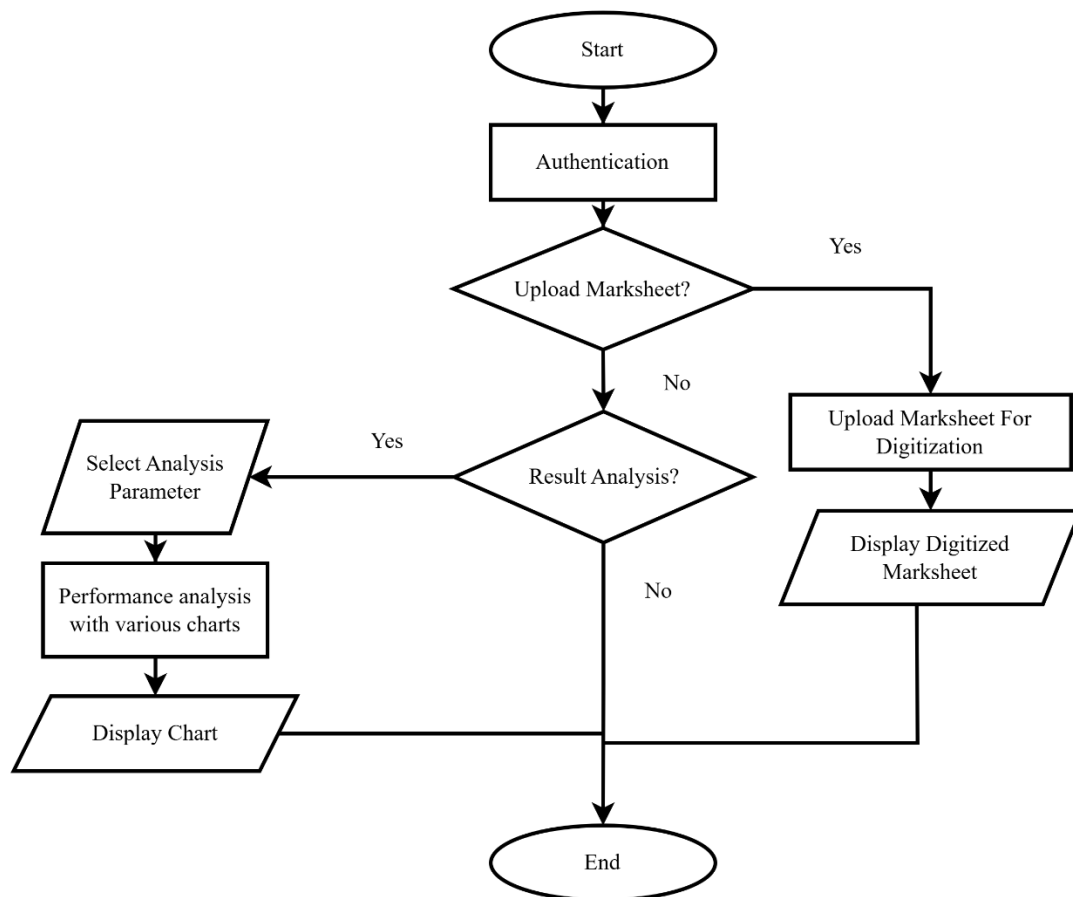


Figure 4-9: Flowchart for Admin Dashboard

4.6.2.1 Uploading Marksheet

When an admin chooses to upload a marksheet, they are redirected to an upload page where they can select an image file. The image file, typically a scanned marksheet, is then uploaded to the server and saved in the database. The backend extracts text and table in excel file from the uploaded image. This extracted excel file is subsequently displayed to the admin on the same page, facilitating immediate review and validation.

On this section, data extracted is visible and the administrator can validate the result based on hardcopy. The data then goes into the database.

4.6.2.2 Performance Analysis

When an admin chooses to result analysis , they are redirected to an analysis page where they can select on what basis they want to analyze the results. The admin is allowed to analyze the academic performance of the students. The administration can obtain

required information about student such as number of pass/fail student, percentage of each student, ranks and so on. This dashboard have various modules to analyze the performance. Some of them are:

- **Rank of a class**

This metric evaluates the performance of students in a specific class by assigning ranks based on their total marks or percentage obtained. The rank is calculated by sorting students in descending order of their scores, allowing comparison within the class.

- **Subject wise pass Ratio**

This ratio indicates the percentage of students who successfully passed a particular subject. It serves as a useful tool for identifying subjects that may be more difficult for students, highlighting areas where additional support or resources could be beneficial. By analyzing this ratio, educators can pinpoint subjects that may require changes in teaching strategies or extra help to improve student success.

- **Programme wise Pass Ratio**

This ratio represents the percentage of students who pass within a specific program, such as Computer Engineering or Electronics and Communication Engineering. It provides a basis for comparing student performance across different programs, helping to evaluate their overall effectiveness. By analyzing this metric, educators can identify strengths and areas needing improvement within each program.

- **Overall pass/fail Ratio**

This metric provides a comprehensive view of the total number of students passing or failing in a semester or exam session. It is a high-level indicator of the institution's overall academic success rate.

- **Comparison of department wise results**

This compares the overall performance of different departments, often through average pass percentages or ranks. It helps in evaluating the relative success of academic programs and identifying areas for improvement or investment.

- **Comparison of Assessment and Final marks**

This analysis compares the overall average of assessment and final marks for a specific subject, providing a brief overview of the differences between internal and final marks. It highlights any variations or discrepancies that may exist between the two, offering insight into how the assessment scores align with the final evaluation. This can help identify areas where improvements or adjustments might be needed in the grading process.

- **Programme wise Passing Trend**

This metric tracks the performance trends of students in selected programs over time. It highlights changes in pass percentages, providing insights into academic progress and consistency. By analyzing these trends, educators can identify patterns of improvement or decline, helping to assess the effectiveness of teaching strategies and curriculum adjustments.

- **Distribution of Grades**

This metric represents the distribution of students based on specific mark intervals using graphical visualization. It illustrates how many students fall within each range of scores, making it easier to analyze performance patterns. By examining these distributions, educators can identify trends, such as the concentration of students in certain score ranges, helping to assess overall academic achievement.

- **Subject wise Passing Trends**

This analysis tracks student pass rates in specific subjects over time, highlighting changes in performance. It helps identify patterns, assess subject difficulty, and evaluate the effectiveness of teaching strategies.

- **Average marks of Subject**

This analysis calculates the average marks obtained by students in a selected subject, providing an overview of overall performance. It helps in understanding how well students are performing in the subject and identifying areas that may need improvement. By analyzing these averages, educators can assess the effectiveness of teaching methods and make data-driven decisions to enhance learning outcomes.

4.6.3 Student Dashboard

The Admin Dashboard for the Django-based marksheet digitization project serves to view the results by the students. It requires login authentication process for the students. From login credential the roll number (username) is taken, from which marksheet is shown to user extracting the stored data on the database. Here students must enter which semester marksheet they wanted to view. It allows to view / print /download the marks.

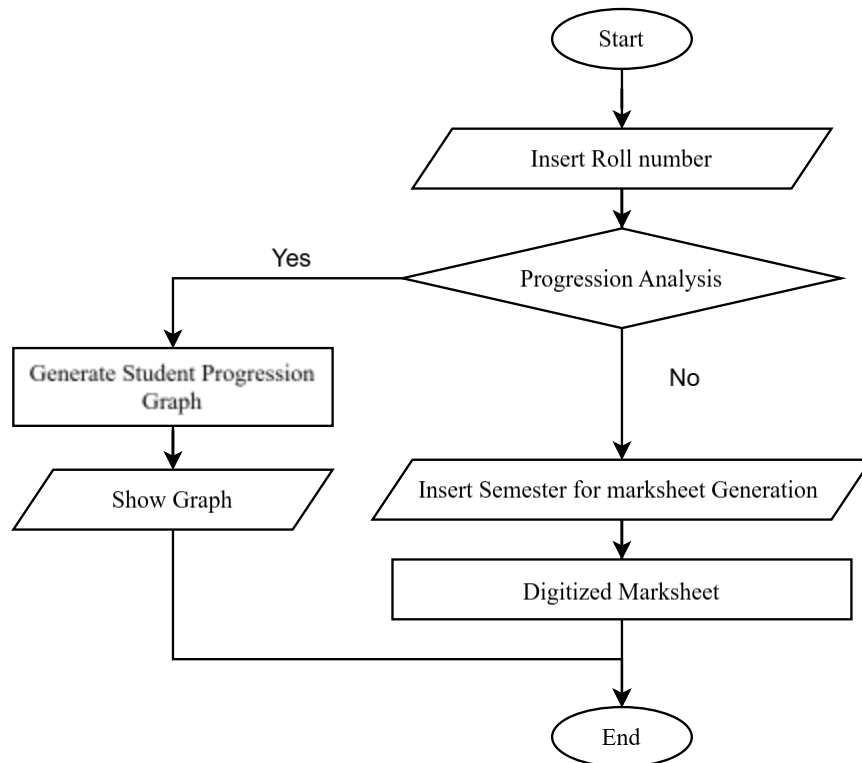


Figure 4-10: Flowchart for Student Dashboard

Individual Performance of students

Using the roll number, a student's individual performance across all exams can be analyzed in detail. It identifies trends in performance, such as improvements or declines, by comparing scores across semesters or years. Visual representations, like graphs, highlight strengths and weaknesses, while rankings and comparisons with class averages provide additional context. This analysis enables a comprehensive view of academic progress and areas for improvement.

5. IMPLEMENTATION DETAILS

5.1 Integrating various image preprocessing steps

To integrate various image preprocessing steps efficiently, we structured our workflow as a pipeline where each step applied a transformation to the image sequentially. By applying and integrating these preprocessing techniques, OCR performance can be significantly enhanced, leading to more effective digitization of marksheets.

5.2 Integrating YOLOv8s for table detection

Integrating YOLOv8s (from Ultralytics) for table detection involved multiple steps, including setting up the model, preprocessing the input image, running inference, and post-processing the results. YOLOv8s is a smaller and faster variant of YOLOv8, ideal for real-time applications. For table detection, we used a pre-trained version. By combining YOLOv8s with OpenCV, we set up an efficient table detection pipeline.

5.3 Integrating tesseract-OCR and paddle-OCR

Integrating Tesseract OCR and PaddleOCR allows you to leverage the strengths of both OCR engines in a single pipeline. Tesseract is great for quick and simple text recognition, while PaddleOCR excels in complex, multilingual, or high-accuracy tasks. Combining them can produce robust results. Tesseract is used for generating the text outside of boundary box and PaddleOCR is used to generate the text inside the table in this project. At last both OCR results are combined and a separate excel file is generated.

5.4 Integrating obtained excel file into the database

To integrate an Excel file into the database, we developed a robust database schema and established a seamless connection between the Excel file and the SQLite database using Django ORM. The data is systematically extracted, transformed, and loaded into the database, ensuring it aligns perfectly with the schema's structure and requirements. This process guarantees data integrity, consistency, and optimal performance.

5.5 Integration in user interface

The UI acts as the front-facing layer, allowing users to interact effortlessly while the backend handles the core processing, database management, and system logic. When users upload marksheets through the UI, the backend processes these files using Optical Character Recognition (OCR) to extract data. This data is then validated against predefined rules and stored in a secure database. For role-based access, the backend manages authentication and authorization processes, ensuring that users (students or administrators) can only access features permitted for their roles.

5.6 Database Design

The database is designed to store detailed information about students and their marks in various subjects using multiple interconnected tables. This data will be used for analyzing different parameters related to student performance. Additionally, the database will generate marksheets that students can view. The interconnected tables ensure that data is efficiently organized and easily accessible for both analysis and reporting purposes. The table design, along with the ER diagram, is illustrated below.

5.6.1 Table Information

Table Name: Student Table

Primary Key: Student_id

Table 5-1: Student Table

| Column Name | Data Type | Description |
|--------------|-----------|---------------------|
| Student_id | Varchar | Student Roll Number |
| Name | Varchar | Student Name |
| Campus | Varchar | Campus Name |
| Exam_roll_no | Int | Exam Roll Number |
| Level | Varchar | Level |
| Programme | Varchar | Faculty |

Table Name: Subject Table

Primary Key: Title

Table 5-2: Subject Table

| Column Name | Data Type | Description |
|------------------|-----------|-----------------------|
| Subject_Code | Varchar | Subject Code |
| Title | Varchar | Subject Name |
| Full_marks_final | Int | Full Marks Final |
| Full_marks_ass | Int | Full Marks Assessment |
| Pass_marks_final | Int | Pass Marks Final |
| Pass_marks_ass | Int | Pass Marks Assessment |

Table Name: Marks Obtained table

Foreign Key: Student_id, Title

Table 5-3: Marks Obtained Table

| Column Name | Data Type | Description |
|------------------|-----------|------------------|
| Student_id | Varchar | Exam Roll Number |
| Year_Part | Varchar | Year/Part |
| Title | Varchar | Subject title |
| Marks_assessment | Int | Assessment Marks |
| Marks_theory | Int | Theory Marks |
| Total_marks | Int | Total_marks |

5.6.2 Explanation of Relationship between Tables

- **Student To Marks**
 - ➔ One-to-Many (A student can have multiple marks entries)
- **Marks To Student**
 - ➔ Many-to-One (Each marks entry belongs to one student)
- **Subject To Marks**
 - ➔ One-to-Many (Each subject can appear in multiple marks entries)
- **Marks To Subject**
 - ➔ Many-to-One (Each marks entry is associated with one subject)

Explanation:

The Marks table stores individual marks entries, linking each entry to a student (Std_Rollno) and a subject (Subject_title). This allows the system to store and retrieve marks for all students across various subjects. Handling Multiple Semesters: The Marks table includes a Semester column to differentiate between different semesters. Aggregating Total Marks. By structuring the database this way, it will effectively store and organize marks for all students in a class, across different subjects and semesters.

5.6.3 Entity Relationship Diagram

The given ER (Entity-Relationship) diagram describes the relationships and attributes

of three entities: StudentInformation, MarksObtained, and SubjectInformation.

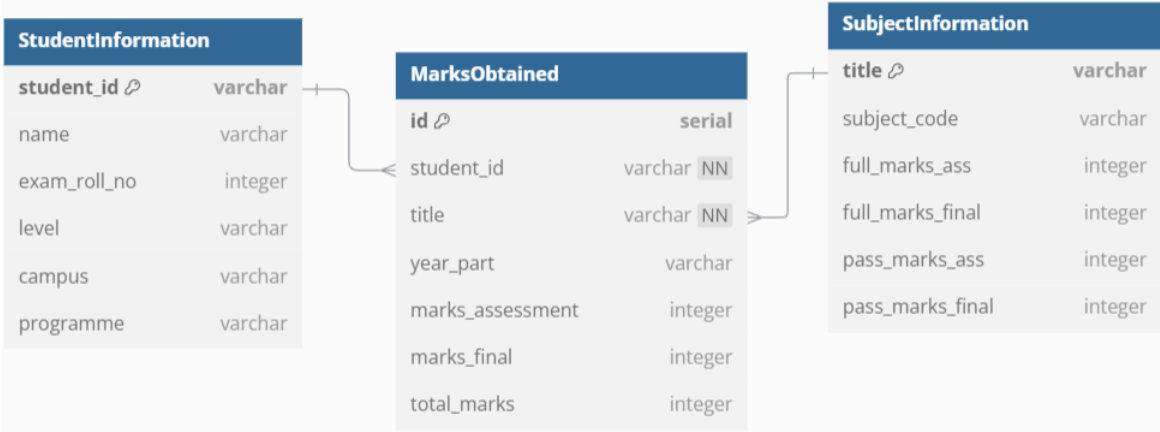


Figure 5-1: ER Diagram

5.7 Use Case Diagram

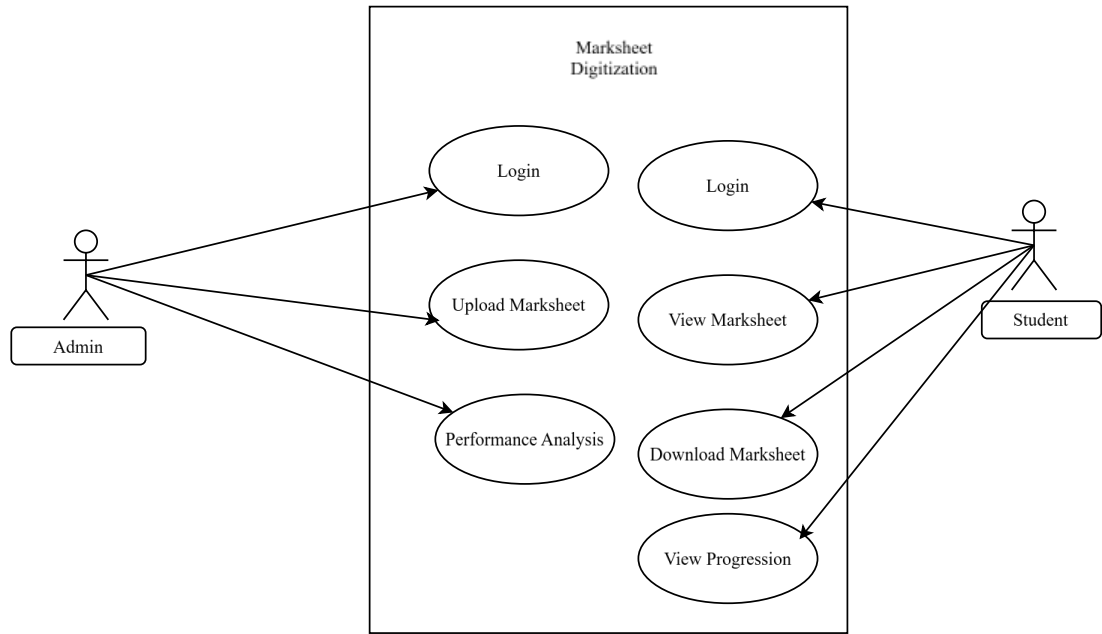


Figure 5-2: Use Case Diagram

This use case diagram represents the Marksheet Digitization using OCR and report generation for QAA and depicts interactions between two primary actors: Admin and

Student, and their corresponding functionalities.

i. Actors:

- Admin: Responsible for managing marksheets and system users.
- Student: Accesses personal marksheet-related features.

ii. Use Cases:

- Login: Both Admin and Student have a login use case to authenticate themselves.
- Upload Marksheet (Admin only): Admin can upload marksheets to the system for digitization.
- View Marksheet (Student only): Students can view their marksheets uploaded by the admin.
- Download Marksheet (Student only): Students can download their marksheets for offline access.
- View Progression (Student only) : Student Can analyze their individual performance through graphs.
- Performance Analysis (Admin only): Admin can analyze the performance data of students, likely based on the marksheets. .
- Relationships: Both Admin and Student have a Login relationship with the system.

The Admin is associated with all the administrative functionalities (Upload Marksheet, Performance Analysis). The Student is associated with marksheet-specific functionalities (View Marksheet, Download Marksheet, view his/her progression).

5.8 Sequence Diagram

This diagram illustrates the interaction between Admin, Student, the System, and external components (SQLite database and PyTesseract & PaddleOCR) within the Marksheet Digitization with OCR and Report Generation For QAA System. It captures the processes involved in managing and accessing marksheets.

Actors and Components:

- i. **Admin & Student** - Users interacting with the system.
- ii. **System** - The main backend handling authentication, data processing, and retrieval.
- iii. **SQLiteDB** - Database storing marksheet and student performance data.
- iv. **PyTesseract & PaddleOCR** - Optical Character Recognition (OCR) tools used to extract text from marksheets.

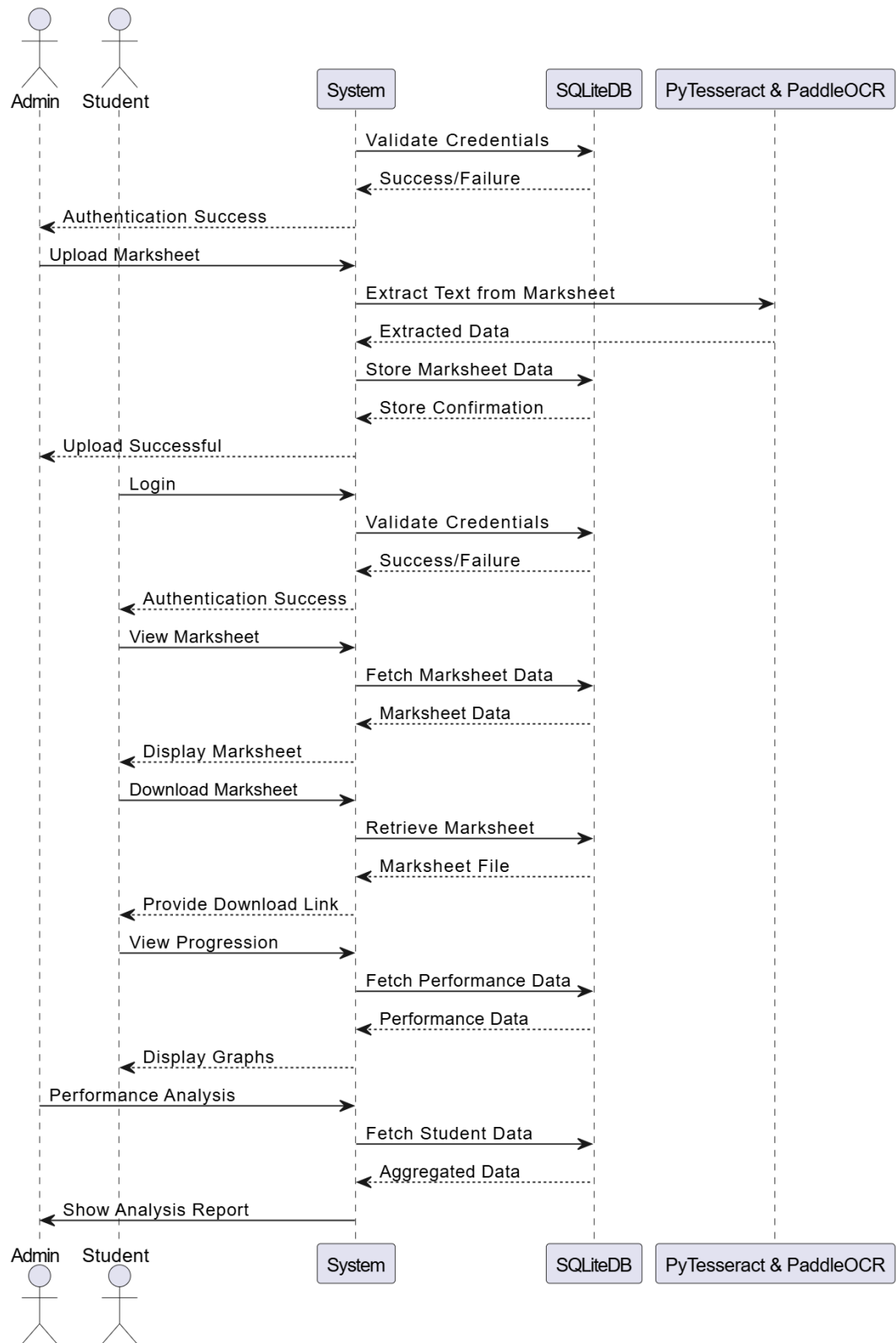


Figure 5-3: Sequence Diagram

Workflow:

The Admin starts by logging into the system. Their credentials are validated against the SQLiteDB. If the authentication is successful, the Admin proceeds to upload a marksheet. The system then sends this marksheet to PyTesseract & PaddleOCR, which extracts the text from the document. Once the text is extracted, it is returned to the system, which then stores the marksheet data in SQLiteDB. After a successful upload, the Admin receives a confirmation message.

The Student also begins by logging into the system. Their credentials are verified using SQLiteDB. If the authentication is successful, the student can request to view their marksheet. The system retrieves the marksheet data from SQLiteDB and displays it to the student.

If a Student wants to download their marksheet, they can request it from the system. The system fetches the corresponding marksheet file from SQLiteDB and provides a download link, allowing the student to save a copy of their marksheet.

For students interested in tracking their academic performance, the system offers a progression view feature. When a student requests this, the system retrieves their performance data from SQLiteDB. The data is then processed and displayed as graphs, helping the student visualize their academic trends over time.

The Admin has access to performance analysis tools to evaluate the academic progress of multiple students. When the Admin initiates this analysis, the system fetches student data from SQLiteDB. The retrieved data is aggregated and processed to generate a detailed analysis report, which provides insights into overall student performance.

5.9 DFD Diagram

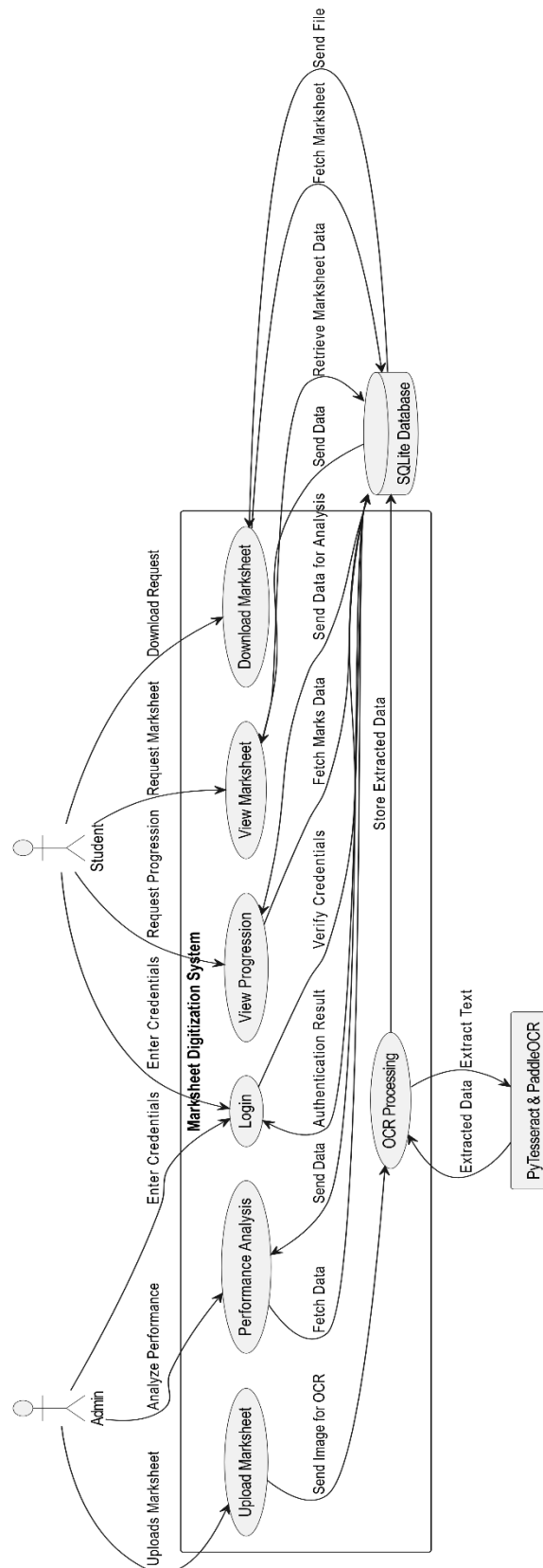


Figure 5-4: Data Flow Diagram

The DFD diagram illustrates the Marksheet Digitization System, showing how data flows between admins, students, and system processes. Admins upload marksheets, which undergo OCR processing to extract text, storing it in an SQLite database. Students can view their marksheets, track academic progression, and download results, while admins can analyze performance using extracted data. The system ensures secure login authentication and smooth data retrieval, making marksheet management efficient and digitalized.

Level 0 (Context Diagram)

Actors:

- i. Admin
- ii. Student

System: "Marksheet Digitization using OCR and Report Generation for QAA"

Data Flows:

- Users (Admin & Student) → System Processes
- System Processes → OCR Processing
- OCR Processing → SQLite Database
- SQLite Database → System Processes (Fetching Data, Retrieving Marksheet, Sending Files, etc.)

Level 1 (Expanded View)

A. External Entities (Actors)

a) Admin

- Uploads marksheets
- Analyzes performance
- Enters login credentials

b) Student

- Enters login credentials
- Requests marksheets
- Requests academic progression
- Downloads marksheets

B. Processes (Oval Shapes)

a) Login

- Accepts credentials from admins and students.
- Verifies credentials and sends authentication results.

b) Upload Marksheet (Admin)

- Takes uploaded marksheet.
- Sends image for OCR processing.

c) OCR Processing

- Uses PyTesseract & PaddleOCR to extract text from uploaded marksheets.
- Extracted data is stored in the SQLite Database.

d) Performance Analysis (Admin)

- Fetches extracted data from the database.
- Analyzes student performance.

e) View Progression (Student)

- Requests for progression data.

f) View Marksheet (Student)

- Requests a specific marksheet.
- Retrieves the marksheet from the database.
- Requests to download marksheets.

C. Data Stores (Cylinder Shapes)

a) SQLite Database

- Stores extracted marksheet data.
- Retrieves marks data for viewing and downloading.
- Sends data for analysis.

D. Data Flow (Arrows)

The diagram shows data movement between:

- Admin → Provide login credentials → System
- System → Validate login → Database
- Admin → Upload marksheet → System
- System → Extract text → OCR Tools (PyTesseract & PaddleOCR) → Store marks → Database
- Admin → Request Performance Analysis → System → Fetch Data → Database

- Generate Graphs → Admin
- vi. Student → Provide roll number and year → System → Fetch marksheet → Database → Show marksheet → Student
- vii. Student → Request marksheet download → System → Provide downloadable file → Student
- viii. Student → Provide roll number and year → Request Progression Graph → System → Fetch marksheet → Database → Show Progression Graph → Student

5.10 Activity Diagram

Activities Involved:

- Admin uploads a marksheet (OCR extraction and storage).
- Student logs in and views/downloads marksheet.
- Student views performance progression.
- Admin performs performance analysis.

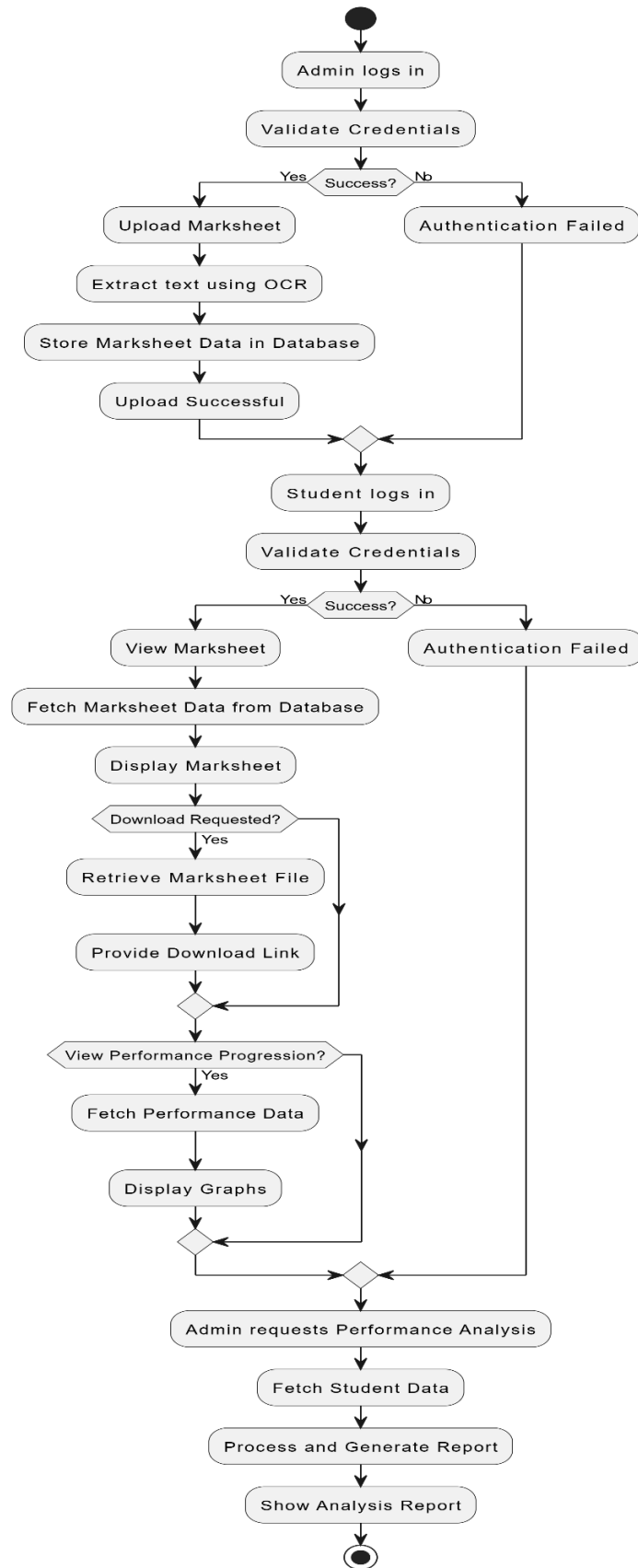


Figure 5-5: Activity Diagram

This activity diagram shows how a Marksheet Digitization System works for both admins and students. First, the admin logs in and uploads a marksheet, which is processed using OCR to extract text. The system then stores the data in a database. If the admin enters the wrong credentials, they cannot log in. Next, a student logs in to view their marksheet. If their login is successful, the system fetches the marksheet data from the database and displays it. Students can also download their marksheet if needed.

If a student wants to track their performance progression, the system fetches performance data and shows graphs. Additionally, the admin can request a performance analysis of students. The system then processes student data and generates a report showing overall performance. This diagram outlines the steps involved in managing and accessing student marksheets efficiently.

6. RESULTS AND ANALYSIS

Below are the detailed results of each step involved in text and table detection and extraction as well as Images of user interface that have been completed.

6.1 User interfaces

This is the start of User interface(Home window).

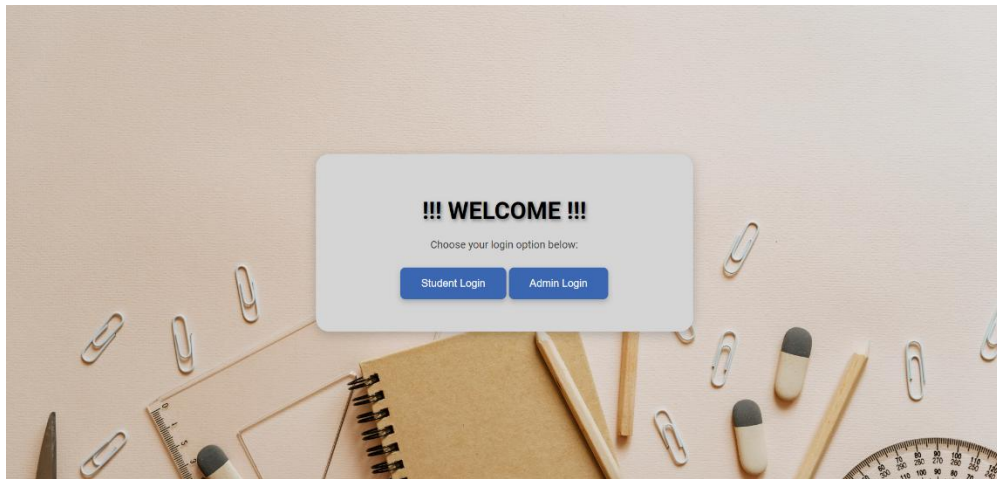


Figure 6-1: Home Page

When a user clicks on student it redirects to the student login section. When the student enters the Roll Number (Student ID) it redirects to Student dashboard where he/she enters the Year/Part for the marksheets of the desired semester.

6.1.1 Student Dashboard

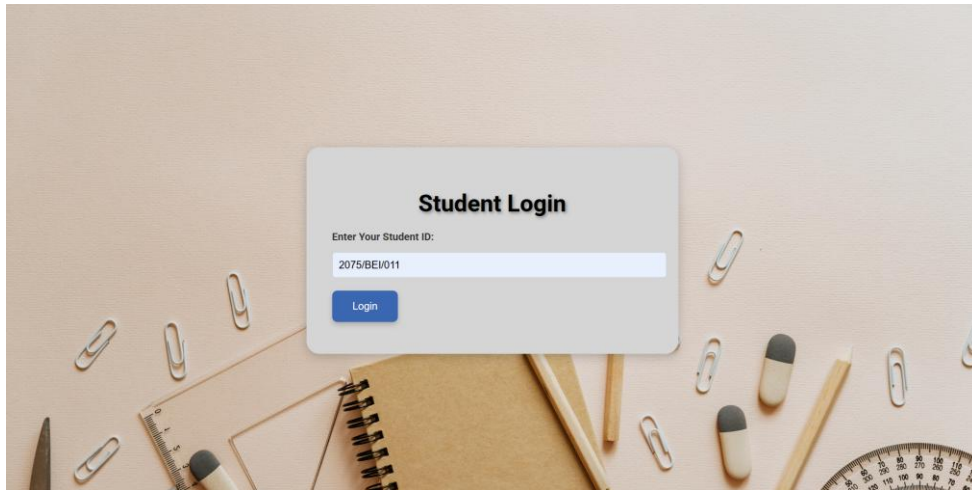


Figure 6-2: Student Login

The image shows a student login interface with a clean and minimal design. Students can use their roll number to login to the page after which they will be redirected to the webpage below.

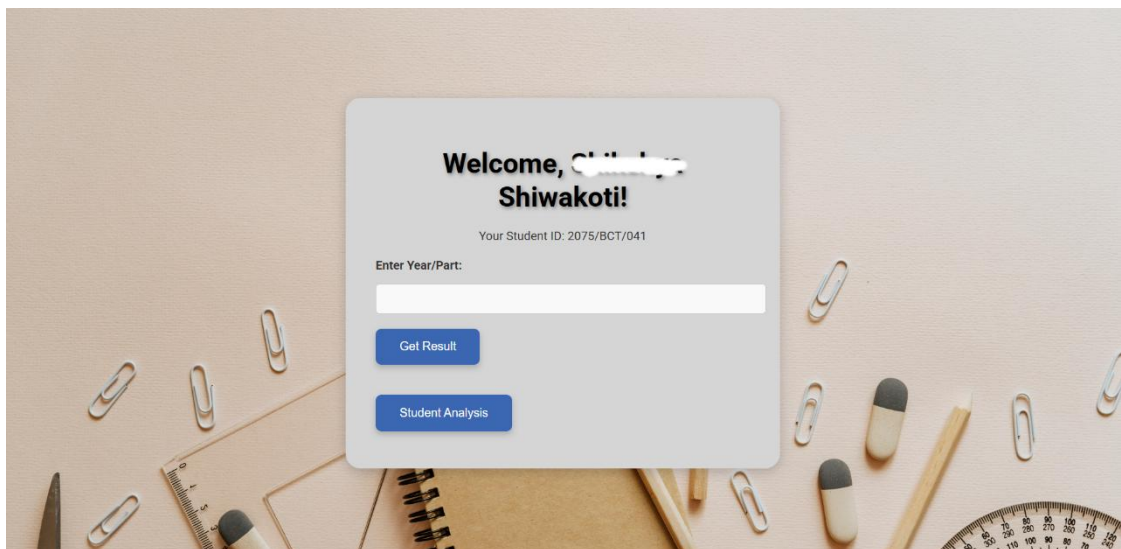


Figure 6-3: Student Dashboard

6.1.1.1 Get Result

Now when the students enters the semester of which he/she wants to generate result and click on get result button, the result is shown in website of entered semester from which

he/she can generate the pdf for their use. The image below shows the result of the student shown on the webpage.

Results for [REDACTED] Shiwakoti (ID: 2075/BCT/041)

Programme: Computer Engineering
Year/Part: IV/I

| Subject Title | Full Marks Ass. | Full Marks Final | Pass Marks Ass. | Pass Marks Final | Marks Ass. | Marks Final | Total Marks |
|--|-----------------|------------------|-----------------|------------------|------------|-------------|-------------|
| Distributed System PRACTICAL | 25 | 0 | 10 | 0 | 22.0 | | 22.0 |
| Digital Signal Analysis & Processing | 20 | 80 | 8 | 32 | 17.0 | 67 | 84.0 |
| Digital Signal Analysis & Processing PRACTICAL | 25 | 0 | 10 | 0 | 24.0 | | 24.0 |
| Project I PRACTICAL | 50 | 0 | 20 | 0 | 46.0 | | 46.0 |
| Energy Environment & Society | 10 | 40 | 4 | 16 | 9.0 | 23 | 32.0 |
| Organization & Management | 20 | 80 | 8 | 32 | 19.0 | 61 | 80.0 |
| Web Technologies and Applications (Elective I) | 20 | 80 | 8 | 32 | 17.0 | 62 | 79.0 |
| Web Technologies and Applications (Elective I) PRACTICAL | 25 | 0 | 10 | 0 | 23.0 | | 23.0 |

[Download as PDF](#)

Figure 6-4: Generated Result

Institute of Engineering, Thapathali Campus
Student name: [REDACTED] Shiwakoti
Student ID: 2075/BCT/041
Programme: Computer Engineering
Year/Part: IV/I

| Subject Title | Full Marks Ass. | Full Marks Final | Pass Marks Ass. | Pass Marks Final | Marks Ass. | Marks Final | Total Marks |
|--|-----------------|------------------|-----------------|------------------|------------|-------------|-------------|
| Project Management | 20 | 80 | 8 | 32 | 19.0 | 65 | 84.0 |
| Computer Network | 20 | 80 | 8 | 32 | 19.0 | 58 | 77.0 |
| Computer Network PRACTICAL | 50 | 0 | 20 | 0 | 46.0 | 0 | 46.0 |
| Distributed System | 20 | 80 | 8 | 32 | 19.0 | 46 | 65.0 |
| Distributed System PRACTICAL | 25 | 0 | 10 | 0 | 22.0 | | 22.0 |
| Digital Signal Analysis & Processing | 20 | 80 | 8 | 32 | 17.0 | 67 | 84.0 |
| Digital Signal Analysis & Processing PRACTICAL | 25 | 0 | 10 | 0 | 24.0 | | 24.0 |
| Project I PRACTICAL | 50 | 0 | 20 | 0 | 46.0 | | 46.0 |
| Energy Environment & Society | 10 | 40 | 4 | 16 | 9.0 | 23 | 32.0 |
| Organization & Management | 20 | 80 | 8 | 32 | 19.0 | 61 | 80.0 |
| Web Technologies and Applications (Elective I) | 20 | 80 | 8 | 32 | 17.0 | 62 | 79.0 |
| Web Technologies and Applications (Elective I) PRACTICAL | 25 | 0 | 10 | 0 | 23.0 | | 23.0 |

Figure 6-5: Result in PDF

The above image is the Generated result in a PDF which can be downloaded.

6.1.1.2 Student Analysis

Students can analyze their performance in examinations through the Student Analysis

button and use it to their advantage. The analysis is presented through a line graph, where the X-axis represents different academic year parts, while the Y-axis denotes the overall percentage, providing students with a brief insight into their academic progression. This enables them to identify their strengths and areas that need improvement. The chart below illustrates a student's academic performance, helping us understand the type of analysis conducted and how it can be utilized for academic growth.

Academic Progression Analysis

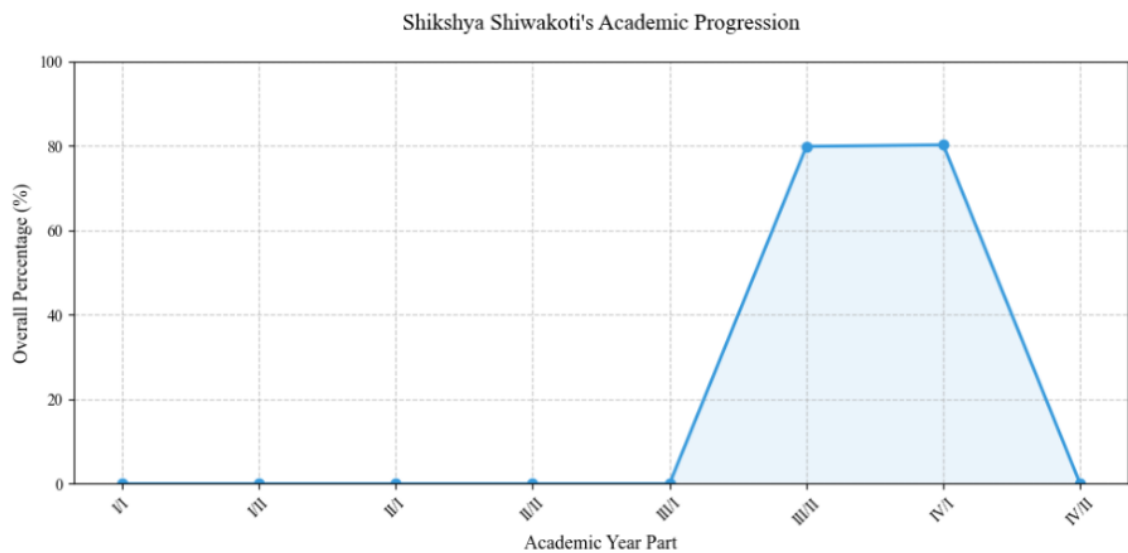


Figure 6-6: Student Progression

6.1.2 Admin Dashboard

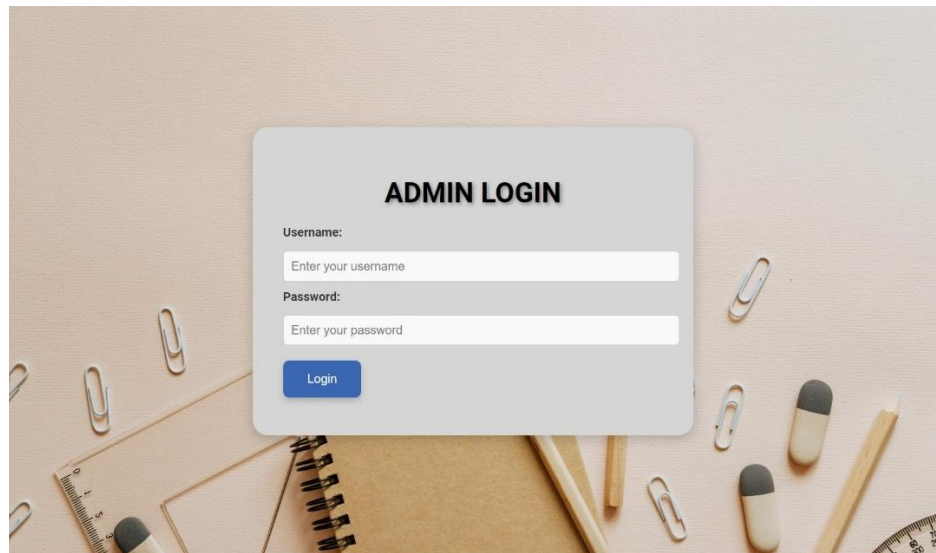


Figure 6-7: Admin Login

When the user clicks on Admin it redirects to Admin login section. When the provided credentials are correct it redirects to Admin dashboard which contains two buttons having their own features. When admin clicks on upload marksheet, it redirects to the page where admin can upload the marksheet.

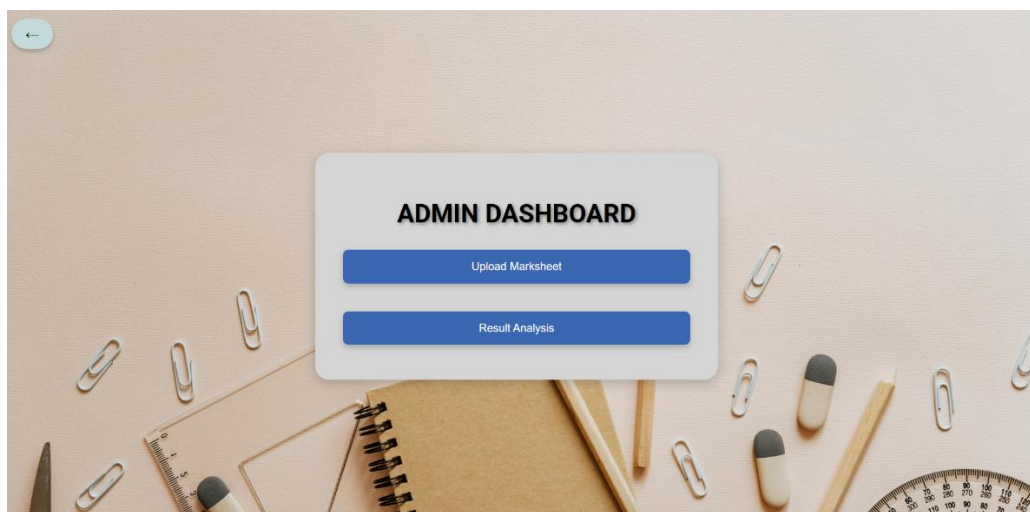


Figure 6-8: Admin Dashboard

6.1.2.1 Upload Marksheet

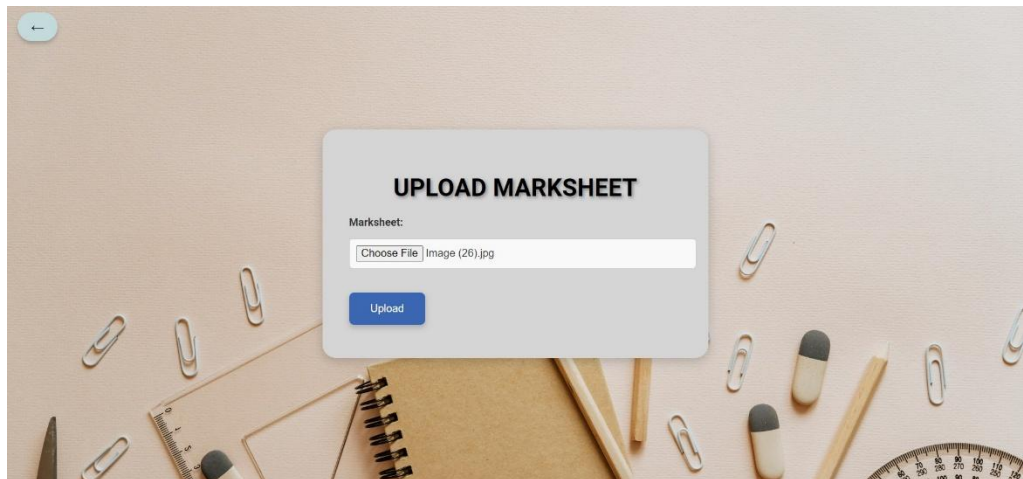


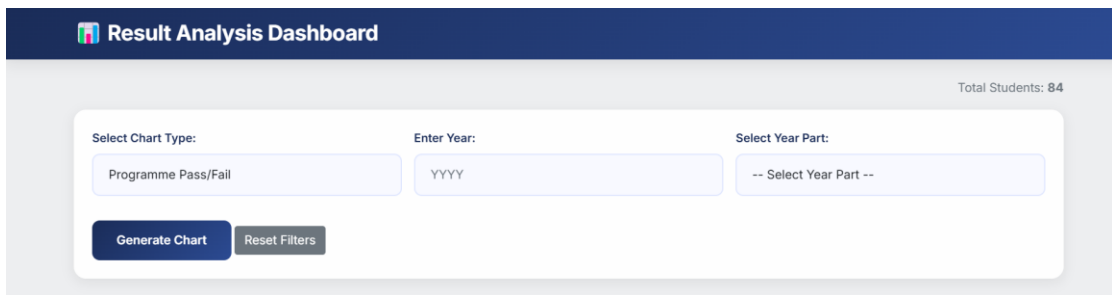
Figure 6-9: Upload Marksheet

Here the admin uploads the marksheet which undergoes text extraction and then data is stored into the database. The uploaded marksheet is also shown in the webpage.

| Marksheet | | | | |
|---|------------------------------|------------------|-------------|-------------|
| Name: VIKAS K | | | | |
| Exam Roll No: 58225 | | | | |
| Level: Bachelors in Engineering | | | | |
| Campus: Thapathali Campus | | | | |
| Programme: Electronics, Communication & Information | | | | |
| Subject Code | Subject Title | Marks Assessment | Marks Final | Total Marks |
| EE460 | Electric Circuits & Machines | 17.0 | A | - |

Figure 6-10: Marksheet Uploaded

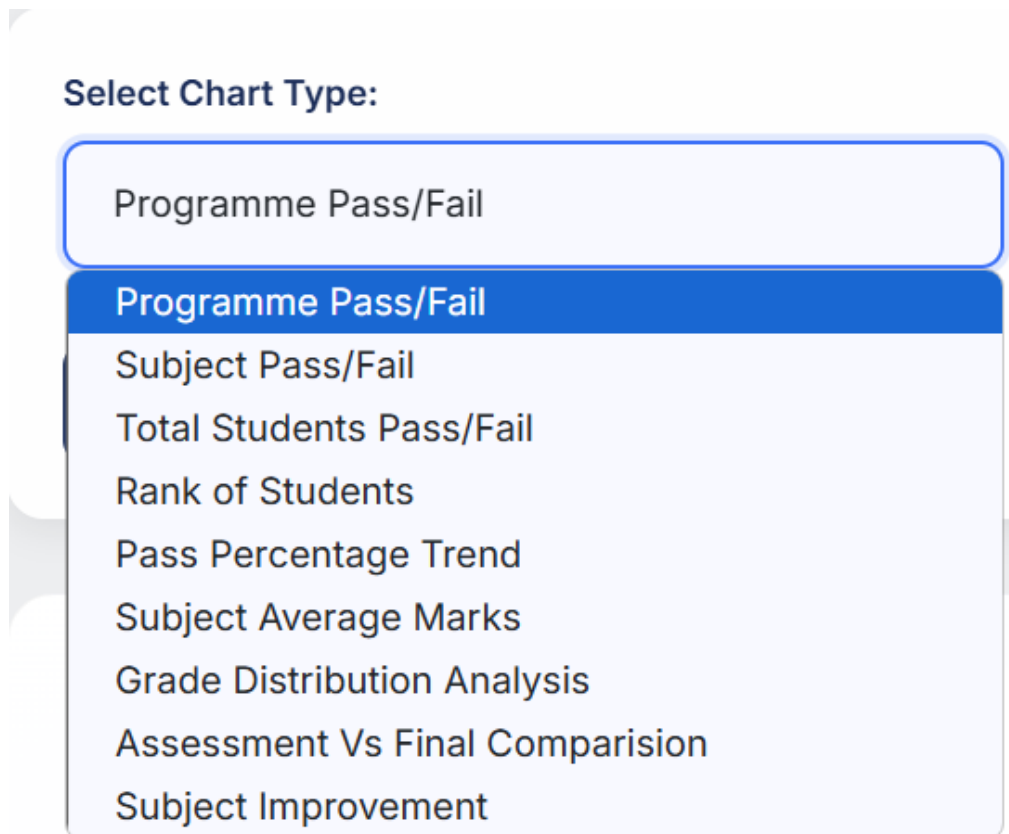
6.1.2.2 Result Analysis



The screenshot shows the 'Result Analysis Dashboard' with a dark blue header. In the top right corner, it says 'Total Students: 84'. Below the header, there are three input fields: 'Select Chart Type:' with a dropdown menu showing 'Programme Pass/Fail', 'Enter Year:' with a text input showing 'YYYY', and 'Select Year Part:' with a dropdown menu showing '-- Select Year Part --'. Below these fields are two buttons: 'Generate Chart' and 'Reset Filters'.

Figure 6-11: Result Analysis Webpage

“Result Analysis” button redirects to this webpage which is designed for analyzing student results. It displays the total number of students which is recorded in the database. Admin can select the desired analysis type (currently set to "Programme Pass/Fail"). Following are the charts which can be generated for desired analysis type.



The screenshot shows the 'Select Chart Type:' dropdown menu. The menu is open, displaying a list of options. The first option, 'Programme Pass/Fail', is highlighted with a blue background. The other options are: 'Subject Pass/Fail', 'Total Students Pass/Fail', 'Rank of Students', 'Pass Percentage Trend', 'Subject Average Marks', 'Grade Distribution Analysis', 'Assessment Vs Final Comparision', and 'Subject Improvement'.

Figure 6-12: Analysis Type Selection

Programme Pass/Fail

Once the options for chart type is selected, users can enter the required inputs (such as year, semester, subject and so on) and click "Generate Chart" button to create a visual representation of the chosen data.

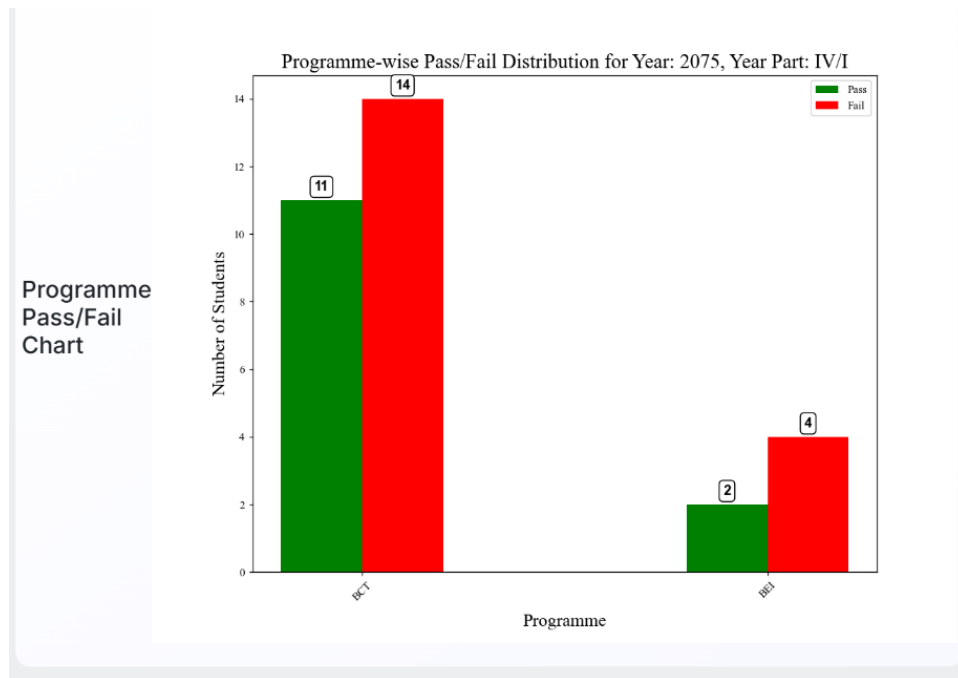


Figure 6-13: Programme wise Pass/Fail Bar Graph

The bar graph visually represents the number of students who passed and failed in different programmes for the specified year they are from, and part of the year (semester). The x-axis labels the programme, while the y-axis indicates the number of students. Green bars denote the students who passed, whereas red bars indicate those who failed. Out of total students registered in the database, The BEI program shows that 11 students passed, while 14 failed. Similarly, in the BCT program, only 2 students passed, whereas 4 failed. The chart visually emphasizes that the number of failing students is higher than the passing students in both programmes.

Subject Pass/Fail

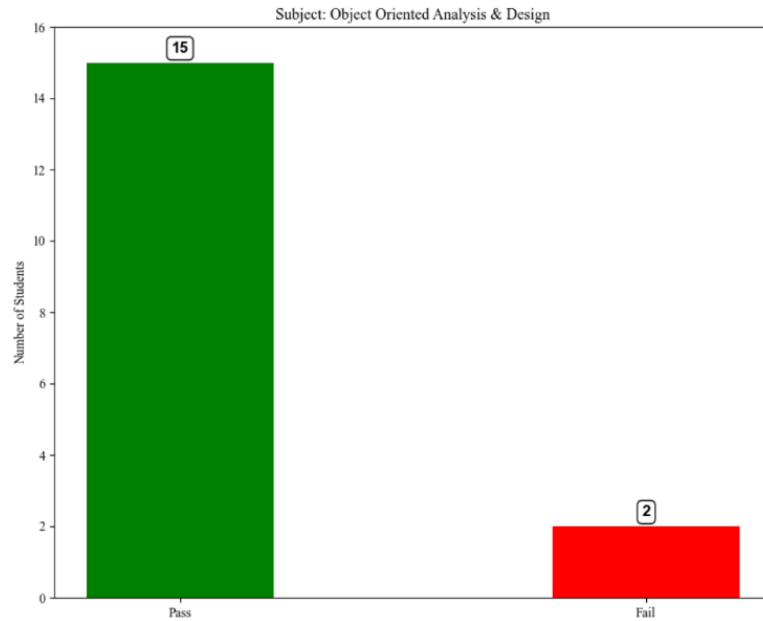


Figure 6-14: Subject wise Pass/Fail Bar Graph

The bar chart illustrates the pass/fail distribution for the subject "Object Oriented Analysis & Design." The x-axis represents the two categories: Pass and Fail, while the y-axis indicates the number of students. The green bar represents the students who passed, whereas the red bar represents those who failed. According to the chart, 15 students passed, while only 2 students failed, demonstrating a high pass rate for this subject. The significant difference between the two bars suggests that most students performed well in Object Oriented Analysis & Design, with only a few struggling to clear the subject.

Total Students Pass/Fail

Total Students for Year 2075, Year/Part IV/I: 31

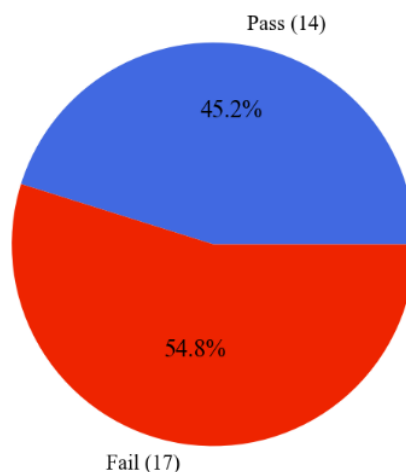


Figure 6-15: Semester wise Pass/Fail Percentage Pie-Chart

The pie chart in the image represents the total number of students who passed or failed in the year 2075 for Year/Part IV/I. The chart is divided into two segments: a blue section representing the students who passed and a red section representing those who failed. Out of a total of 31 students, 14 students (45.2%) passed, while 17 students (54.8%) failed. The chart visually conveys that the failure rate is slightly higher than the pass rate. The labels and percentages on the chart clearly indicate the distribution of results, making it easy to interpret.

Rank Of Students

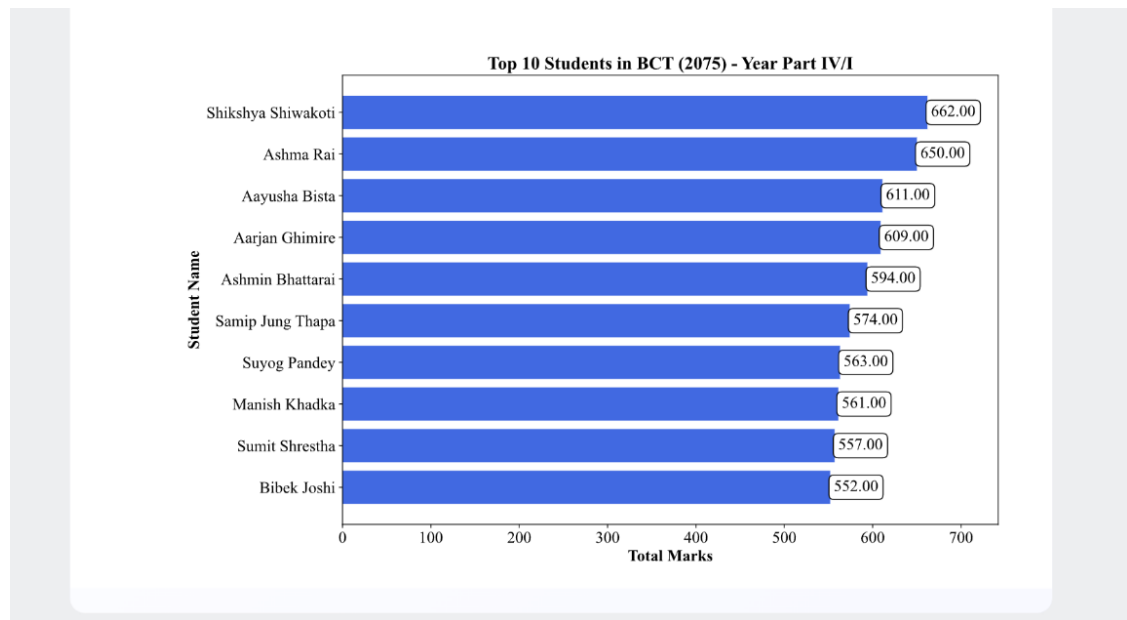


Figure 6-16: Rank of Students

The bar chart displays the top 10 students in the BCT (Bachelor in Computer Engineering) program for the year 2075, Year Part IV/I. The chart ranks students based on their total marks, with names listed on the y-axis and corresponding total marks on the x-axis. We can see that Shikshya Shiwakoti secured the highest marks with 662, followed closely by Ashma Rai with 650. The chart effectively visualizes student performance, with horizontal bars making it easy to compare marks. Each bar is labeled with the corresponding total marks for clarity.

Pass Percentage Trend

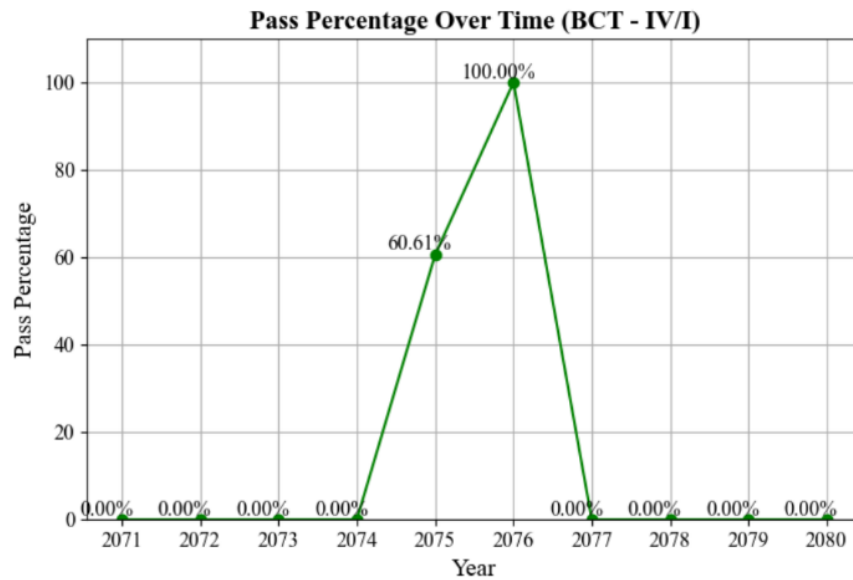


Figure 6-17: Pass Percentage Trend

The line chart illustrates the pass percentage over time for BCT (Bachelor in Computer Engineering) IV/I from the year 2071 to 2080. The x-axis represents the years, while the y-axis represents the pass percentage. We can see that in 2075, the pass percentage is 60.61%. The trend peaked in 2076, achieving a 100% pass rate. The pass percentage is 0% for most of the year due to not having the marksheets data stored in database. This chart indicates variations in examination patterns and academic performance with year.

Subject Average Marks

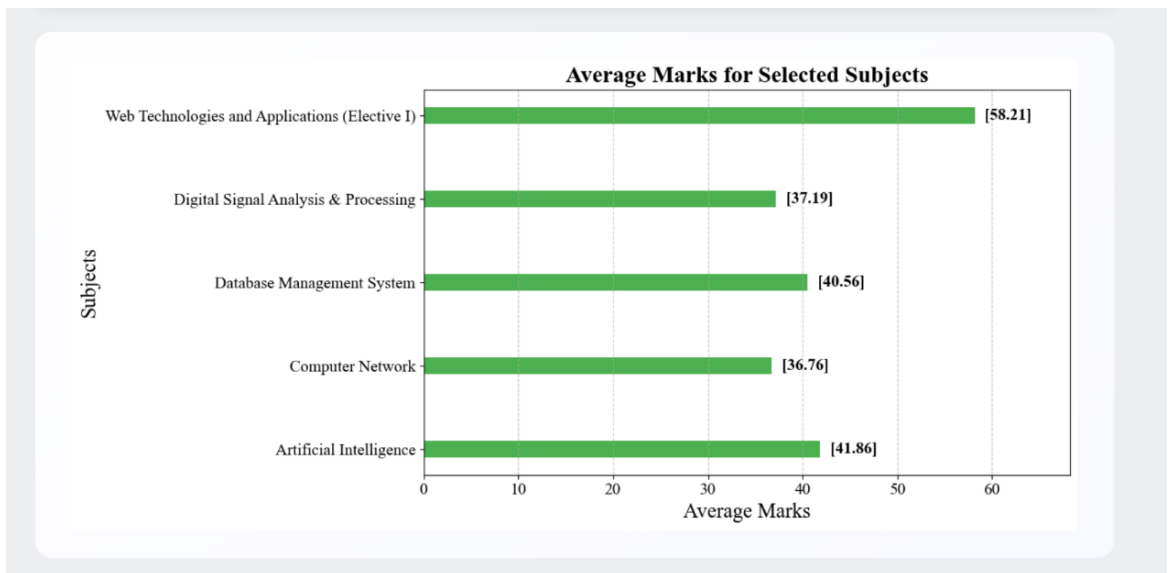


Figure 6-18: Subject Average Marks

The bar chart illustrates the average marks obtained in selected subjects. The x-axis represents the average marks, while the y-axis lists the subjects. Among the subjects, Web Technologies and Applications (Elective I) recorded the highest average score of 58.21, indicating strong student performance in this area. On the other hand, Computer Network had the lowest average score of 36.76, suggesting potential challenges faced by students in this subject. The other subjects—Artificial Intelligence (41.86), Database Management System (40.56), and Digital Signal Analysis & Processing (37.19)—show relatively close average scores, mostly within the 30s and 40s range. This data provides insights into subject-wise student performance, which can help in identifying areas that may require additional academic support or curriculum improvements.

Grade Distribution Analysis

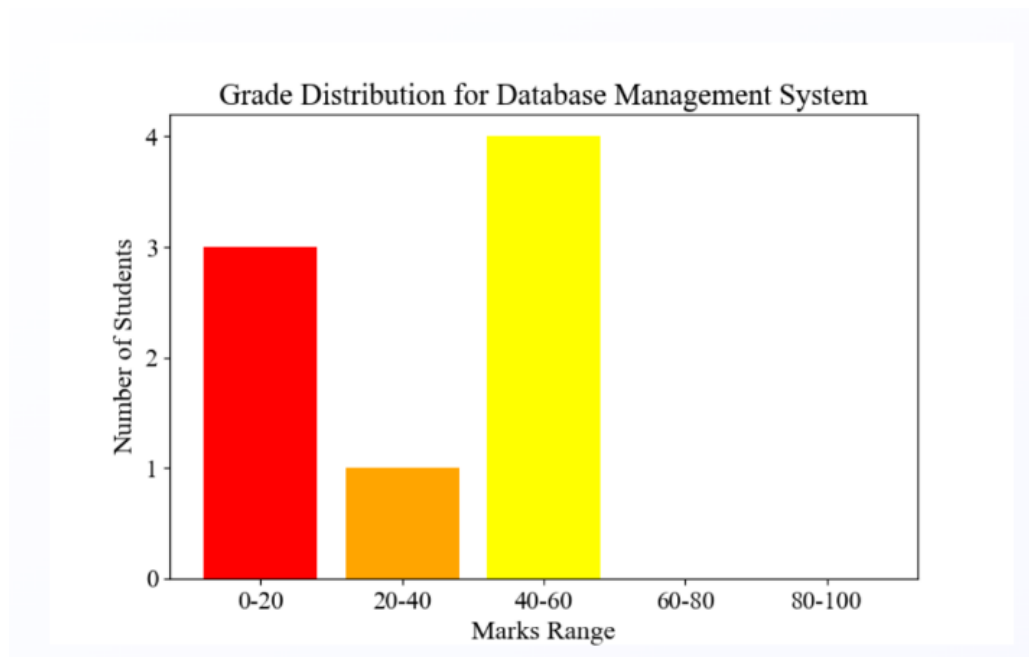


Figure 6-19: Grade Distribution Analysis

The bar chart presents the grade distribution for the Database Management System course, categorizing students based on their marks. The data reveals that a majority of students scored below 60, with the highest concentration in the 40-60 marks range, where four students are placed. Additionally, three students scored between 0-20 marks, indicating a significant portion struggling with the subject, while one student fell within the 20-40 range. Notably, no students scored above 60 marks, suggesting that achieving higher grades in this subject may be challenging. This distribution highlights the need for further academic support, particularly for students scoring in the lower ranges, to improve overall performance and understanding of the subject.

Assessment Vs Final Comparison

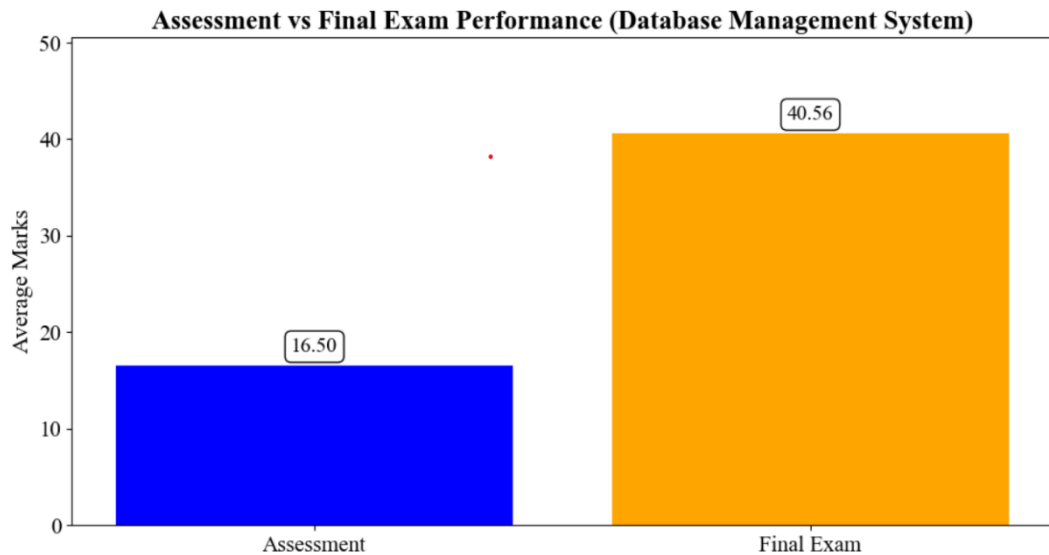


Figure 6-20: Assessment Vs Final Comparison

The bar chart compares assessment performance and final exam performance for the Database Management System course. The x-axis represents the two evaluation components—Assessment and Final Exam—while the y-axis indicates the average marks obtained by students. The data reveals a significant improvement in performance from the assessment to the final exam. The average marks for the assessment stand at 16.50, whereas the final exam average is considerably higher at 40.56. This analysis can help educators evaluate the effectiveness of assessments in preparing students for the final exam.

Subject Improvement

Pass Percentage Trend for Web Technologies and Applications (Elective I) (2072-2078)

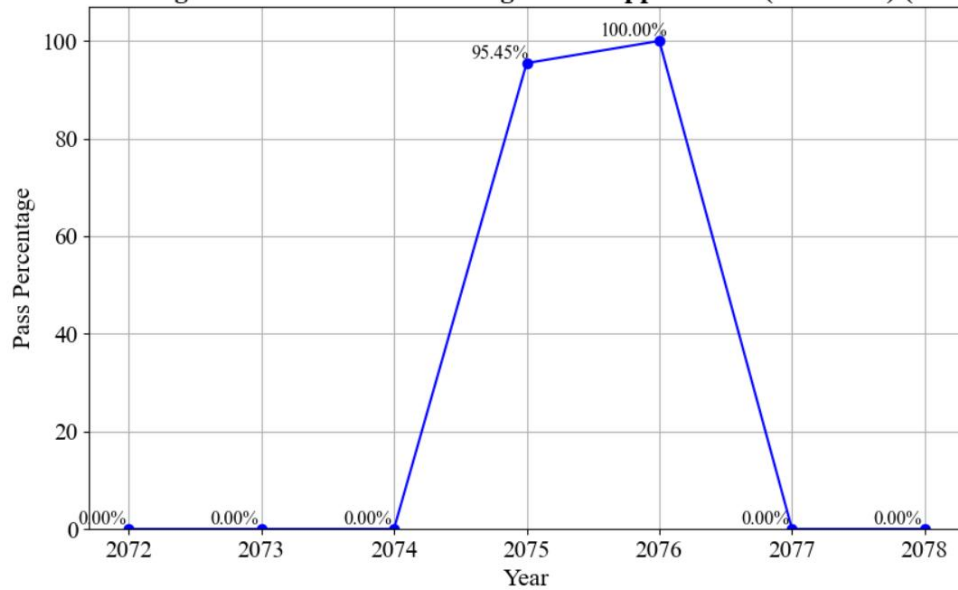


Figure 6-21: Subject Improvement

The chart illustrates the Pass Percentage Trend for Web Technologies and Applications (Elective I) from 2072 to 2078. The data shows that in 2075 the pass percentage is 95.45%, followed by a peak of 100% in 2076, meaning all students who appeared for the exam passed. This analysis helps to observe the passing trends of students in particular subjects over the period of time.

6.2 Table and text extraction

The text inside table and table layout are better detected by paddle-OCR than tesseract-OCR. This is due to tesseract being simple and it focuses on straightforward text extraction but no layout whereas paddle-OCR is trained to detect the complex layout. The text outside the table is more accurately detected by Tesseract, while Paddle-OCR excels in detecting and extracting both the text and layout within tables. Therefore, a combination of both methods is utilized to detect the text and layout of tables as well as to extract the text outside the tables.

Firstly, the table in the image of marksheet is detected using YOLOv8 from where the table part is cropped.



Tribhuvan University
Institute of Engineering
Examination Control Division
Chakrapati, Lalitpur
Back-paper Examination 2080 Ashwin
STATEMENT OF MARKS

Name:- Sanku B. Thakur
Level :- Bachelor's in Engineering
Campus:- Thapathali Campus
Year/Part- III/II

Exam Roll No:- 72254
CRN:- 2075/BCT/037
T.U. Regd. No:- 3.2.26.404.2018
Programme:- Computer Engineering

| Subjects | | Full Marks | | Pass Marks | | Marks Obtained | | Total | Remarks |
|----------|-----------------------------------|------------|-------|------------|-------|----------------|-------|-------|---------|
| | | Asst. | Final | Asst. | Final | Asst. | Final | | |
| CE655 | Engineering Economics | 20 | 80 | 8 | 32 | 12 | A | — | |
| CT651 | Object Oriented Analysis & Design | 20 | 80 | 8 | 32 | 15 | A | — | |
| CT652 | Database Management System | 20 | 80 | 8 | 32 | 14 | A | — | |
| CT653 | Artificial Intelligence | 20 | 80 | 8 | 32 | 17 | A | — | |
| CT655 | Embedded System | 20 | 80 | 8 | 32 | 16 | A | — | |
| CT656 | Operating System | 20 | 80 | 8 | 32 | 16 | A | — | |

Marks Enter By:-
Verified By:-
Date:- 11 JAN 2024
* - Fail A - Absent

Grand Total

Result Absent

Asst. Dean

Figure 6-22: Table Detection

| Subjects | | Full Marks | | Pass Marks | | Marks Obtained | | Total | Remarks |
|----------|-----------------------------------|------------|-------|------------|-------|----------------|-------|-------|---------|
| | | Asst. | Final | Asst. | Final | Asst. | Final | | |
| CE655 | Engineering Economics | 20 | 80 | 8 | 32 | 12 | A | — | |
| CT651 | Object Oriented Analysis & Design | 20 | 80 | 8 | 32 | 15 | A | — | |
| CT652 | Database Management System | 20 | 80 | 8 | 32 | 14 | A | — | |
| CT653 | Artificial Intelligence | 20 | 80 | 8 | 32 | 17 | A | — | |
| CT655 | Embedded System | 20 | 80 | 8 | 32 | 16 | A | — | |
| CT656 | Operating System | 20 | 80 | 8 | 32 | 16 | A | — | |

Figure 6-23: Table Extraction

Now using the paddle OCR the text inside the table is extracted.


| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------------------------|------------|------------|----------------|-------|-----------------------|-------|-------|----------------------------|-------|----|---|-------|----|---|-------|-------------------------|----|----|---|----|----|---|-------|-----------------|----|----|---|----|----|---|-------|------------------|----|----|---|----|----|---|
| Subjects | Full Marks | Pass Marks | Marks Obtained | Code | Title | Asst. | Final | Asst. | Final | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Asst. | Final | Total | Remarks | CE655 | Engineering Economics | 20 | 80 | 8 | 32 | 12 | A | CT651 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Object Oriented Analysis & Design | 20 | 80 | 8 | 32 | 15 | A | CT652 | Database Management System | 20 | 80 | 8 | 32 | 14 | A | CT653 | Artificial Intelligence | 20 | 80 | 8 | 32 | 17 | A | CT655 | Embedded System | 20 | 80 | 8 | 32 | 16 | A | CT656 | Operating System | 20 | 80 | 8 | 32 | 16 | A |

Now the extracted text is converted to excel as in table by using PP(Paddle-OCR Pipeline) structure. PP structure is a library provided by paddleOCR in order to structure the data as tabular structure as replica of image.

| Subjects | | Full Marks | | Pass Marks | | Marks Obtained | | | |
|----------|-----------------------------------|------------|-------|------------|-------|----------------|-------|-------|---------|
| Code | Title | Asst. | Final | Asst. | Final | Asst. | Final | Total | Remarks |
| CE655 | Engineering Economics | 20 | 80 | 8 | 32 | 12 | A | | |
| CT651 | Object Oriented Analysis & Design | 20 | 80 | 8 | 32 | 15 | A | | |
| CT652 | Database Management System | 20 | 80 | 8 | 32 | 14 | A | | |
| CT653 | Artificial Intelligence | 20 | 80 | 8 | 32 | 17 | A | | |
| CT655 | Embedded System | 20 | 80 | 8 | 32 | 16 | A | | |
| CT656 | Operating System | 20 | 80 | 8 | 32 | 16 | A | | |
| | | | | | | | | | |

Figure 6-24: Extracted text in a .xlsx file

Then the table is masked from detected image in order to obtain the information which is outside the table and the text is extracted using tesseract-OCR and formatted in excel file.



Tribhuvan University
Institute of Engineering
Examination Control Division
 Chakrapati, Lalitpur

Back-paper Examination 2080 Ashwin
STATEMENT OF MARKS

Name:- [REDACTED]

Level :- Bachelors in Engineering

Campus:- Thapathali Campus

Year/Part- III/II

Exam Roll No:- 72254

CRN:- 2075/BCT/037

T.U. Regd. No:- 3.2.26.404.2018

Programme:- Computer Engineering

Figure 6-25: Masking of Table

| | |
|---|---|
| Name:- [REDACTED] Level :- Bachelors in Engineering Campus:- Thapathali Campus Year/Part- III/II | Exam Roll No:- 72254 CRN:- 2075/BCT/037 T.U. Regd. No:- 3.2.26.404.2018 Programme:- Computer Engineering |
|---|---|

Figure 6-26: Contextual text extracted

| Name | Exam Roll No | Level | CRN | Campus | T.U. Regd. No | Year/Part | Programme |
|------------|--------------|--------------------------|--------------|-------------------|-----------------|-----------|----------------------|
| [REDACTED] | 72254 | Bachelors in Engineering | 2075/BCT/037 | Thapathali Campus | 3.2.26.404.2018 | III/II | Computer Engineering |
| | | | | | | | |

Figure 6-27: Extracted contextual text in a .xlsx file

Now the excel file obtained from these two files are merged together in order to obtain final excel file.

| Subjects | Unnamed: 1 | Full Marks | Unnamed: 2 | Pass Marks | Unnamed: 3 | Marks Obtained | Unnamed: 4 | Total | Remarks | Name | ka |
|----------------|---------------------------------|--------------|-------------------|------------|------------|---|------------|-------|---------|------|----|
| Code | Title | Asst. | Final | Asst. | Final | Asst. | Final | | | | |
| CT501 | Object Oriented Programming | 20 | 80 | 8 | 32 | 17 | A | | | | |
| EE501 | Electric Circuit Theory | 20 | 80 | 8 | 32 | 18 | A | | | | |
| EE502 | Electrical Engineering Material | 20 | 80 | 8 | 32 | 10 | A | | | | |
| EX501 | Electronic Devices & Circuits | 20 | 80 | 8 | 32 | 17 | A | | | | |
| EX503 | Electromagnetics | 20 | 80 | 8 | 32 | 13 | A | | | | |
| SH501 | Engineering Mathematics III | 20 | 80 | 8 | 32 | 17 | A | | | | |
| Sushmita Bhatt | | | | | | | | | | | |
| Exam Roll No | Level | CRN | Campus | U. Regd. N | Year/Part | Programme | | | | | |
| 15857 | Bachelors in Engin | 2072/BEX/347 | Thapathali Campus | 3.2.26.222 | II/I | Electronics & Communication Engineering | | | | | |

Figure 6-28: Merged excel file

6.3 Database Schema and Instances

Now the obtained data of multiple marksheets from 84 students where a student is inserted in the SQLite database from which the recently uploaded marksheet was shown in the html page.

| student... | name | level | campus | exam_roll... | programme |
|--------------|--------------------|--------------------------|-------------------|--------------|---|
| Filter... | Filter... | Filter... | Filter... | Filter... | Filter... |
| 2075/BCT/037 | Samridha Shrestha | Bachelors in Engineering | Thapathali Campus | 72254 | Computer Engineering |
| 2075/BEV/009 | Bhuwan Khatiwada | Bachelors in Engineering | Thapathali Campus | 52208 | Electronics Communication & Information |
| 2072/BEX/310 | Bibek Dhakal | Bachelors in Engineering | Thapathali Campus | 70852 | Electronics & Communication Engineering |
| 2075/BE/006 | Anjal Bam | Bachelors in Engineering | Thapathali Campus | 52205 | Electronics Communication & Information |
| 2078/BEI/001 | Aayush Chhetri | Bachelors in Engineering | Thapathali Campus | 58223 | Electronics Communication & Information |
| 2078/BEI/015 | Diwas Dahal | Bachelors in Engineering | Thapathali Campus | 58225 | Electronics Communication & Information |
| 2072/BEX/347 | Sushmita Bhatt | Bachelors in Engineering | Thapathali Campus | 70855 | Electronics & Communication Engineering |
| 2073/BEX/316 | Kapalik Khanal | Bachelors in Engineering | Thapathali Campus | 54852 | Electronics & Communication Engineering |
| 2074/BEX/036 | Satkrit Raj Pandey | Bachelors in Engineering | Thapathali Campus | 70862 | Electronics & Communication Engineering |
| 2075/BEI/009 | Bhuwan Khatiwada | Bachelors in Engineering | Thapathali Campus | 58201 | Electronics Communication & Information |
| 2077/BEI/009 | Ashim Panthi | Bachelors in Engineering | Thapathali Campus | 58215 | Electronics Communication & Information |
| 2075/BEI/003 | Abhishek Poudel | Bachelors in Engineering | Thapathali Campus | 73201 | Electronics Communication & Information |
| 2076/BEI/005 | Anuka K.C. | Bachelors in Engineering | Thapathali Campus | 73207 | Electronics Communication & Information |
| 2076/BEI/007 | Ayush Acharya | Bachelors in Engineering | Thapathali Campus | 73208 | Electronics Communication & Information |
| 2076/BEI/008 | Anupam Bhattarai | Bachelors in Engineering | Thapathali Campus | 73209 | Electronics Communication & Information |
| 2075/BCT/033 | Prashant Bhusal | Bachelors in Engineering | Thapathali Campus | 72253 | Computer Engineering |
| 2072/BEX/308 | Anil Tamang | Bachelors in Engineering | Thapathali Campus | 70851 | Electronics & Communication Engineering |
| 2073/BEX/311 | Gokul Adhikari | Bachelors in Engineering | Thapathali Campus | 70856 | Electronics & Communication Engineering |
| 2073/BEX/314 | James Shrestha | Bachelors in Engineering | Thapathali Campus | 70857 | Electronics & Communication Engineering |
| 2074/BEX/002 | Abhinait Kumar Das | Bachelors in Engineering | Thapathali Campus | 70858 | Electronics & Communication Engineering |
| 2074/BEX/025 | Pukar Giri | Bachelors in Engineering | Thapathali Campus | 70859 | Electronics & Communication Engineering |
| 2075/BCT/023 | Kumar Tiwari | Bachelors in Engineering | Thapathali Campus | 65804 | Computer Engineering |

Figure 6-29: Student Information Table

| id | year_part | marks_as... | marks_final | studen... | title_id | total_mar... |
|----|-----------|-------------|-------------|--------------|---|--------------|
| 54 | I/II | 12.0 | A | 2075/BCT/037 | Engineering Economics | - |
| 55 | I/II | 15.0 | A | 2075/BCT/037 | Object Oriented Analysis & Design | - |
| 56 | I/II | 14.0 | A | 2075/BCT/037 | Database Management System | - |
| 57 | I/II | 17.0 | A | 2075/BCT/037 | Artificial Intelligence | - |
| 58 | I/II | 16.0 | A | 2075/BCT/037 | Embedded System | - |
| 59 | I/II | 16.0 | A | 2075/BCT/037 | Operating System | - |
| 60 | IVA | 14.0 | 32 | 2075/BEV/009 | Digital Signal Analysis & Processing | 46.0 |
| 61 | IVA | 22.0 | '' | 2075/BEV/009 | Digital Signal Analysis & Processing PRACTICAL | 22.0 |
| 62 | IVA | 15.0 | 41 | 2075/BEV/009 | Artificial Intelligence | 56.0 |
| 63 | IVA | 24.0 | '' | 2075/BEV/009 | Artificial Intelligence PRACTICAL | 24.0 |
| 64 | IVA | 44.0 | '' | 2075/BEV/009 | Project I PRACTICAL | 44.0 |
| 65 | IVA | 15.0 | 32 | 2075/BEV/009 | RF& Microwave Engineering | 47.0 |
| 66 | IVA | 24.0 | '' | 2075/BEV/009 | RF & Microwave Engineering PRACTICAL | 24.0 |
| 67 | IVA | 18.0 | 45 | 2075/BEV/009 | Organization & Management | 63.0 |
| 68 | IVA | 21.0 | '' | 2075/BEV/009 | Web Technologies and Applications (Elective I PRACTICAL | 21.0 |
| 69 | II/I | 15.0 | A | 2075/BCT/037 | Object Oriented Programming | - |
| 70 | II/I | 17.0 | A | 2075/BCT/037 | Theory of Computation | - |
| 71 | II/I | 12.0 | A | 2075/BCT/037 | Electric Circuit Theory | - |
| 72 | II/I | 15.0 | A | 2075/BCT/037 | Electronic Devices & Circuits | - |
| 73 | II/I | 13.0 | A | 2075/BCT/037 | Digital Logic | - |
| 74 | II/I | 18.0 | A | 2075/BCT/037 | Electromagnetics | - |
| 75 | II/I | 13.0 | A | 2075/BCT/037 | Engineering Mathematics III | - |

Figure 6-30: Marks Obtained Table

| subject_c... | title | full_mark... | full_mark... | pass_mar... | pass_mar... |
|--------------|---|--------------|--------------|-------------|-------------|
| CE655 | Engineering Economics | 20 | 80 | 8 | 32 |
| CT651 | Object Oriented Analysis & Design | 20 | 80 | 8 | 32 |
| CT652 | Database Management System | 20 | 80 | 8 | 32 |
| CT653 | Artificial Intelligence | 20 | 80 | 8 | 32 |
| CT655 | Embedded System | 20 | 80 | 8 | 32 |
| CT656 | Operating System | 20 | 80 | 8 | 32 |
| CT704 | Digital Signal Analysis & Processing | 20 | 80 | 8 | 32 |
| CT704 | Digital Signal Analysis & Processing PRACTICAL | 25 | 0 | 10 | 0 |
| CT710 | Artificial Intelligence PRACTICAL | 25 | 0 | 10 | 0 |
| EX707 | Project I PRACTICAL | 50 | 0 | 20 | 0 |
| EX716 | RF& Microwave Engineering | 20 | 80 | 8 | 32 |
| EX716 | RF & Microwave Engineering PRACTICAL | 25 | 0 | 10 | 0 |
| ME708 | Organization & Management | 20 | 80 | 8 | 32 |
| 72505 | Web Technologies and Applications (Elective I PRACTICAL | 25 | 0 | 10 | 0 |
| CT451 | Object Oriented Programming | 20 | 80 | 8 | 32 |
| CT502 | Theory of Computation | 20 | 80 | 8 | 32 |
| EE501 | Electric Circuit Theory | 20 | 80 | 8 | 32 |
| X501 | Electronic Devices & Circuits | 20 | 80 | 8 | 32 |
| EX502 | Digital Logic | 20 | 80 | 8 | 32 |
| EX503 | Electromagnetics | 20 | 80 | 8 | 32 |
| SH501 | Engineering Mathematics III | 20 | 80 | 8 | 32 |
| EX601 | Advanced Electronics | 20 | 80 | 8 | 32 |

Figure 6-31: Subject Information Table

6.4 Result Analysis (Tesseract VS Paddle)

| Subjects | | Full Marks | | Pass Marks | | Marks Obtained | | |
|----------|--|------------|-------|------------|-------|----------------|-------|-------|
| Code | Title | Asst. | Final | Asst. | Final | Asst. | Final | Total |
| CT701 | Project Management | 20 | 80 | 8 | 32 | 18 | 46 | 64 |
| CT702 | Computer Network | 20 | 80 | 8 | 32 | 12 | 32 | 44 |
| CT702 | Computer Network PRACTICAL | 50 | | 20 | | 35 | | 35 |
| CT703 | Distributed System | 20 | 80 | 8 | 32 | 18 | 40 | 58 |
| CT703 | Distributed System PRACTICAL | 25 | | 10 | | 24 | | 24 |
| CT704 | Digital Signal Analysis & Processing | 20 | 80 | 8 | 32 | 19 | 47 | 66 |
| CT704 | Digital Signal Analysis & Processing PRACTICAL | 25 | | 10 | | 25 | | 25 |
| CT707 | Project I PRACTICAL | 50 | | 20 | | 45 | | 45 |
| EX701 | Energy Environment & Society | 10 | 40 | 4 | 16 | 6 | 24 | 30 |
| ME708 | Organization & Management | 20 | 80 | 8 | 32 | 18 | 45 | 63 |
| CT72502 | Data Mining (Elective I) | 20 | 80 | 8 | 32 | 14 | 45 | 59 |
| CT72502 | Data Mining (Elective I) PRACTICAL | 25 | | 10 | | 20 | | 20 |

Figure 6-32: Table obtained by using paddle OCR

| Code | Title | Asst. | Final | Asst. | Final | Asst. | Final | Total | Remarks |
|-----------|--|-------|-----------|-------|-------|-------|-------|-------|---------|
| CT701 | Project Management | 20 | 80 | 8 | 32 | 18 | 46 | 64 | |
| Sia | Computer Network | 20 | 80 | 8 | 32 | 12 | 32 | 44 | |
| CT702 | Computer Network PRACTICAL SO oes Ge ee tC | | | | | | | | |
| CT703 | Distributed System | 20 | 80 | 8 | 32 | 18 | 40 | 58 | |
| CT703 | Distributed System PRACTICAL | 25 | | 10 | 2s | 24 | 24 | | |
| CT704 | Digital Signal Analysis & Processing | 20 | 80 | 8 | 32 | 19 | 47 | 66 | |
| CT704 | Digital Signal Analysis & Processing | 25 | | 10 | | 25 | | 25 | |
| PRACTICAL | | | | | | | | | |
| CT707 | Project | 1 | PRACTICAL | 50 | | 20 | | 45 | 45 |
| i | | | | | | | | | |
| EX701 | Energy Environment & Society | 10 | 40 | 4 | 16 | 6 | 24 | 30 | |
| ME708 | Organization & Management | 20 | 80 | 8 | 32 | 18 | 45 | 63 | |
| CT72502 | Data Mining Elective I | 20 | 80 | 8 | 32 | 14 | 45 | 59 | i |
| CT72502 | Data Mining Elective I PRACTICAL | 25 | | 10 | | 20 | 20 | | |

Figure 6-33: Table obtained by using paddle OCR

First, Tesseract OCR was used to extract table data, but the results were not satisfactory. It struggled with structured layouts, leading to multiple errors in the extracted data. Since Tesseract is primarily designed for simple text recognition without considering complex layouts, it was unable to accurately process the table format found in marksheets. To address this issue, Paddle-OCR was tested and found to perform significantly better for table data extraction. Unlike Tesseract, Paddle-OCR is trained to detect and process structured layouts, making it more effective for handling complex tabular data. Several marksheets were tested, and the results consistently showed that Paddle-OCR produced more accurate extractions.

However, for text outside the tables, Tesseract proved to be more reliable. It performed

well in recognizing simple, linear text patterns and efficiently extracted straightforward textual information. This suggests that while Paddle-OCR is better suited for structured data like tables, Tesseract is still a strong choice for extracting unstructured or plain text.

6.5 Character Accuracy Rate

Character Accuracy Rate (CAR) measures the percentage of correctly recognized characters in an OCR output compared to the ground truth. It is commonly used to evaluate the performance of Optical Character Recognition (OCR) systems, especially in scenarios where fine-grained accuracy is required. This provides insight into how well an OCR system can recognize individual letters, numbers, and punctuation marks, regardless of the overall structure of the text or formatting.

$$\text{Character Accuracy Rate (CAR)} = (1 - ((S + D + I)/N)) \times 100 \quad 6-1$$

Where:

- S = Number of substitutions (wrong characters)
- D = Number of deletions (missing characters)
- I = Number of insertions (extra characters)
- N = Total number of characters in the ground truth

Evaluation:

For the evaluation of OCR accuracy, we utilized 10 marksheets from the 2075 batch, IV/I part. The extracted text was compared against the ground truth to calculate the Character Accuracy Rate (CAR).

Using the Levenshtein distance method, we identified substitutions, deletions, and insertions in the OCR-generated text. Based on these calculations, the **average CAR across the 10 samples was determined to be 94.62%**, indicating a high level of accuracy in character recognition.

7. FUTURE ENHANCEMENTS

The improvements that can be worked on the project in the future includes:

7.1 Integration of Student Records for Enhanced Analysis in QAA Reports

College student records can be integrated to facilitate multi-parameter analysis of examination results. This allows for insights such as gender-wise pass rates, district-wise performance, and comparisons between regular and paying students. The analyzed data can be effectively utilized in Quality Assurance and Accreditation (QAA) reports.

7.2 Enhancing Accuracy with Paid OCR Integration

Integrating a paid OCR (Optical Character Recognition) service can significantly improve the accuracy of text extraction from hardcopy marksheets, especially for noisy images with stamps, handwritten notes, or low-quality prints.

8. CONCLUSION

In conclusion, the Marksheet Digitization using OCR and Report Generation for QAA project has successfully fulfilled its objectives of automating marksheet processing and report generation. By leveraging Tesseract OCR and Paddle OCR, the system efficiently extracts text from scanned marksheets and stores the data in an SQLite database. This structured storage enables seamless result generation for both students and campuses.

The admin plays a key role in managing the system, as they can upload scanned marksheets to store records, analyze student performance, and generate various reports based on extracted data. The system not only enhances administrative efficiency but also ensures accuracy in record-keeping. Meanwhile, students can easily access their marks through a user-friendly dashboard and have the option to download their results in PDF format, making the process more convenient and transparent.

Despite these achievements, there is room for further improvement. Expanding the system to include marksheets from all academic programs and years will provide deeper insights and generate more comprehensive reports with better graphical representations. Additionally, refining the OCR processing to handle noisy or stamped marksheets more effectively would enhance the accuracy and reliability of text extraction. Future upgrades could also involve integrating advanced machine learning techniques to improve OCR accuracy and incorporating data visualization tools for better performance analysis. Overall, this project represents a significant step toward modernizing academic record management, reducing manual effort, and improving accessibility for both administrators and students.

9. APPENDICES

Appendix A: Project Schedule

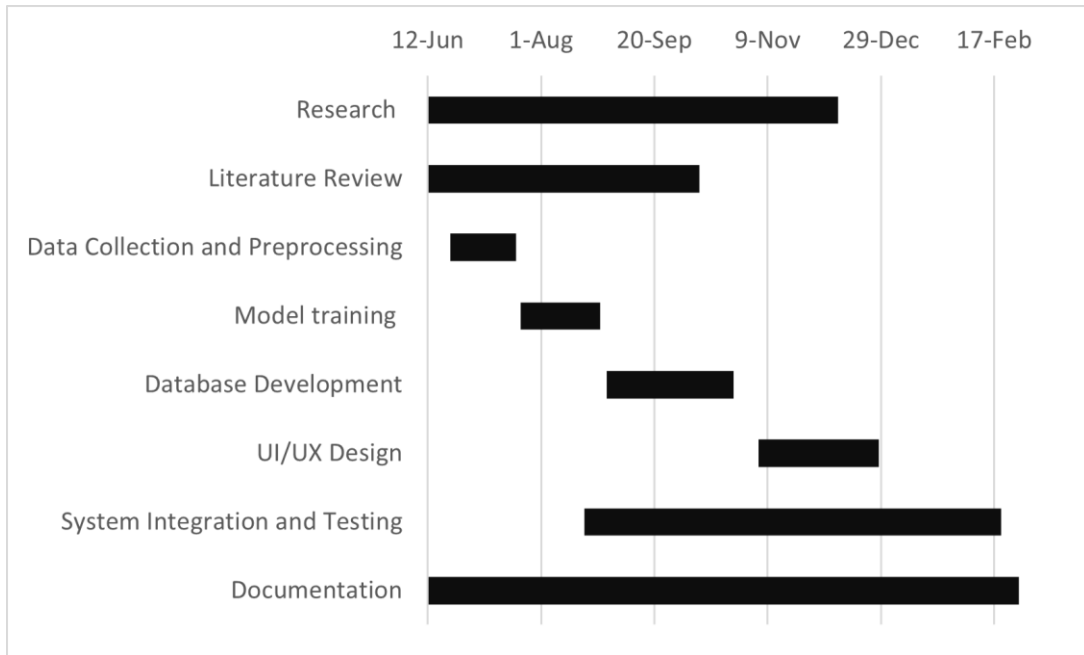


Figure 9-1: Gantt Chart

Appendix B: Project Budget

Since many resources are freely available and open, purchasing additional resources may not be necessary. But there may be additional cost such as printing report, during dataset collection and so on.

Table 9-1: Project Budget

| Particulars | Price |
|---------------|-------|
| Miscellaneous | 10000 |
| Total | 10000 |

Appendix C: Code Snippets

```
# Configure tesseract
custom_config = r'-l eng --oem 3 --psm 6 -c tesseract_char_whitelist="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789*:-./& "'

# Perform OCR
d = pytesseract.image_to_data(dilated, config=custom_config, output_type=Output.DICT)
# Convert to DataFrame
df = pd.DataFrame(d)
```

Figure 9-2: Code for Extraction from Tesseract OCR

```
ocr = PaddleOCR(use_angle_cls=True, lang='en')
table_engine = PPStructure(show_log=True, lang='en', layout=False)

save_folder='temp2'
if not os.path.exists(save_folder):
    os.makedirs(save_folder)

cropped_image_path='cropped_image.png'
# Get table structure result
img = cv2.imread(cropped_image_path)
result = table_engine(img)

# Save the structure result
save_structure_res(result, save_folder, os.path.basename(cropped_image_path).split('.')[0])
```

Figure 9-3: Code for Extraction from Paddle OCR and Structured Table

References

- [1] P. Pyreddy and W. B. Croft , "TINTIN: A System for Retrieval in Text Tables," Proceedings of the second ACM international conference on Digital libraries, 1997.
- [2] T. Kasar, P. Barlas, S. Adam, C. Chatelain and T. Paquet, "Learning to Detect Tables in Scanned Document Images Using Line Information," 2013 12th International Conference on Document Analysis and Recognition. IEEE, 2013.
- [3] A. C. e. Silva, "Learning rich hidden markov models in document analysis: Table location.," 2009 10th International Conference on Document Analysis and Recognition. IEEE, 2009.
- [4] T. Ojala, M. Pietikäinen and T. & Mäenpää, "Gray scale and rotation invariant texture classification with local binary patterns," Springer Berlin Heidelberg, 2000.
- [5] R. P. dos Santos, G. S. Clemente, T. I. Ren and G. D. Cavalcanti, "Text line segmentation based on morphology and histogram projection.," 2009 10th International Conference on Document Analysis and Recognition. IEEE, 2009.
- [6] L. Likforman-Sulem, A. Zahour and B. Taconet, "Text line segmentation of historical documents: a survey.," International Journal of Document Analysis and Recognition (IJDAR) 9, 2007.
- [7] K. Wang, B. Babenko and S. Belongie, "End-to-end scene text recognition.," 2011 International conference on computer vision. IEEE, 2011.
- [8] D. N. Tran, T. A. Tran, A. Oh, S. H. Kim and I. S. & Na, "Table detection from document image using vertical arrangement of text blocks," International Journal of Contents, 2015.

- [9] Z. Q. Zhao, P. Zheng, S. T. Xu and X. Wu, "Object detection with deep learning: A review.," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212-3232., 2019.
- [10] X. Zhao, E. Niu, Z. Wu and X. Wang, "CUTIE: Learning to Understand Documents with Convolutional Universal Text Information Extractor," *arXiv preprint arXiv:1903.12363*., 2019.
- [11] R. B. Palm, O. Winther and F. Laws, "CloudScan - A Configuration-Free Invoice Analysis System Using Recurrent Neural Networks," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017.
- [12] S. Schreiber, S. Agne, I. Wolf, A. Dengel and S. & Ahmed, "Deepdesrt: Deep learning for detection and structure recognition of tables in document images.," *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*. Vol. 1. IEEE, 2017.
- [13] I. Kavasidis, C. Pino, S. Palazzo, F. Rundo, D. Giordano, P. Messina and C. Spampinato, "A saliency-based convolutional neural network for table and chart detection in digitized documents.," in *Image Analysis and Processing–ICIAP 2019: 20th International Conference, Trento, Italy, September 9–13, 2019, Proceedings, Part II 20.*, Springer International Publishing, 2019.
- [14] V. Sunder, A. Srinivasan, L. Vig, G. Shroff and R. Rahul, "One-shot information extraction from document images using neuro-deductive program synthesis," *arXiv preprint arXiv:1906.02427*, 2019.
- [15] S. S. Paliwal, D. Vishwanath, R. Rahul, M. Sharma and L. Vig, "TableNet: Deep Learning model for end-to-end," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019.
- [16] A. Berthe, "Text extraction from ID card using deep learning," *Ikomia*, 2022.

