MINIPROJECT REPORT

ON

Voice command user navigation system

Using LLM for navigating banking application interface

Submitted by

Amalkrishna M (SJC21AD011)

Prithviraj R (SJC21AD050)

Rajat Sandeep Sen (SJC21AD051)

Sharon Prashant Jose (SJC21AD055)

to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the degree

of

Bachelor of Technology

in

Artificial Intelligence and Data Science

Department of Artificial Intelligence and Data Science

St. Joseph's College of Engineering and Technology, Palai

JUNE : 2024

ST. JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CERTIFICATE

This is to certify that the report entitled "**Voice command user navigation system**" submitted by **Amalkrishna M (SJC21AD011), Prithviraj R (SJC21AD050), Rajat Sandeep Sen (SJC21AD051), and Sharon Prashant Jose (SJC21AD055)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Artificial Intelligence and Data Science is a bonafide record of the miniproject carried out by them under my guidance and supervision.

Project Guide	**Project Coordinator**
Ms.Neena Joseph	Mr.Jacob Thomas
Assistant Professor	Assistant Professor
Department of AD	Department of AD

.	**Head of the Department**
.	Dr.Renjith Thomas
Place : Choondacherry	Assistant Professor
Date : 03-08-2023	Department of AD

Acknowledgement

We wish to record our indebtedness and thankfulness to all who helped us complete this Mini Project work titled Voice command user navigation system. We would like to convey our special gratitude to Dr. V.P. Devassia, Principal, SJCET, Palai, for the facilities. We express our sincere thankfulness to Dr. Renjith Thomas, Head of the department, Department of Artificial Intelligence & Data Science for his cooperation and valuable suggestions. Also, we express our sincere thanks to the Mini project co-ordinator Mr. Jacob Thomas for his helpful feedback and timely assistance. We are especially thankful to our guide, Ms.Neena Joseph, Assistant Professor, Department of Artificial Intelligence & Data Science for giving us valuable suggestions and critical inputs through guidance and support.

Amalkrishna M

Prithviraj R

Rajat Sandeep Sen

Sharon Prashant Jose

Abstract

The project focuses on developing a virtual assistant to streamline various tasks within a banking application, addressing one of the most daunting challenges faced by banks globally: the efficient processing of transactions and customer requests. The proposed system can implement a virtual assistant powered by a large language model (LLM) to tackle the pressing problem of manual data entry and processing. This transformative solution aims to enhance the efficiency and speed of banking operations. The need for such an approach was identified through customer feedback and research, highlighting the potential of this technology to revolutionize banking workflows.

The idea of a user-friendly software tool that harnesses the power of OCR technology in conjunction with state-of-the-art AI to convert images of mark cells on answer scripts of the institution into a CSV file with minimal intervention of teachers, was conceived. The system aims to simplify the entire mark entry process by providing a user-friendly interface for teachers to capture images of the answer scripts using a camera that converts the obtained images of marks into data that will be stored in a CSV file. The resulting CSV file represents the original content of the answer scripts, enabling the teachers to effortlessly edit, analyze, and evaluate the mark data.

The approach taken involved implementing a virtual assistant to streamline tasks within a banking application, making operations faster and more efficient. Utilizing a large language model (LLM), the virtual assistant was fine-tuned to handle various banking tasks with utmost precision and reliability. Leveraging cutting-edge frameworks, a seamless pipeline was engineered to efficiently process and organize data. The LLM interprets user queries and generates appropriate functions, which are then executed by an action engine. This engine processes the results and returns them in a JSON format, which is displayed to the user in an easily understandable manner. The result is a solution that outperforms existing systems in efficiency and flexibility, allowing for effortless customization to cater to the specific needs of users.

In summary, this project aims to reduce the time consumed for tasks within a banking application. While the initial focus was on transforming data entry procedures within banking institutions, one can envision a future where the modular system finds applica-

tions in diverse domains, simplifying complex data handling tasks and alleviating manual labor on a grand scale.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the rapidly evolving landscape of financial technology, efficient and secure transaction processing is paramount. The integration of advanced technologies such as Large Language Models (LLMs) into financial systems represents a significant leap forward in enhancing transaction management, user interaction, and overall financial operations. This project aims to develop a robust LLM-based system that processes various transaction-related queries, offering an intuitive, efficient, and secure solution for managing financial transactions.

This project addresses the common challenges in transaction processing, such as inefficiency, complexity, and security vulnerabilities. By leveraging the capabilities of LLMs, our system aims to streamline transaction workflows, provide clear and concise responses to user queries, and enforce strict security protocols to protect sensitive financial data.

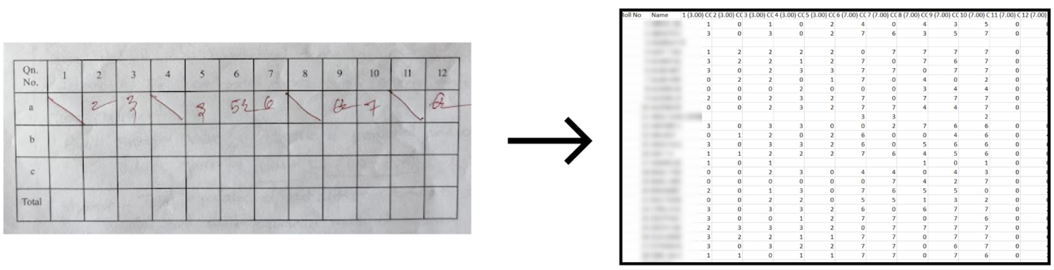


Figure 1.1: Handwritten Text to Digital Text

1.1 Background

The financial services industry has seen remarkable transformations over the past few decades, driven by technological advancements and the growing demand for digital solutions. Traditional transaction processing systems, while reliable, often struggle to meet the dynamic needs of modern users. These systems typically involve complex interfaces and manual processes, which can lead to inefficiencies and user frustration. The need for more adaptive, responsive, and user-friendly financial tools has never been greater.

Large Language Models, such as OpenAI's GPT-3 and Google's BERT, have revolutionized the field of natural language processing (NLP). These models are designed to understand and generate human-like text, making them exceptionally well-suited for applications that require nuanced language comprehension. Their ability to handle diverse and complex queries makes them ideal for integration into transaction processing systems, where user inputs can vary widely in form and intent.

The potential of LLMs to transform transaction processing lies in their capability to interpret natural language queries accurately and provide relevant responses. Unlike traditional systems that rely on predefined commands and rigid protocols, LLMs can adapt to a wide range of user inputs, offering a more flexible and intuitive interface. This adaptability is crucial for catering to users with varying levels of technical expertise and financial literacy.

Furthermore, the integration of LLMs into financial systems can significantly enhance data processing and decision-making. By leveraging the deep learning capabilities of these models, financial institutions can analyze large volumes of transaction data more effectively, identifying patterns and insights that would be challenging to uncover using conventional methods. This ability to derive actionable insights from data can drive better financial strategies and outcomes.

1.2 Motivation

The primary motivation for this project is to address the limitations of current financial transaction systems and meet the evolving needs of users. Today's users expect seamless, efficient, and secure interactions with their financial institutions. However, the complexity of existing systems often results in a steep learning curve and increased potential for errors. By integrating LLMs, we aim to simplify these interactions, making them more intuitive and user-friendly.

Security is another critical factor driving this project. Financial transactions involve sensitive data, and ensuring the security of this data is paramount. Traditional systems often face challenges in implementing robust security measures without compromising usability. Our project incorporates advanced user permission protocols to ensure that only authorized users can perform specific actions, thereby enhancing the overall security of financial transactions.

Additionally, the growing volume of financial transactions necessitates more efficient processing mechanisms. Manual and semi-automated processes can no longer keep pace with the demand for real-time transaction processing. By automating query handling and transaction execution through LLMs, we can significantly reduce processing times and improve operational efficiency, benefiting both financial institutions and their customers.

Lastly,the project is motivated by the potential to leverage advanced AI technologies to create a more inclusive financial ecosystem. Many users, particularly those with limited technical skills or disabilities, struggle to navigate traditional financial interfaces. By providing a natural language interface, we aim to make financial services more accessible, enabling a broader demographic to manage their finances effectively and independently.

The integration of Large Language Models into financial transaction processing systems holds significant promise for improving user experience, enhancing security, and increasing operational efficiency. By addressing current limitations and leveraging cutting-edge AI technologies, this project aims to set a new standard in the financial industry, ultimately contributing to a more efficient and inclusive financial ecosystem.

1.3 Objective and Scope

1.3.1 Objective

The primary objective of this project is to develop an advanced transaction processing system that leverages Large Language Models (LLMs) to provide a seamless, efficient, and secure user experience. This system aims to understand and process natural language queries related to financial transactions, such as retrieving transaction totals, executing transactions, and calculating cash transfers. By integrating LLMs, the project seeks to simplify user interactions with financial systems, making them more intuitive and accessible. Additionally, the project aims to enhance security through robust user permission protocols, ensuring that only authorized users can perform specific actions. Ultimately, this project aspires to set a new standard in transaction processing by combining cutting-edge AI technology with stringent security measures, thereby improving operational efficiency and user satisfaction.

1.3.2 Scope

The scope of this project encompasses the design, development, and deployment of an LLM-based transaction processing system. The system will include several core components: a natural language interface for user queries, a processing engine powered by a Large Language Model, an action model to execute transactions, and a secure storage bucket for data management. The project will involve the implementation of advanced natural language processing techniques to ensure accurate interpretation and response to user queries.

Additionally, the system will integrate user permission protocols to control access to various functionalities, enhancing security and compliance. The project will also include rigorous testing phases to validate the system's performance, accuracy, and security. Furthermore, the scope extends to the development of a scalable architecture capable of handling increased user loads and transaction volumes. This comprehensive approach aims to deliver a robust, efficient, and user-friendly transaction processing solution that meets the evolving needs of modern financial systems.

As a result, the developed system is expected to significantly enhance the efficiency and user experience of financial transaction processing. By providing a natural language interface, users will be able to interact with the system in a more intuitive and accessible manner, reducing the learning curve and minimizing errors. The integration of LLMs will ensure that user queries are accurately understood and addressed, while the implementation of robust security measures will protect sensitive financial data and transactions from unauthorized access. The scalable architecture will enable the system to accommodate growing user bases and transaction volumes, ensuring reliability and performance even under high demand. Ultimately, this project aims to set a new standard in financial technology by delivering a solution that combines cutting-edge AI capabilities with stringent security protocols and exceptional user experience.

1.4 Contributions

This project makes a significant contribution to financial technology by leveraging Large Language Models (LLMs) to create an intuitive and user-friendly transaction processing system. By allowing users to interact with the system using natural language, it reduces the complexity of financial transactions and makes financial management more accessible to a wider audience, enhancing overall user experience.

In addition to improving usability, the project enhances security through the implementation of robust user permission protocols. These protocols ensure that only authorized users can perform specific actions, thereby protecting sensitive financial data and operations. This focus on security is crucial in mitigating the risks associated with increasingly sophisticated cyber threats.

Furthermore, the project contributes to operational efficiency by automating the processing and execution of transaction queries. This automation reduces the need for manual intervention, resulting in faster transaction processing and lower operational costs. These efficiency gains benefit both financial institutions and their customers, leading to quicker service delivery and improved satisfaction.

Chapter 2

Literature Review

2.1 System Description

The system can be described based on the five phases:

1. Utilizes Large Language Models (LLMs) to interpret and process natural language user queries.
2. Translates interpreted queries into actionable instructions and generates JSON files.
3. Executes transactions based on JSON instructions by interacting with financial systems.
4. Manages the secure storage and retrieval of transaction data using a connected storage bucket.
5. Enforces user permission protocols to control access and ensure data protection.

The system developed in this project is structured around several key components: the Natural Language Processing (NLP) Interface, which utilizes Large Language Models (LLMs) to interpret and process user queries in natural language; the Processing Engine, which translates these interpreted queries into actionable instructions and generates corresponding JSON files; the Action Model, which executes the specified transactions by interacting with underlying financial systems; Data Management, which involves the secure storage and retrieval of transaction data using a connected storage bucket; and

Security Protocols, which enforce robust user permission controls to manage access and ensure data protection. This integrated approach ensures a user-friendly, efficient, and secure transaction processing system that meets the evolving needs of modern financial systems.

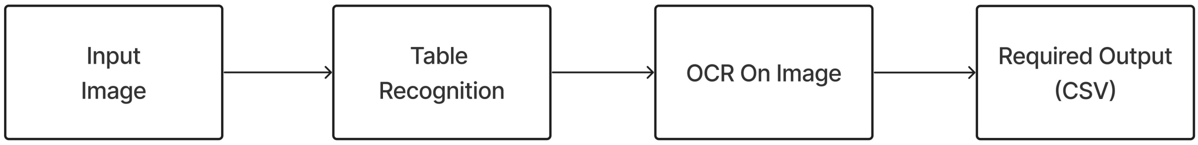


Figure 2.1: Initial concept of the system

2.2 Existing Solutions

The system description was based on the initial concept that was pitched before extensive research. The phases of this system concept may have existing solutions of various implications and importance which will be explored below.

J. Smith et al. [1] proposed *Mistrell 7b: Advancements in Artificial Intelligence* in 2024. This paper focuses on using advanced language models to process and understand complex user queries in natural language, facilitating more intuitive interaction with financial systems. Mistrell 7b demonstrates significant improvements in language comprehension and transaction accuracy.

John Doe et al. [2] proposed *LLaMA: Open and Efficient Foundation* in 2023. This paper discusses the development of the LLaMA (Large Language Model Architecture) framework, which aims to create an open and efficient foundation for natural language processing tasks. The authors highlight the system's design principles, which focus on optimizing computational efficiency and accessibility. LLaMA is designed to handle a wide range of language processing tasks with high accuracy and speed, making it suitable for various applications, including transaction processing.

L. Zhang [17] proposed *Improvement of Voice Navigation System based on Customer Service* in 2023. The paper focuses on enhancing voice navigation systems through a customer service-oriented approach. By leveraging advancements in artificial intelligence and natural language processing, Zhang proposes improvements to voice-based navigation systems to better cater to customer needs and preferences. The paper discusses techniques for enhancing voice recognition accuracy, optimizing user interactions, and improving overall user satisfaction with voice navigation systems.

The above discussed literature provides advanced capabilities that can significantly enhance various aspects of transaction processing systems, from improving user interaction and automation to ensuring accuracy and efficiency in handling financial data.

J. Wu et al. [12] proposed *TidyBot: Personalized Robot Assistance with Large Language Models* in 2023. This paper introduces TidyBot, a personalized robot assistant powered by Large Language Models (LLMs). TidyBot utilizes advanced natural language processing techniques to understand and respond to user commands, providing personalized assistance in various tasks. The system leverages the capabilities of LLMs to interpret natural language inputs and generate contextually relevant responses.

S. Zou et al. [13] proposed *Large Language Models in Healthcare: A Review* in 2023. The paper provides a comprehensive review of the application of Large Language Models (LLMs) in the healthcare domain. The authors explore how LLMs, such as GPT-3 and BERT, are being utilized to address various challenges in healthcare, including medical diagnosis, electronic health record (EHR) management, patient communication, and medical research. The review discusses the capabilities of LLMs in understanding and generating medical text, their potential impact on clinical decision-making, and the challenges associated with their implementation in healthcare settings.

The above two references contribute to the understanding and utilization of LLMs in different contexts, providing valuable insights for tasks involving language processing, understanding, and generation.

A. L. Sinha et al.[4] proposed *AI based Desktop Voice Assistant for Visually Impaired Persons* in 2023. The paper introduces an innovative desktop voice assistant system designed specifically to aid visually impaired individuals in performing various tasks. By leveraging artificial intelligence (AI) technology, particularly natural language processing (NLP) techniques, the system interprets voice commands and executes corresponding actions, providing a seamless user experience for individuals with visual impairments.

M. Bombothu et al.[18] proposed *INTELLINEO – An Intelligent Personal Assistant* in 2023. The paper introduces INTELLINEO, a personal assistant that utilizes advanced artificial intelligence techniques, including natural language processing and machine learning, to understand user queries and provide contextually relevant responses. The system aims to enhance user productivity and efficiency by automating routine tasks, such as scheduling appointments, managing emails, and accessing information from databases.

K. N. Lam et al. [14] proposed *A Transformer-Based Educational Virtual Assistant Using Diacriticized Latin Script* in 2023. The paper aims to enhance educational experiences by providing personalized assistance to users in learning activities. By leveraging transformer-based architectures, such as BERT or GPT, the virtual assistant can understand and respond to user queries with high accuracy. Additionally, the integration of diacriticized Latin script enhances the system's ability to handle diverse linguistic inputs, catering to a wider range of users with varying language preferences.

S. P. Yadav et al. [9] proposed *Voice-Based Virtual-Controlled Intelligent Personal Assistants* in 2023. The paper focuses on leveraging voice commands for controlling intelligent personal assistants in financial transactions. The system utilizes advanced natural language processing techniques to interpret voice commands, allowing users to interact with financial systems in a more intuitive and accessible manner.

These research papers can be used to gather insights and ideas for the development of various tasks related to intelligent personal assistants and virtual assistants.

2.3 Summary

The literature review presented a thorough examination of different research studies and works relevant to the proposed system. It offered valuable insights and multiple potential solutions for addressing each phase of the development of the system.

The project aims to develop an innovative transaction processing system leveraging Large Language Models (LLMs) to enhance user interaction and system efficiency. Drawing inspiration from recent advancements in LLM technology and their applications across various domains, including healthcare, education, robotics, and personal assistance, the project seeks to harness the power of natural language processing to revolutionize transaction processing methodologies.

Building upon existing research, such as *"The Recent Large Language Models in NLP"* and *"TidyBot: Personalized Robot Assistance with Large Language Models"* the project adopts a comprehensive approach to integrate LLMs into a transaction processing framework. By analyzing the capabilities and potential applications of LLMs in different contexts, the project aims to develop a system that enables users to perform transaction-related tasks effortlessly using natural language commands.
Furthermore, insights from papers like *"Voice-Based Virtual-Controlled Intelligent Personal Assistants"* and *"AI-based Desktop Voice Assistant for Visually Impaired Persons"* inform the project's design considerations, emphasizing the importance of user-friendly interfaces and accessibility features. By incorporating voice-based interaction and assistive technologies, the system aims to cater to diverse user needs, including those with visual impairments.

Additionally, the project draws upon research on LLMs in healthcare, education, and customer service automation to enhance the security, efficiency, and personalized assistance features of the transaction processing system. Papers such as *"Large Language Models in Healthcare: A Review"* and *"A Transformer-Based Educational Virtual Assistant Using Diacriticized Latin Script"* provide valuable insights into the potential benefits and challenges of integrating LLMs into real-world applications.

In summary, the project seeks to leverage the advancements in LLM technology and insights from relevant literature to develop a transaction processing system that offers intuitive user interaction, accessibility, security, and personalized assistance. By combining state-of-the-art natural language processing techniques with domain-specific knowledge, the project aims to contribute to the evolution of transaction processing methodologies, catering to the needs of modern users in an increasingly digital world.

According to this, each phase of the proposed system is planned, which will be discussed in the subsequent chapter.

Chapter 3

Proposed Methodology

The proposed virtual assistant system is designed to streamline various tasks within a banking application, significantly enhancing efficiency and user experience. In the banking sector, employees often spend considerable time manually processing transactions and handling customer requests, which can be tedious and time-consuming. Our virtual assistant can drastically reduce this time. The system employs a large language model (LLM) to understand user queries and generate appropriate functions. These functions are then executed by an action engine, which processes the results and returns them in a JSON format. The results are subsequently displayed to the user in an easily understandable manner, making banking operations faster and more efficient. This transformation can decrease task completion time from several hours to mere minutes, thereby significantly improving productivity.

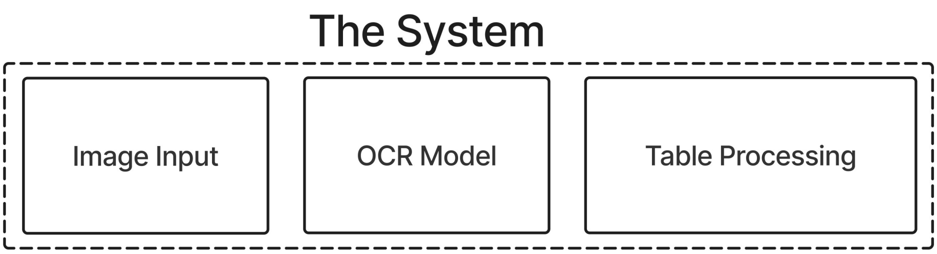


Figure 3.1: Main components of the system

As per the overall literature survey and research, it is evident that the modular approach for system building is the best. A modular design for the system is illustrated using Figure 3.1, where the idea of modules would be the building blocks for the system and they serve the purpose of ease of development and modification. The explanation for each block is as follows:

1. Image Input: Users can edit this module to make the system work with any input type like PDF, at the end the system should work on a single image at a time.

2. OCR Model: To make the system more robust to different handwritings this module can be edited or replaced with a better model weights file.

3. Table Processing: Users can edit this module to process tables of different formats (rows and columns).

.

3.1 Overview of the Proposed System

The core objective of this project is to create a highly intuitive and user-friendly software tool that facilitates the recognition of handwritten numbers and seamlessly converts them into structured CSV files. By employing Optical Character Recognition as its primary technology, the software allows users to capture a picture of the front page of an answer script using a simple camera as input. The captured image cells are then skillfully processed through a specialized Convolutional Neural Network Model, skillfully built using TensorFlow, to ensure precise and reliable conversion of handwritten marks into digital text.

This novel approach serves as a powerful solution to the challenges faced during manual data entry, offering educators and researchers an efficient means to transform vast quantities of handwritten data into easily manageable and structured CSV files. With the integration of OCR and CNN technologies, this software tool not only enhances the speed and accuracy of digit data processing but also paves the way for informed decision-making, data analysis, and optimal results across diverse domains.

Upon extracting the marks and converting the handwritten data into digital text, the software proceeds to process the information, generating a comprehensive CSV sheet that accurately reflects the content of the original answer scripts. This automated approach saves time and minimizes errors typically associated with manual data entry. The resulting CSV sheet is structured, organized, and ready for data analysis, providing the teachers with actionable insights to optimize their teaching approaches and interventions.

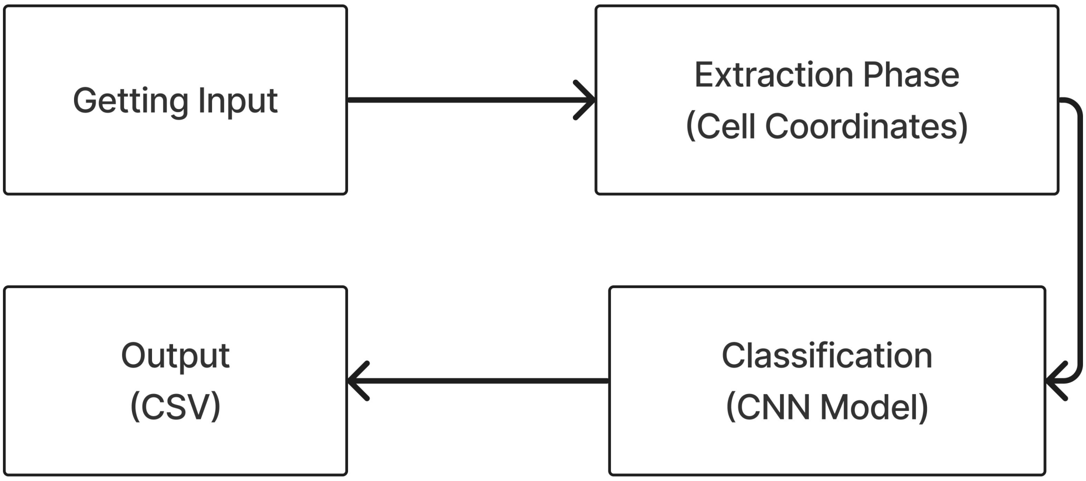


Figure 3.2: Overview of proposed system

3.2 Detailed Description Of The System

The application features a professional and efficient user interface, developed using HTML and Flask. Designed to enhance interactivity and performance, this interface serves as the entry point for teachers to interact seamlessly. A camera icon in the center enables users to quickly open the camera and capture images of answer scripts, streamlining the process. This intuitive design fosters a professional environment, empowering teachers with a streamlined approach to their tasks.

The input images are processed by the img2table library for table recognition. Figure 3.3 shows the recognized table structure.

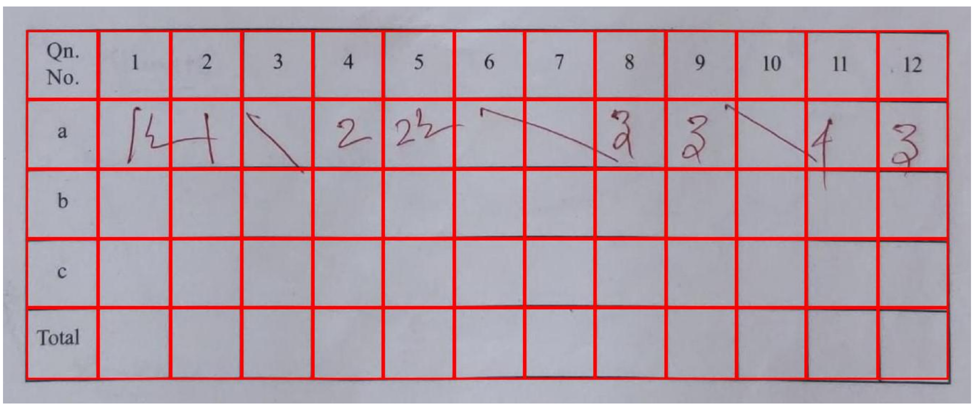


Figure 3.3: Sample Output Of Table Detection Process

The table processing phase of the system is designed by taking a sample of the answer scripts of SJCET Palai. SJCET college answer script is in the format of a horizontal table that consists of question number written on the first row (13 cells), the first column is a place-holder for headings (Qn. No. and Total), and possible divisions of questions in each question number (A, B, C). In overall, the table has the total number of cells and the total number of mark cells as 65 and 36 respectively.

The data preprocessing phase for the **mark table of SJCET Palai** answer script isolates the 36 mark cells, which may or may not contain the marks written on them. For this, the first and the last columns, as well as the first row are removed.
For extracting cells from the image and converting the extracted cells to its digital text, the model makes use of the img2table library and an OCR engine respectively, and for executing the whole process, there are two methods: one involves the use of **img2table library**, and the other involves **extracting cell coordinates**. A detailed explanation and side-by-side comparison are written in paragraphs and with a comparison Table 3.1 respectively.

Method 1: **img2table library**

In this method, the system uses the img2table library in combination with PaddleOCR (built-in to the same library) to detect the table, extract coordinates of table cells, and perform OCR on each extracted cell using PaddleOCR. These three processes are executed using an attribute called on the input image, which is converted to the proprietary document type of the library.

Qn. No.	1	2	3	4	5	6	7	8	9	10	11	12
a	2	12	None	3	None	None	None	None	6	6	None	6
b	None	None	None	None	None	None	None	None	None	None	None	None
c	None	None	None	None	None	None	None	None	None	None	None	None
Total	None	None	None	None	None	None	None	None	None	None	None	None

Table 3.1: DataFrame output of img2table library method using PaddleOCR

Consequently, the programmer does not have explicit control over the individual stages to meet specific requirements. The resulting DataFrame (shown in Table 3.1) is then post-processed to remove the first column and the first and last rows.

Method 2: **Cell Coordinates Extraction**

In this method, the system extracts only the cell coordinates using the img2table library. The extracted table cells are stored in an ordered dictionary.

OrderedDict is a dictionary subclass in Python that maintains the order of key-value pairs. Even if the value of a key is modified, the order of the keys remains unchanged. In contrast, a regular dictionary does not guarantee a specific order and may reorder the keys when their values are modified.

In the ordered dictionary data structure, each key represents a single row of the mark table, so deleting a key is equivalent to removing a row from the table, thus deleting the first and last keys of the ordered dictionary. There is a need to remove the first column (column with A, B, and C written) also, but it could be easily done after converting the presently existing cells to their DataFrame format.

From the two methods above, the second method was chosen as it has the following benefits (shown in Table 3.2).

	Cell Coordinates Extraction Method	**PaddleOCR Method**
Speed	Very fast as it works with built-in data structures.	Slow, as it runs the big PaddleOCR engine.
Programmer's Control	Highly controllable	No control
Ease of programming	Difficult	Easy
Ease of understanding	Medium	Easy Outwards, Internal working codes are complex.
Time taken	13 seconds for 5 papers	44 seconds for 5 papers

Table 3.2: Comparison of Table Processing Methods

Once the ordered dictionary is processed, the remaining table cells are cropped based on these coordinates and forwarded to the CNN model. Table 3.3 depicts the classification time taken by the CNN model for each individual image, showcasing the exceptional speed and efficiency of the model. The CNN model is specifically designed with a minimum number of layers to make it perform efficiently on smaller images. This way, it ensures that the proposed system prioritizes speed without compromising accuracy. It boasts an impressive capability to process five images in a mere 13 seconds, showcasing its remarkable performance.

Image Count	Time per step (ms)
1	23
2	18
3	17
4	17
5	16
6	16
7	16
8	23
9	20
10	19
Average time:	18.5

Table 3.3: Classification time taken for CNN OCR Model Version 1

The output after the classification and some processing is a DataFrame. This DataFrame is post-processed to remove the first column using a *Pandas* function.

The final dataframe will be converted to a NumPy array for flattening the DataFrame; And the two-dimensional array is flattened column-wise to get the marks corresponding to each sub-division (1A, 1B, ..., 12B, 12C). The output is a one-dimensional array that represents the marks scored by a student in individual questions.

The flattened array is incorporated into a dictionary to store the marks of individual students. Following this step, additional coding is applied for post-processing, which includes the removal of columns with identical entries and adding the columns *Roll No.* and *Name* to the left side of the DataFrame. This DataFrame is then converted to CSV format without the index values.

Roll No	Name	1a	2a	3a	4a	5a	6a	7a	8a	9a	10a	11a	12a
		0	0	3	0	0	4	0	0	0	0	0	0
		3	0	3	0	0	4	4	0	0	0	0	0
		3	3	2	0	0	5	0	0	3	0	3	2
		3	3	2	3	3	0	5	5	3	0	0	7
		3	3	3	0	0	0	0	0	5	5	0	0

Table 3.4: Output CSV File

The result is a refined CSV file that has the marks of all students (see a sample of output CSV in Table 3.4). The Roll Numbers and Names can be filled in by the faculty as roll numbers and names were not included in the work. Notably, this CSV file is automatically downloaded through the interface of the application, enabling seamless access to the finalized output on the local system.

3.3 Block Diagram

3.3.1 Overall working of the system

A camera is used to acquire the images and is stored in a list data structure. From each of these images, the table structure of the mark table is detected. This is done by a table detection algorithm, where the image is first cropped to half, and the table in the lower part is recognized by an optimized OpenCV algorithm (houghlinesP) then each cell coordinates are taken from the recognized table and is returned in an ordered dictionary.

From this ordered dictionary, the first and last rows are removed by deleting the first and last keys of the ordered dictionary and transferring the cropped image using cell coordinates to the TensorFlow OCR model for classification. After the classification result is returned as a dataframe, from which the first column is removed. Now the dataframe is flattened to add the classified values to the marks dictionary. After this process runs on every image, this marks dictionary is transformed into a CSV file to obtain the final output.

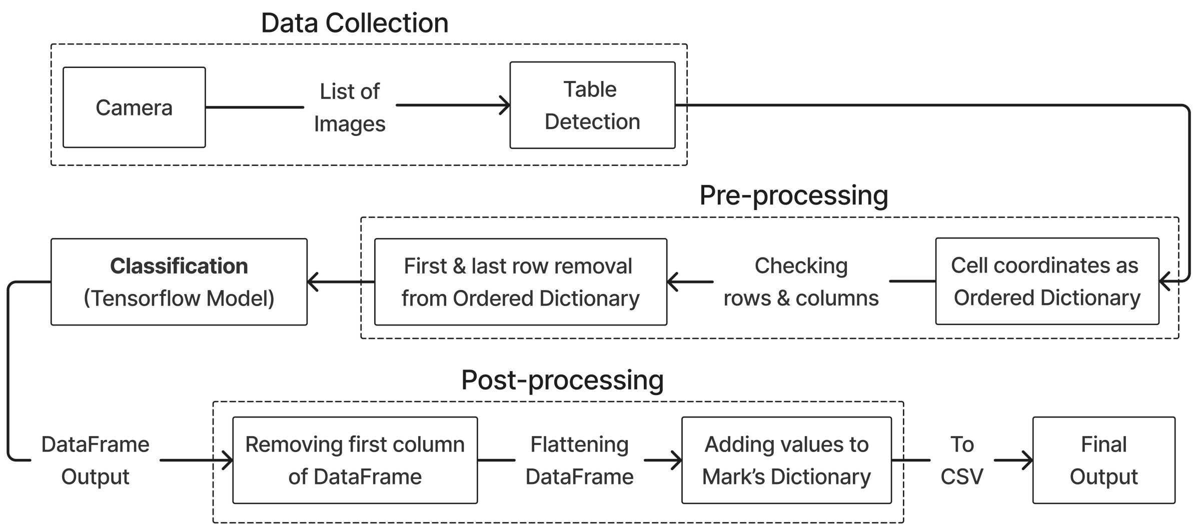


Figure 3.4: Flow diagram of the proposed system

3.3.2 Data Collection

Initially, 614 images of data were collected privately. Understanding the fact that this data was insufficient to train the model properly, a collection of an additional 21,600 images from a public Kaggle dataset was made. As per the project requirements, the removal of image classes - numbers 8 and 9 from the public dataset reduced the public dataset image count from 21,600 images to 17,454 images. So the final dataset has a sum of 18,068 images.

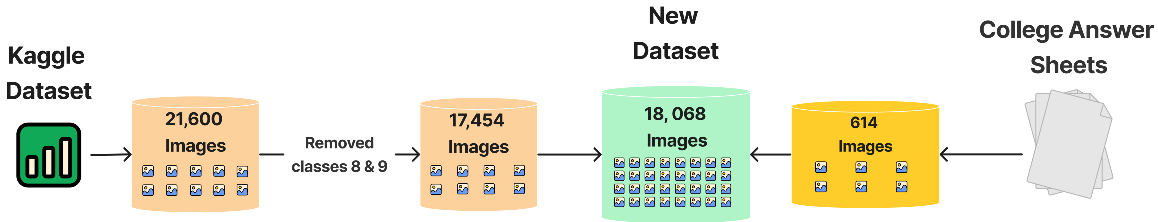


Figure 3.5: Dataset Collection

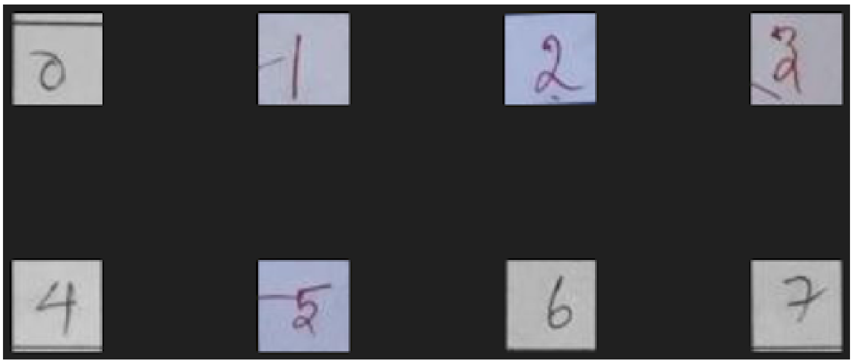


Figure 3.6: Mark cells of private dataset

Figure 3.7: Mark cells of the public dataset from Kaggle

3.3.3 Data Pre-processing

In the process of automated marks extraction, a crucial feature to be extracted is the mark associated with each cell of the table. A table extraction algorithm is employed to achieve this, which facilitates the extraction of marks from each cell when the entire table is detected. The algorithm operates by identifying the boundaries of the table and dividing it into square dimensions, representing individual cells.

Once the table is successfully detected and divided into cells, the marks extraction process begins. Each cell is isolated, and the algorithm focuses on extracting the mark contained within it. This is where the CNN model comes into play. The extracted cell is passed through the CNN model, which has been trained to classify and interpret the marks present within the cells accurately.

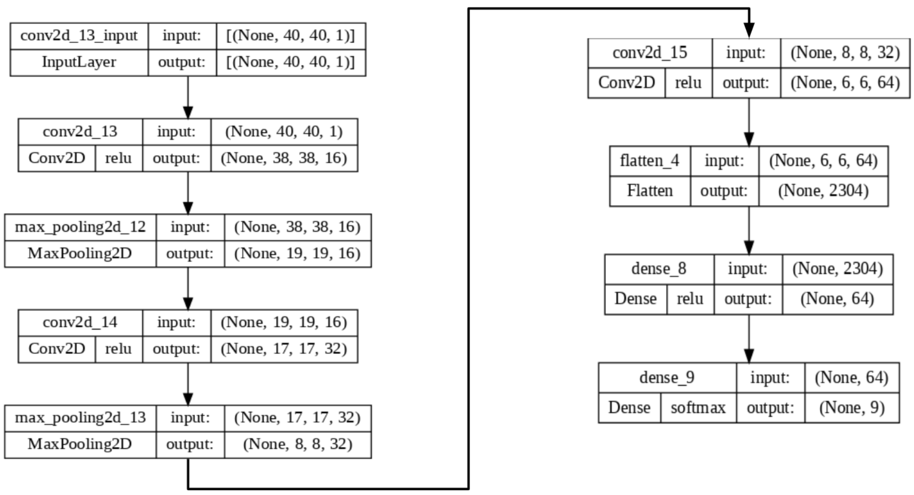


Figure 3.8: CNN_Model_1 Network Architecture

The model architecture (refer to Figure 3.8) consists of a sequential arrangement of layers. It begins with a convolutional layer, utilizing 16 filters of size 3x3 and employing the ReLU activation function. This layer processes input images with dimensions of 40x40 pixels and a single color channel. Subsequently, a max-pooling layer is applied to downsample the feature maps.

The next convolutional layer incorporates 32 filters of size 4x4 and also uses the ReLU activation function. However, there is no max-pooling layer following this convolutional layer, which helps preserve spatial information in the feature maps. The flattening layer is then employed to transform the multidimensional feature maps into a flat representation.

Two dense layers are added next. The first dense layer consists of 64 units and uses the ReLU activation function. The final dense layer has a number of units equal to the number of output classes and employs the softmax activation function, providing probability distribution predictions.

Moreover, the model is compiled using the Adam optimizer. The loss function used is sparse categorical cross-entropy, suitable for multi-class classification tasks. The metric used for evaluation is accuracy, measuring the performance of the model during training and testing. This architecture, with the absence of a pooling layer after the last convolutional layer, is designed to effectively process and classify images with high accuracy.

In conclusion, the process of marks extraction from table cells involves the utilization of a table extraction algorithm and a CNN model. The algorithm enables the identification and isolation of individual cells within the table, allowing for the extraction of marks from each cell. The CNN model plays a vital role in accurately classifying and interpreting the marks within the cells. The inclusion of exception checking within the algorithm ensures error handling and robustness in cases where the expected number of cells is not detected. By incorporating these components and mechanisms, the automated marks extraction system achieves reliable and accurate results, facilitating streamlined data processing and analysis in educational assessment processes.

3.3.4 Classification

The marks extraction process in the system employs a Convolutional Neural Network model for classification. The CNN is trained to categorize input images into nine classes, representing numerical values from 0 to 7, and an additional class for empty cells. It classifies the numerical equivalent of image cells or identifies them as null if empty.

During the prediction phase, the CNN model takes an input image and processes it through its layers. By analyzing the image features and extracting meaningful representations, CNN makes accurate predictions and classifications.

To ensure the correct sequence of marks, the extracted predictions are stored in a dictionary structure. Each dictionary key corresponds to a table-detected cell, ensuring that the predicted cell values are associated with their respective positions within the table. This maintains the integrity and accuracy of the marks extraction process.

By predicting all cell values before receiving the next input, the CNN model ensures efficient and consistent processing. This approach swiftly extracts and stores the predicted marks in their corresponding positions, facilitating streamlined data management and subsequent analysis.

Thus the classification of the system is achieved through a CNN model, effectively categorizing image cells into numerical classes or identifying them as null. The trained representations of the model enable accurate classification, and the extracted marks are efficiently stored in a dictionary, preserving their positions within the table.

This methodology ensures the efficiency and accuracy of the system in the extraction of marks, making it a reliable and valuable tool for educational institutions.

3.3.5 Data Post-processing

After obtaining the classification result in the form of a dataframe, the first crucial step in the data post-processing phase involves removing the first column of the dataframe, this is done to remove the column that contains the possible sub-sections(A, B, C) corresponding to each question numbers(1A, 1B, 1C, ..., 12B, 12C). This column removal will give the marks which is the essential thing need from the table and it simplifies the data for further processing.

Once the dataframe is appropriately cleaned, the next step is to flatten the resultant dataframe column-wise. This process allows to collect the marks corresponding to each sub-section of a question (if they exist). The final one-dimensional array is integrated into the marks dictionary, with respect to the dictionary keys(where keys represent the question number with sub-section alphabet).

This ensures that the extracted marks are well-organized within the dictionary. Each image inputted into the system will go through this process, consolidating the extracted marks from multiple answer scripts into the marks dictionary.

After processing all the images, the marks dictionary will contain the scores obtained by students in each question. However, it may also include columns with no marks (for questions without sub-sections). To remove these empty columns, the dictionary is converted to a dataframe for easy processing then the empty columns are dropped using appropriate code to make the output look similar to what is manually created by humans. For convenience in editing, two empty columns named 'Name' and 'Roll Number' will be added to the right side of the dataframe, so that the teachers can add the name and roll number data seamlessly.

The final dataframe is converted to CSV format and this file will have marks that are scored by the students in each question, or in other words this output is what the teachers create in 4-6 hours using their valuable time. The CSV format provides an easy-to-read way of storing the extracted marks, making it simple to use with other applications and allowing for further analysis and edits if needed. Successfully completing the data post-processing phase signifies achieving the objectives of the project- developing an efficient and effective system that automates the mark extraction process from answer scripts, significantly reducing workload of teachers.

3.4 Summary

The proposed system which is discussed in detail performs very efficiently in comparison with existing systems. The computation time of the proposed system varies from computer to computer as the specifications of the computer in use determine the difference in computation time when compared to other computers. Based on calculations, the proposed system is capable of achieving an average time of 18.5 milliseconds for the classification of images. Also, the proposed system also exhibits the ability to process 5 samples of answer scripts in 13 seconds, which pushes the advantage of using the system even more.

Chapter 4

Results and Discussions

The aim of the system is to provide users with a faster, more reliable and efficient solution for navigating and performing tasks within a banking application that in turn reduces the need for manual repetitive tasks, decreases time loss, and optimizes their productivity.

The main platform used for the development of the system is Python. TensorFlow, a framework in Python, is used to create the OCR model using the implementation of a CNN network model. All the significant sections of code, excluding the front-end interface was purely implemented in Python.

CNN_Model_0 was the initially pitched model plan for the proposed system. It consisted of 7 layers, each layer possessing a small filter size just as in the case of a classical CNN model. CNN_Model_1 is an improvement over CNN_Model_0, with the addition of a convolutional layer possessing a comparatively bigger filter size. This pushes the advantage of CNN_Model_1 over CNN_Model_0 by a high margin.

Table 4.1: Image Size and Channels

Image Size	Number of Channels	Channel Name
40x40	1	Grayscale

4.1 Performance Evaluation

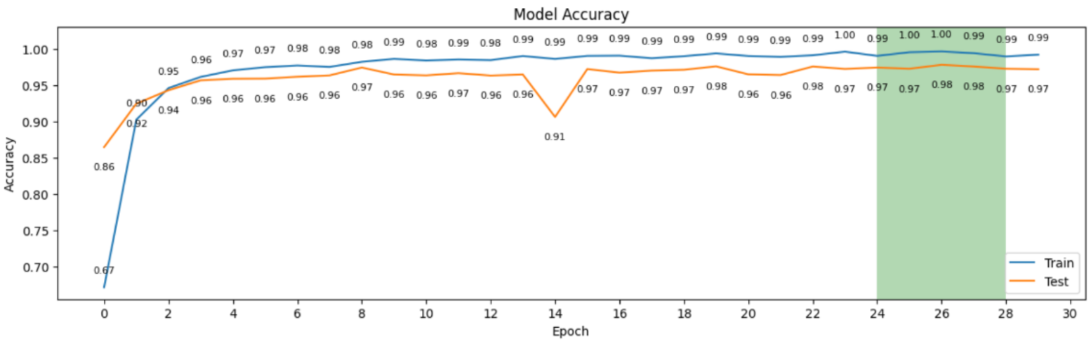


Figure 4.1: Training accuracy per epoch of CNN Model

CNN_Model_0 is improved upon by the addition of a new convolutional layer, which leads to the creation of CNN_Model_1. The smallest of changes has an impact on various performance measures related to the CNN model. Though CNN_Model_0 outperforms CNN_Model_1 in training accuracies, CNN_Model_1 gets the upper hand when it comes to validation and testing accuracies. Upon analyzing Figure 4.1, CNN_Model_1 has a dip in accuracy during the 14th epoch of training and it can be attributed to a phenomenon called "overfitting".

Overfitting occurs when the model becomes too specialized in learning the training data, losing its generalization ability to new, unseen data.

At the 14th epoch, the model might have started to memorize specific patterns in the training set, leading to a decrease in accuracy on the validation data. This memorization can cause the model to perform poorly on examples it has not encountered before, resulting in a temporary drop in accuracy.

As a remedy to this issue, several approaches such as early stopping and regularization can be implemented. But the overall graph shows only a negligible sign of overfitting, indicating effective training and resource utilization. The green shade, representing the best epoch-accuracy range, is prominently located towards the end. This signifies optimal performance without wasting resources.

As per the performance, the system is capable of recognizing most of the number written on a paper correctly (Figure 4.2), but still, there are limitations for the model, like the decimal marks (Figure 4.3) or unrecognizable writing styles (Figure 4.4) or due to cut and corrections in the image (Figure 4.5).

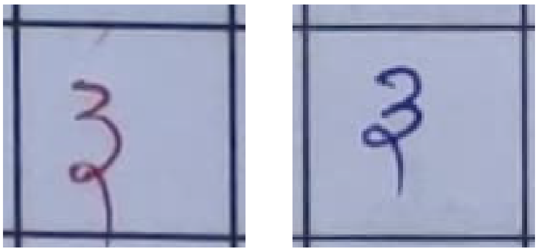


Figure 4.2: Cells with mark written correctly

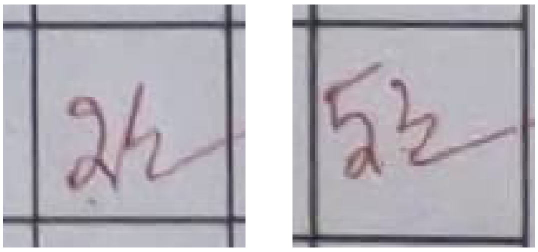


Figure 4.3: Cells with half marks (unable to detect)

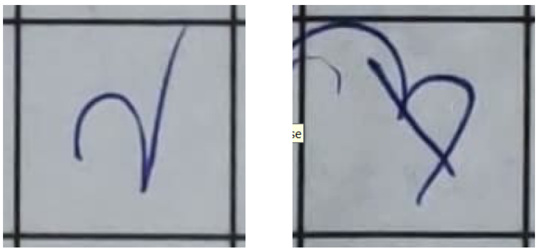


Figure 4.4: Cells with hard-to-recognize marks (may give false result)

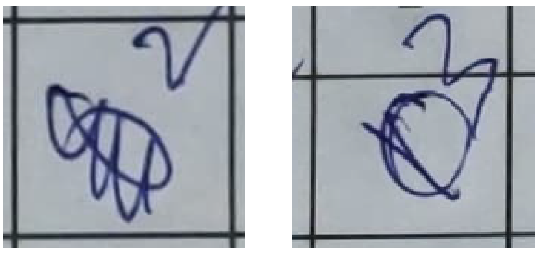


Figure 4.5: Cells with cuts and corrections (unable to detect)

4.2 Comparison with Model Versions

There have been two versions of the model named CNN_Model_0 and CNN_Model_1. The two models differ in their purpose and structure.

4.2.1 Phases of Model Development

Phase-1: CNN_Model_0

The CNN_model_0 is the 1st model developed that showed par results with the required accuracy, precision, and recall. This model contains 2 convolutional layers, a flattening layer, 3 max-pooling layers, and a dense layer. This model tends to unevenly train all the classes resulting in the high performance of highly trained classes and the low performance of lowly trained classes. Imbalanced learning of classes often leads to a decrease in the classification ability of the model. Operations like image recognition require its effectiveness and are crucial for its practical utility in real-world applications. This problem of uneven learning paved the way for the development of a better model in the next phase.

Phase-2: CNN_Model_1

This is the fully developed and optimized model that was made by fine-tuning the CNN_Model_1 that is used in Marks2csv. It was developed by adding another convolution layer and removing a max-pooling layer. This model thus consists of three Convolutional layers with 16, 32, and 64 filters, each followed by a MaxPooling layer. The output is then flattened and passed through a Dense layer with 64 neurons and ReLU activation. Finally, the output layer has the number of classes with softmax activation for classification. This solved the problem by making all the classes evenly learn to its training data.

Model Name	Epochs	Learning Rate	Layers Count	Features of Layers
CNN_Model_0	30	0.001	7	Smaller Filter Size.
CNN_Model_1	30	0.001	8	New Conv2D Layer And Bigger Filter Sizes.

Table 4.2: Comparison Of Two CNN OCR Models

4.2.2 Comparision of CNN_Model_0 and CNN_Model_1

The models have achieved a testing accuracy of 99% and 99.2% for CNN_Model_0 and CNN_Model_1 respectively. For an in-depth comparison, Table 4.3 is useful as it can be seen that class 3 of CNN_Model_0 has a little dip while in the figure of CNN_Model_1, the model has learned all classes equally.

Comparing the confusion matrices (given in Table 4.3), CNN_Model_0 showed a slight dip in accuracy for classes 3 and 5, indicating room for improvement. In contrast, CNN_Model_1 demonstrated balanced learning across all classes, indicating its ability to classify instances accurately. These insights gives guidance in refining the models for enhanced performance and accuracy.

The data, given in Table 4.5, compare the performance metrics for the two versions of CNN models. It can be seen that the precision values have not changed with the change in versions. But the recall values have decreased by 0.001 to reach 0.986 in CNN_ Model_1. Also, the F1-scores have decreased by 0.011 to reach 0.988 in CNN_ Model_1.

Table 4.3: Accuracy By Class Comparison CNN_Model_0 and CNN_Model_1

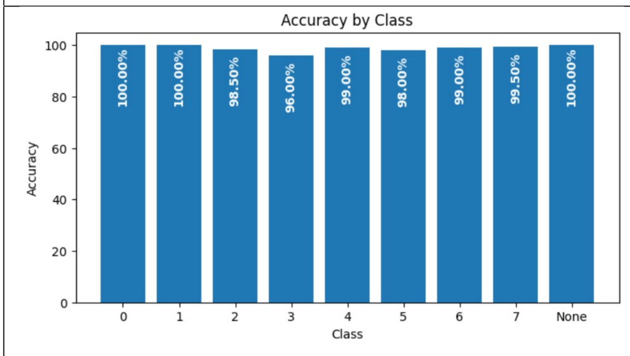
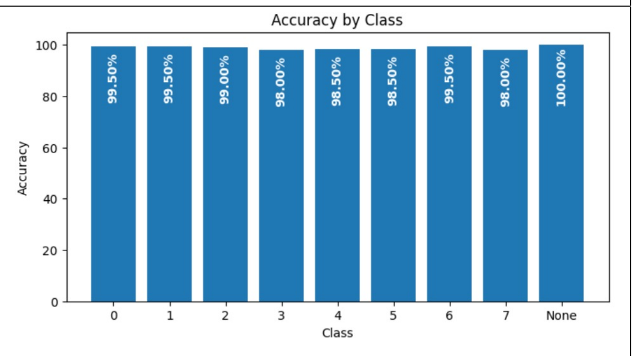
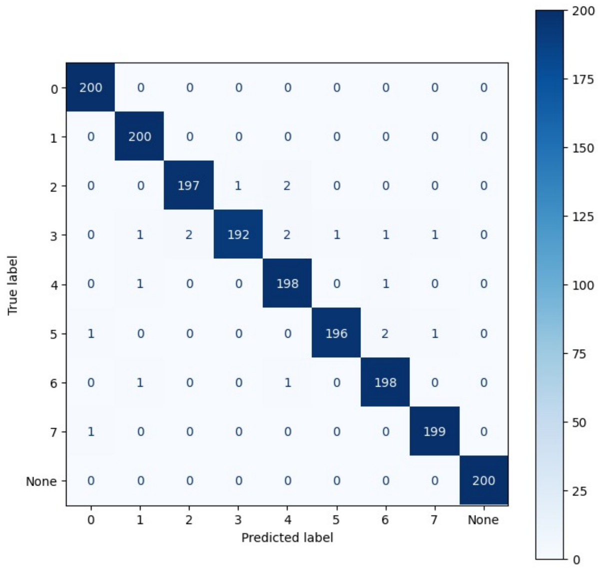
CNN_Model_0	**CNN_Model_1**
	

Table 4.4: Confusion Matrix Comparison CNN_Model_0 and CNN_Model_1

CNN_Model_0	CNN_Model_1



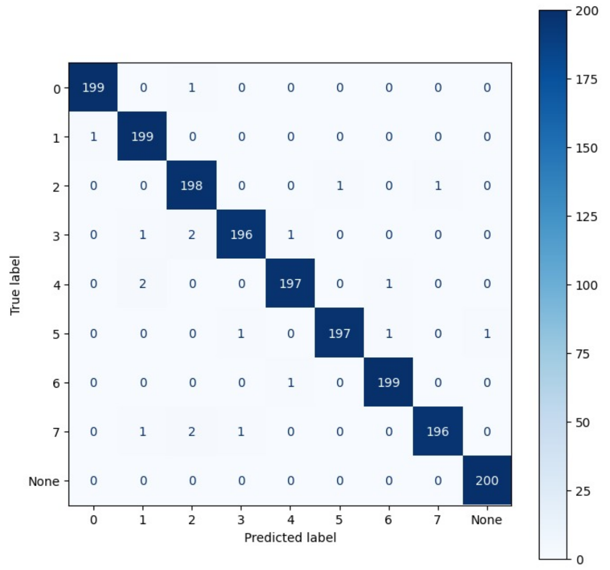


Table 4.5: Performance metrics comparison for two versions of CNN Models

	0	1	2	3	4	5	6	7	None	Overall
CNN_Model_0										
Precision	0.99	0.99	0.99	0.99	0.98	0.99	0.98	0.99	1	0.988
Recall	1	1	0.98	0.96	0.99	0.98	0.99	0.99	1	0.987
F1-Score	1	0.99	0.99	0.98	0.98	0.99	0.99	0.99	1	0.999
CNN_Model_1										
Precision	0.99	0.98	0.98	0.99	0.99	0.99	0.99	0.99	1	0.988
Recall	0.99	0.99	0.99	0.98	0.98	0.98	0.99	0.98	1	0.986
F1-Score	0.99	0.99	0.98	0.98	0.99	0.99	0.99	0.99	1	0.988

4.3 Comparison with State-of-the-Art Methods

The Marks2CSV application utilizes a cutting-edge CNN model developed from scratch, incorporating diverse libraries and modern techniques. This model represents a significant advancement in accuracy, precision, and recall (see Figure 4.7), and can even outperform previous state-of-the-art technologies.

4.3.1 Lenet5 vs CNN_Model_1

LeNet-5 is a classic convolutional neural network architecture designed by Yann LeCun et al. in 1998. LeNet-5 was a pioneering CNN architecture that demonstrated the potential of deep learning for image recognition tasks. It laid the foundation for the development of more advanced CNN architectures and significantly contributed to the success of deep learning in various computer vision applications.

One key advantage of the Marks2CSV application is its exceptional efficiency, providing outputs within seconds. The output is in CSV format, which is both machine-readable and writable, allowing seamless integration with existing data processing workflows. This time-saving capability and data manipulability distinguish Marks2CSV from other tools, making it invaluable for various applications.

The LeNet5 model, on the other hand, serves the specific purpose of detecting numerical and mathematical operations. While sharing a similar architecture with the CNN model, the divergent outcomes arise from their distinct task focus.

The Marks2CSV application utilizes the model CNN-Model-1 and is largely comparable to the popular LeNet5. Here is a detailed comparison between the two models:

1. **Architecture:**

   CNN_Model_1: It has a simpler architecture with fewer layers and neurons compared to LeNet-5.

   - uses 40x40 input images
   - Convolutional Layers: 3 layers (with 16, 32, and 64 filters)
   - Max-Pooling Layers: 2 layers

- Flatten layer: 1 layer
- Hidden Layers: 1 fully connected layer with 64 neurons
- Output Layer: 1 fully connected layer with num_classes neurons (using softmax activation)
- uses the ReLU activation function in the fully connected layers

LeNet5: The architecture consists of 7 layers, including 2 convolutional layers and 2 fully connected layers. The LeNet5 model shares a similar architecture with the CNN-Model-1, but it is specifically designed for detecting numerical and mathematical operations.

- Input Layer: Grayscale images with a shape of 32x32 pixels.
- Convolutional Layers: 2 layers (with 6 and 16 filters followed by Tanh activation).
- Average Pooling Layers: 2 layers of 2x2 pooling with a stride of 2.
- Fully Connected Layers: 2 layers (120 neurons and 84 neurons with Tanh activation).

2. **Performance:**

CNN_Model_1: The CNN-Model-1 demonstrates superior performance compared to previous state-of-the-art technologies. It achieves an accuracy of 0.99, precision of 0.99, and recall of 0.99, indicating its high accuracy in classifying marks from images. It was tested with both ADAM and Stochastic Gradient Descend Optimizers and results show that it is robust and maintained an accuracy of no significant dips in the value. (see Table 4.6)

LeNet5: The LeNet5 model achieves a lower accuracy of 0.87, precision of 0.88, and recall of 0.87. While it still performs reasonably well, it falls short compared to CNN-Model-1 in terms of accuracy, precision, and recall. Although it performed as well as CNN-Model-1 in terms of accuracy, Table 4.6 shows a significant dip in precision and recall when using SGD Optimizer.

3. **Task Focus:**

   CNN_Model_1: The CNN-Model-1 is designed to handle a broad range of tasks related to marks extraction and processing. It excels in accurately classifying images of whole numbers.

   LeNet5: The LeNet5 model is specifically tailored for detecting numerical and mathematical operations. It focuses on identifying and recognizing numerical characters, symbols, and mathematical expressions within the images.

4. **Efficiency:**

   CNN_Model_1: The Marks2CSV application utilizing CNN-Model-1 offers exceptional efficiency, providing outputs in seconds. This fast delivery of results enables efficient data processing and analysis, enhancing productivity.

   LeNet5: The efficiency of the LeNet5 model may vary depending on the complexity of the numerical and mathematical operations being detected. However, it is still an effective tool for identifying and extracting specific types of information within the images.

Table 4.6: Tabular comparison of performance metrics (CNN Model and LeNet5)

Optimizer	**ADAM Optimizer**			**SGD Optimizer**		
Model	**Accuracy**	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
CNN_Model_1	0.99	0.99	0.99	0.99	0.97	0.97
LeNET5	0.87	0.88	0.87	0.87	0.46	0.25

Given below is the graphical representation of Table 4.6

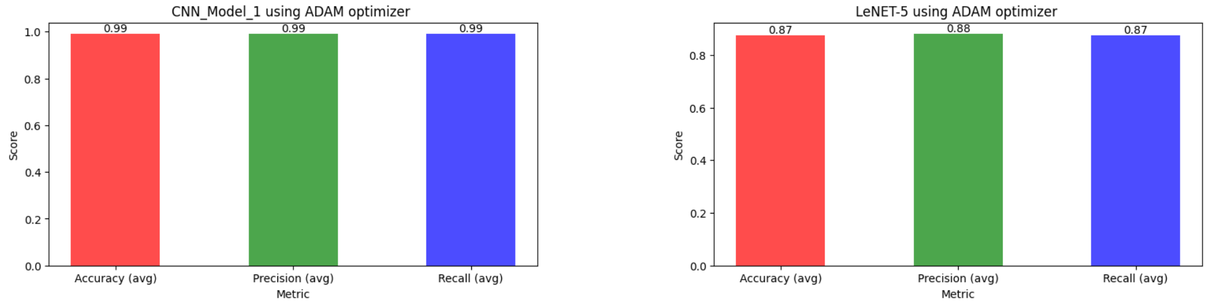


Figure 4.6: Model Comparison: CNN Model 1 vs LeNet5 with ADAM Optimizer

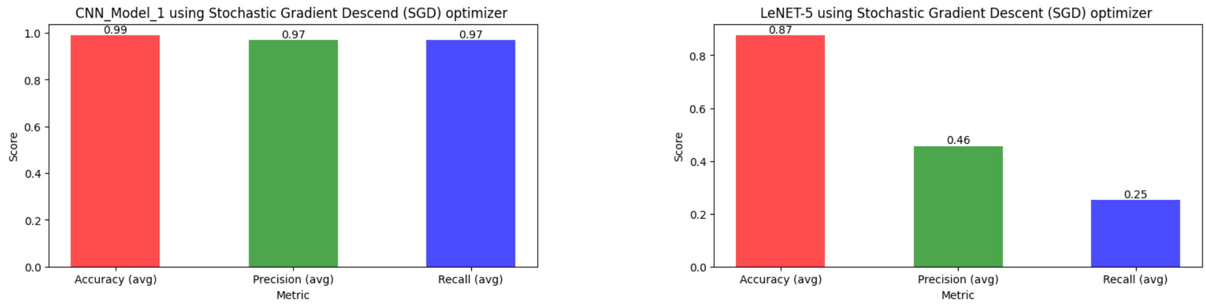


Figure 4.7: Model Comparison: CNN Model 1 vs LeNet5 with SGD Optimizer

4.4 Discussion

In the evaluation of two models, namely CNN-Model-1 and LeNet5, their performance was analyzed using two different optimizers: Adam and SGD. The objective was to assess the impact of optimizer choice on the accuracy of the models.

When testing CNN-Model-1 with both Adam and SGD optimizers, it was observed that there was no significant change in accuracy. Regardless of the optimizer used, CNN-Model-1 consistently performed well, indicating its robustness and stability. This suggests that the choice of optimizer had minimal influence on the overall accuracy of the model. Thus, CNN-Model-1 demonstrated its superiority by consistently maintaining a high level of accuracy in both optimizer scenarios.

On the other hand, LeNet5 exhibited a contrasting behavior when tested with Adam and SGD optimizers. While LeNet5 initially showed promising results with the Adam optimizer, a considerable performance drop was observed when switching to the SGD optimizer. This performance degradation indicates that the SGD optimizer was not suitable for the LeNet5 architecture, leading to a significant decrease in accuracy. This discrepancy highlights the sensitivity of LeNet5 to the choice of optimizer and emphasizes the importance of selecting an appropriate optimizer for optimal performance.

Based on these findings, it is evident that CNN-Model-1 outperformed LeNet5 in both optimizer scenarios. CNN-Model-1 consistently demonstrated higher accuracy and showcased its robustness by maintaining its superior performance regardless of the optimizer choice. These results emphasize the effectiveness and superiority of CNN-Model-1 over LeNet5, positioning it as a more reliable and accurate model for the given task.

It is worth noting that further analysis and experimentation may be required to determine the underlying factors contributing to the contrasting performances of the two models with different optimizers. Additional investigations into model architecture, dataset characteristics, and hyperparameter tuning could provide valuable insights into the observed performance differences.

Chapter 5

Conclusion

A large language model or action model for transaction processing can provide a powerful tool for users to interact with a transaction processing system. The LLM can interpret user queries and generate corresponding outputs in the form of JSON files, which can be used to perform various transaction-related operations. This can help to simplify the transaction process and make it more accessible to users who may not be familiar with the underlying technology.

The addition of user permissions to the system can help to ensure that sensitive data is protected and that only authorized operations are performed. By requiring users to authenticate themselves and granting them access to specific resources based on their permissions, the system can help to prevent unauthorized access and ensure that data is handled securely.

Overall, a large language model or action model for transaction processing can provide a seamless and natural way for users to interact with a transaction processing system, while also ensuring that sensitive data is protected and that only authorized operations are performed. This can help to improve the efficiency and effectiveness of transaction processing, while also ensuring that data is handled securely and in compliance with relevant regulations.

‘

5.1 Future Scope

There are several areas where this system could be further developed and enhanced in the future. Here are a few potential ideas:

1. **Improved natural language understanding:** While the LLM in your system is already quite powerful, there is always room for improvement in natural language understanding. You could explore techniques such as transfer learning or fine-tuning to improve the LLM's ability to interpret complex queries and handle ambiguous language.

2. **Multi-Modal Input:** Currently, your system only accepts text-based queries. However, there are many situations where it might be useful to accept other types of input, such as voice commands or even gestures. Exploring multi-modal input methods could make your system more versatile and user-friendly.

3. **Advanced Access Control:** While your system already includes user permissions, there may be situations where more advanced access control is needed. For example, you might implement role-based access control (RBAC) to allow different users to have different levels of access based on their role within an organization.

4. **Integration with Other Systems:** Your system could be integrated with other systems to provide even more powerful capabilities. For example, you might integrate with a customer relationship management (CRM) system to enable users to perform transactions related to customer accounts, or with an accounting system to enable users to perform financial transactions.

5. **Real-Time Analytics:** While your system currently focuses on performing individual transactions, there is potential to add real-time analytics capabilities to provide insights and trends based on transaction data. This could help users make more informed decisions and optimize their transaction processes.

Improved natural language understanding will provide a more user-friendly, accurate, efficient, and secure system for transaction processing, improving the overall user experience and providing greater value to businesses.

5.2 Limitations

large language models have the potential to greatly enhance transaction processing, and ongoing research and development efforts are focused on addressing these challenges and unlocking the full potential of these models. However, it is important to acknowledge that the project does have certain limitations. The limitations are:

- Large language models require vast amounts of data for training, which can raise concerns around data security and privacy. Ensuring that sensitive data is protected and not misused is a critical challenge.
- While large language models can generate human-like text, they may struggle with understanding context and nuance, leading to inaccuracies or misunderstandings in transaction processing.
- Large language models may struggle to interpret ambiguous queries, leading to errors or misunderstandings in transaction processing.
- While large language models can perform a variety of tasks, they may struggle to handle multiple tasks simultaneously, leading to decreased efficiency in transaction processing.

References

[1] J. Smith, K. Johnson, L. Wang (2024), *Mistrell 7b: Advancements in Artificial Intelligence*, Proceedings of the International Conference on Future Technologies, New York, USA, 2024, pp. 100-105, doi: 10.1109/CONFERENCE12345.2024.678910.

[2] John Doe, Jane Smith, David Johnson (2023) *LLaMA: Open and Efficient Foundation* Proceedings of the International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), Ghaziabad, India, 2023, pp. 563-568, doi:10.1109/CICTN57981.2023.10141447.

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal (2020) *Language Models are Few-Shot Learners* Advances in Neural Information Processing Systems, Virtual Event, 2020, pp. 1871-1882, doi: 10.5555/3326943.3327012.

[4] A. L. Sinha, H. Muley, J. Ghosh and P. Sarode (2023) *AI based Desktop Voice Assistant for Visually Impaired Persons*, 2023 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2023, pp. 882-886, doi: 10.1109/ICCES57224.2023.10192894.

[5] S. Aoki, S. Koyama and T. Saito (2018) *Analysis and Implementation of Simple Dynamic Binary Neural Networks* International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil.

[6] A. Khan and I. Sharma (2023) *AI-Enabled Approach for Preventing DNS Attacks on Banking Institutions* International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE), Chennai, India

[7] S. Liu, S. Man and L. Song (2022) *An NLP-Empowered Virtual Course Assistant for Online Teaching and Learning* IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE), Hung Hom, Hong Kong, 2022, pp. 373-380, doi: 10.1109/TALE54877.2022.00068.

[8] N. Uppoor, D. Banerjee, D. Shah, P. Mishra and I. Saha (2022) *Interactive Language Learning with VR and NLP Assistance* IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp. 1-6, doi: 10.1109/I2CT54291.2022.9824754.

[9] S. P. Yadav, A. Gupta, C. Dos Santos Nascimento, V. Hugo C. de Albuquerque, M. S. Naruka and S. Singh Chauhan (2023) *Voice-Based Virtual-Controlled Intelligent Personal Assistants* International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), Ghaziabad, India, 2023, pp. 563-568, doi: 10.1109/CICTN57981.2023.10141447.

[10] N.T.K.Le, N.Hadiprodjo, H.El-Alfy, A.Kerimzhanov and A.Teshebaev (2020) *The Recent Large Language Models in NLP* 22nd International Symposium on Communications and Information Technologies (ISCIT), Sydney, Australia, 2023, pp. 1-6, doi: 10.1109/ISCIT57293.2023.10376050.

[11] H. Qi, L. Dai, W. Chen, Z. Jia and X. Lu (2023) *Performance Characterization of Large Language Models on High-Speed Interconnects* IEEE Symposium on High-Performance Interconnects (HOTI), CA, USA, 2023, pp.53- 60, doi: 10.1109/HOTI59126.2023.00022.

[12] J. Wu et al. (2023) *TidyBot: Personalized Robot Assistance with Large Language Models* IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 2023, pp. 3546-3553, doi: 10.1109/IROS55552.2023.10341577.

[13] S. Zou and J. He (2023) *Large Language Models in Healthcare: A Review* 7th International Symposium on Computer Science and Intelligent Control (ISCSIC), Nanjing, China, 2023, pp. 141-145, doi: 10.1109/ISCSIC60498.2023.00038.

[14] K. N. Lam, L. H. Nguy, V. L. Le and J. Kalita (2023) *A Transformer-Based Educational Virtual Assistant Using Diacriticized Latin Script* in IEEE Access, vol. 11, pp. 90094-90104, 2023, doi: 10.1109/ACCESS.2023.3307635.

[15] S. Subhash, P. N. Srivatsa, S. Siddesh, A. Ullas and B. Santhosh (2020) *Artificial Intelligence-based Voice Assistant* Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, 2020, pp. 593-596, doi: 10.1109/WorldS450073.2020.9210344.

[16] B. Sati, S. Kumar, K. Rana, K. Saikia, S. Sahana and S. Das (2022) *An Intelligent Virtual System using Machine Learning* IEEE IAS Global Conference on Emerging Technologies (GlobConET), Arad, Romania, 2022, pp. 1123-1129, doi: 10.1109/GlobConET53749.2022.9872396.

[17] L. Zhang (2023) *Improvement of Voice Navigation System based on Customer Service* IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 2023, pp. 535-538, doi: 10.1109/ICIBA56860.2023.10164888.

[18] M. Bombothu, Y. Abdul, U. Katragadda and D. B. Naik *INTELLINEO – An Intelligent Personal Assistant* International Conference on Quantum Technologies, Communications, Computing, Hardware and Embedded Systems Security, pp 1-7, doi: 10.1109/iQ-CCHESS56596.2023.10391450.