A

Project Report on

## MedAI Nexus

Submitted for partial fulfilment of the requirements for the award of the degree of

## BACHELOR OF ENGINEERING

In

## COMPUTER SCIENCE AND ENGINEERING

By

## Mr. Baradi Siddarth Reddy (2451-21-733-067)

## Mr. Kancherla Venkat Sai (2451-21-733-088)

## Mr. Kotha Sai Abhinav (2451-21-733-103)

Under the guidance of

**Mrs. K. Padma**

Assistant Professor Department of CSE



## MATURI VENKATA SUBBA RAO (MVSR) ENGINEERING COLLEGE

## (An Autonomous Institution)

Department of Computer Science and Engineering

(Affiliated to Osmania University & Recognized by AICTE)

Nadergul, Saroor Nagar Mandal, Hyderabad – 501 510

Academic Year: 2024-25

Maturi Venkata Subba Rao Engineering College

(An Autonomous Institution)

(Affiliated to Osmania University, Hyderabad) Nadergul(V), Hyderabad-501510



# Certificate

*This is to certify that the project work entitled “***MedAI Nexus***” is a Bonafide work carried out by* ***Baradi Siddartha Reddy (2451-21-733-067), Kancherla Venkat Sai (2451-21-733-088)*** *and*  ***Kotha Sai Abhinav (2451-21-733-103)*** *in partial fulfilment of the requirements for the award of degree of* ***Bachelor of Engineering*** *in* ***Computer Science and Engineering*** *from* ***Maturi Venkata Subba Rao (MVSR) Engineering College,*** *affiliated to OSMANIA UNIVERSITY, Hyderabad, during the Academic Year 2024-25 under our guidance and supervision.*

*The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma to the best of our knowledge and belief.*

## Internal Guide

K. Padma

Assistant Professor Department of CSE MVSREC.

**Head of the Department** Prof. J Prasanna Kumar Professor

Department of CSE MVSREC.

## External Examiner

# DECLARATION

This is to certify that the work reported in the present project entitled “**MedAI Nexus**” is a record of bonafide work done by us in the Department of Computer Science and Engineering, Maturi Venkata Subba Rao (MVSR) Engineering College, Osmania University during the Academic Year 2024-25. The reports are based on the project work done entirely by us and not copied from any other source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma.

Mr. Baradi Siddartha Reddy Mr. Kancherla Venkat Sai Mr. Kotha Sai Abhinav

(2451-21-733-067) (2451-21-733-088) (2451-21-733-103)

# ACKNOWLEDEGEMENTS

We would like to express our sincere gratitude and indebtedness to our project guide **K. Padma** for his valuable suggestions and interest throughout the course of this project.

We are also thankful to our principal **Dr. Vijaya Gunturu** and **Mr. J Prasanna Kumar,** Professor and Head, Department of Computer Science and Engineering, Maturi Venkata Subba Rao Engineering College, Hyderabad for providing excellent infrastructure for completing this project successfully as a part of our B.E. Degree (CSE). We would like to thank our project coordinator for his constant monitoring, guidance and support.

We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed. We sincerely acknowledge and thank all those who gave directly or indirectly their support in the completion of this work.

Mr. Baradi Siddartha Reddy (2451-21-733-067)

Mr. Kancherla Venkat Sai (2451-21-733-088)

Mr. Kotha Sai Abhinav (2451-21-733

## VISION

* To impart technical education of the highest standards, producing competent and confident engineers with an ability to use computer science knowledge to solve societal problems.

## MISSION

* To make learning process exciting, stimulating and interesting.
* To impart adequate fundamental knowledge and soft skills to students.
* To expose students to advanced computer technologies in order to excel in engineering practices by bringing out the creativity in students.
* To develop economically feasible and socially acceptable software.

## PEOs:

**PEO-1:** Achieve recognition through demonstration of technical competence for successful execution of software projects to meet customer business objectives..

**PEO-2:** Practice life-long learning by pursuing professional certifications, higher education or research in the emerging areas of information processing and intelligent systems at a global level.

**PEO-3:** Contribute to society by understanding the impact of computing using a multidisciplinary and ethical approach.

## PROGRAM OUTCOMES (POs)

At the end of the program the students (Engineering Graduates) will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialisation for the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques,resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of thelimitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions**.**
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these toone’s work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Lifelong learning:** Recognise the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## PROGRAM SPECIFIC OUTCOMES (PSOs)

1. (PSO-1) Demonstrate competence to build effective solutions for computational real-world problems using software and hardware across multi-disciplinary domains.
2. (PSO-2) Adapt to current computing trends for meeting the industrial and societal needs through a holistic professional development leading to pioneering careers or entrepreneurship.

# COURSE OBJECTIVES AND OUTCOMES

## Course Code: PW 881 CS

## Course Objectives:

* + To enhance practical and professional skills.
  + To familiarize tools and techniques of systematic literature survey and documentation.
  + To expose the students to industry practices and teamwork.
  + To encourage students to work with innovative and entrepreneurial ideas.

## Course Outcomes:

**CO1:** Demonstrate the ability to synthesize and apply the knowledge and skills acquired in the academic program to real-world problems.

**CO2:** Evaluate different solutions based on economic and technical feasibility.

**CO3:** Effectively plan a project and confidently perform all aspects of project management.

**CO4:** Demonstrate effective written and oral communication skills.

**CO5:** Present the project using PPT.

# ABSTRACT

Accurately interpreting handwritten doctor prescriptions has long been a critical challenge in healthcare, often resulting in errors and inefficiencies that jeopardize patient safety and hinder care delivery. To overcome this, we present **MedAI Nexus**, an innovative solution designed to detect, recognize, and extract meaningful text from handwritten prescription images. By employing advanced deep learning techniques and Natural Language Processing (NLP), the system highlights essential details such as medicine names, dosages, and instructions, ensuring clarity and accessibility for both patients and healthcare providers.

The system employs a robust architecture that integrates text detection, recognition, and Named Entity Recognition (NER) into a seamless pipeline. Handwritten prescription images are pre-processed and analysed using feature extraction and bounding box detection techniques to localize text regions. A transformer-based encoder-decoder model then processes these regions, transforming raw features into structured text. Advanced post-processing techniques such as noise removal, error correction, and fuzzy matching further refine the output for accuracy and readability.

By leveraging NER powered by contextual embeddings, **MedAI Nexus** identifies and categorizes critical entities, including medication names and dosages, presenting the results in an intuitive and user-friendly format. This functionality not only reduces interpretation errors but also empowers patients with a clear understanding of their prescriptions, improving adherence and healthcare outcomes.

Designed with innovation and usability in mind, **MedAI Nexus** redefines prescription management by automating complex tasks, streamlining healthcare workflows, and enhancing patient safety. The system represents a transformative step in modernizing healthcare operations, offering a powerful tool to ensure precision and efficiency in prescription handling.

.

# TABLE OF CONTENTS

PAGE NOS. Certificate i

Declaration ii

Acknowledgements iii

Vision & Mission, PEOs, POs and PSOs iv

Abstract vii

Table of contents viii

List of Figures x

List of Tables x

**CONTENTS**

CHAPTER I

1. INTRODUCTION 01 – 06

* 1. [PROBLEM STATEMENT 02](#_TOC_250015)
  2. [OBJECTIVE 02](#_TOC_250014)
  3. [MOTIVATION 03](#_TOC_250013)
  4. SCOPE 04
  5. [SOFTWARE AND HARDWARE REQUIREMENTS 05](#_TOC_250012)

CHAPTER II

1. LITERATURE SURVEY 07– 10
   1. [SURVEY OF SMART PRESCRIPTION RECOGNITION 07](#_TOC_250011)

AND TEXT EXTRACTION

CHAPTER III

1. SYSTEM DESIGN 11 – 19
   1. FLOW CHARTS 11
   2. [SYSTEM ARCHITECTURE 11](#_TOC_250010)
   3. [UML DIAGRAMS 14](#_TOC_250009)
   4. [PROJECT PLAN 16](#_TOC_250008)

CHAPTER IV

1. SYSTEM IMPLEMENTATION & METHODOLOGIES 20 – 37
   1. SYSTEM IMPLEMENTATION 20 – 23
   2. DATASET UTILIZATION 23 – 25
   3. HANDWRITTEN PRESCRIPTION RECOGNITION 25 – 27

AND DATA STRCUTURING

* 1. PRETRAINING FOR HANDWRITTEN PRESCRIPTON RECOGNITION 27 – 28
  2. UNIFIED FORMULATION 28 – 30

4.5.1 REVISITING VARIOUS PRESCRIPTION RECOGNITION TASKS 29

4.5.2 PRESCRIPTION CATEGORIZATION AND LABELLING 29 – 30

* 1. UNIFIED MODEL 31 – 33
     1. OVERVIEW 31 – 32
     2. PRETRAINING OBJECTIVES 32 – 33
  2. INFERENCE 33 – 35
  3. INTEGRATION AND DEPLOYMENT 35 – 37

CHAPTER V

1. TESTING RESULTS 38 – 45
   1. EVALUATION METRICS 38 – 39
   2. [PERFORMANCE COMPARISION 39 – 40](#_TOC_250010)
   3. [TEST CASES 40 – 45](#_TOC_250009)

CHAPTER VI

1. CONCLUSION & FUTURE ENHANCEMENTS 46 – 47

REFERENCES 48

APPENDIX 49 – 54

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No.** |
| Fig 3.1 | Data Flow | 11 |
| Fig 3.2 | System Architecture | 12 |
| Fig 3.3 | Use Case Diagram | 14 |
| Fig 3.4 | State Diagram | 15 |
| Fig 3.5  Fig 5.1  Fig 5.2  Fig 5.3 | Sequence Diagram  Test Case I  Test Case II  Test Case III | 16  41  42  43 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Name** | **Page No.** |
| Table 2.1  Table 5.1  Table 5.2  Table 5.3  Table 5.4  Table 5.5 | Survey of the project  Dataset Details  Text Detection Comparison  Text Recognition Comparison  Medical NER Comparison  Test Results | 09  38  39  39  40  45 |

# CHAPTER1

# INTRODUCTION

In the fast-paced world of healthcare, where handwritten prescriptions remain widely used, accurately interpreting these prescriptions is a critical yet challenging task. Misinterpretation of prescriptions can lead to serious consequences, such as incorrect medications or dosages, endangering patient safety and complicating healthcare workflows. For patients, unclear prescriptions result in confusion about their treatment plans, which can further hinder proper medication adherence. Traditional manual methods of deciphering handwritten prescriptions are inefficient, error-prone, and insufficient to meet the demands of modern healthcare.

To address this challenge, we introduce **MedAI Nexus**, an advanced deep learning-based system designed to extract and interpret handwritten prescriptions with precision. The project bridges the gap between prescription images and actionable insights using cutting-edge technologies like Optical Character Recognition (OCR), Natural Language Processing (NLP), and Named Entity Recognition (NER).

MedAI Nexus focuses on accurately identifying essential prescription details, including medicine names, dosages, and instructions, ensuring both healthcare professionals and patients gain clear and accessible information. By automating this process, the system minimizes interpretation errors, enhances prescription clarity, and improves healthcare efficiency. Its robust architecture integrates text detection, recognition, and entity extraction, making it capable of handling even the most complex and noisy handwritten prescriptions.

This project represents a significant leap toward leveraging artificial intelligence in healthcare, improving patient safety, and streamlining prescription management. With its user-friendly interface and high accuracy, MedAI Nexus empowers both healthcare providers and patients, transforming how prescriptions are understood and managed for better outcomes.

## Problem Statement

Handwritten doctor prescriptions remain a significant challenge in healthcare due to their often illegible and inconsistent nature. Misinterpretation of these prescriptions can lead to medication errors, compromising both patient safety and treatment outcomes. For patients, the inability to clearly understand their prescribed medicines, dosages, and instructions further adds to the risk of incorrect medication usage. Relying on manual interpretation is not only time-consuming for healthcare professionals but also leaves patients with limited access to clear and accurate prescription details.

Traditional methods fail to address the complexity of handwritten text and the need for precise extraction of critical information. This highlights the urgent need for an automated, reliable solution that accurately processes handwritten prescriptions, providing clear and accessible information for both patients and healthcare providers. Such a system would improve prescription clarity, reduce interpretation errors, empower patients with a better understanding of their medication, and enhance overall healthcare efficiency. This modernization of prescription handling would ensure better treatment adherence and improved health outcomes.

## Objective

The primary goal of this project is to revolutionize the way handwritten doctor prescriptions are interpreted and managed in healthcare. By implementing **MedAI Nexus**, we aim to provide both healthcare providers and patients with a solution that ensures clarity, accuracy, and accessibility. This objective arises from the challenges posed by illegible and inconsistent handwritten prescriptions, which often lead to errors, inefficiencies, and risks to patient safety.

Through the integration of advanced deep learning techniques, Optical Character Recognition (OCR), and Natural Language Processing (NLP), our project seeks to address these challenges by automating the extraction of critical prescription details such as medicine names, dosages, and instructions. Our ultimate aim is to empower healthcare professionals and patients with a system that transforms raw prescription images into structured, easily

understandable data.

By achieving this objective, we aspire to enhance patient safety, streamline prescription management, and reduce the workload on healthcare professionals. **MedAI Nexus** represents a step forward in modernizing healthcare workflows, ensuring a more efficient, reliable, and user-friendly experience for all stakeholders.

## Motivation

Our motivation stems from the critical need to address real-world challenges in the healthcare domain, particularly the persistent issue of interpreting handwritten doctor prescriptions. Imagine a healthcare system where patients and providers seamlessly understand and manage prescriptions without the risk of errors, delays, or confusion. This vision inspires our commitment to developing **MedAI Nexus**, a solution designed to transform how prescriptions are handled, ensuring safety, clarity, and efficiency for all stakeholders.

In the modern healthcare landscape, handwritten prescriptions remain a cornerstone of medical practice, yet their illegibility often poses significant risks, including medication errors and miscommunication. Patients frequently struggle to comprehend their prescribed medicines, dosages, and instructions, leading to poor adherence and compromised outcomes. This widespread challenge motivates us to create an innovative system that bridges the gap between handwritten prescriptions and accurate, actionable insights, empowering both healthcare providers and patients.

Furthermore, our project is driven by the pressing need for innovation in response to the growing complexities of healthcare workflows. As healthcare systems become increasingly digital, the demand for tools that can seamlessly integrate into existing processes while reducing workloads has never been more critical. By leveraging advanced technologies like OCR, NLP, and NER, we aim to meet this need with a robust solution that not only addresses current challenges but anticipates future demands.

Ultimately, our motivation lies in enhancing patient safety and improving healthcare efficiency by delivering a system that eliminates errors, saves time, and provides accessible prescription information. We are committed to empowering healthcare professionals and patients with a transformative tool that not only meets but exceeds expectations, redefining prescription management in the digital age.

## Scope

The scope of this project encompasses the development of a comprehensive solution to revolutionize the interpretation and management of handwritten doctor prescriptions. The focus lies in creating a robust system capable of addressing challenges in text detection, recognition, and information extraction from handwritten prescription images. Central to this endeavor is the implementation of advanced deep learning algorithms and Natural Language Processing (NLP) techniques to extract and process critical prescription details such as medicine names, dosages, and instructions.

The project aims to deliver a user-friendly interface that ensures accessibility for both healthcare professionals and patients. By automating the interpretation process, the system will enhance prescription clarity, reduce errors, and streamline healthcare workflows. A key aspect of the project is the integration of Named Entity Recognition (NER) to accurately identify and categorize key entities, ensuring the information is presented in a structured and readable format.

Additionally, the project will prioritize scalability and adaptability, ensuring the system can handle diverse and complex handwritten prescriptions. By combining precision, efficiency, and usability, the solution aims to transform prescription management, improve patient safety, and enhance the overall efficiency of healthcare systems.

## Software and Hardware Requirements

## Software Requirements:

1. **Programming Languages:** Python is used as the primary language for developing deep learning models, image preprocessing pipelines, and NLP tasks, while JavaScript (React) builds an interactive and responsive web interface.
2. **Image Preprocessing Libraries:** OpenCV generates binarization maps and performs image enhancement (e.g., noise removal and resizing), and Pillow (PIL) handles image preprocessing efficiently.
3. **Text Detection and Recognition Frameworks:** DBNet provides robust text detection and bounding box generation, and TrOCR (Transformer-based OCR) accurately recognizes handwritten text using transformer encoder-decoder architectures.
4. **Natural Language Processing (NLP):** BERT extracts medicine names, dosages, and instructions from recognized text via Named Entity Recognition, spaCy handles additional text processing and tokenization, and fuzzy matching libraries enhance entity recognition accuracy.
5. **Deep Learning Framework:** PyTorch develops and trains models, including those based on BERT and TrOCR, leveraging GPU acceleration.
6. **Web Interface Development:** React creates a dynamic and user-friendly web interface for uploading prescription images, viewing extracted data, and interacting with the system.
7. **Post-Processing Tools:** Custom scripts perform noise reduction, text structuring, and error correction in recognized text sequences.
8. **IDE and Version Control:** Visual Studio Code is used for writing and debugging code, while Git manages version control, collaboration, and repository management.

## Hardware Requirements:

1. **Processor:** High-performance processors like Intel Core i5/i7 or AMD Ryzen 7/9 for encryption and data processing tasks.
2. **Processor:** A high-performance multi-core processor, such as Intel Core i7/i9 or AMD Ryzen 7/9, efficiently processes complex image and text data.
3. **Memory (RAM):** At least 16GB of RAM is necessary for handling preprocessing tasks, model training, and real-time inference; 32GB or more is ideal for large-scale operations.
4. **Graphics Processing Unit (GPU):** An NVIDIA GPU, such as the RTX 3060/3080 series or Tesla models, accelerates deep learning tasks like feature extraction, transformer-based modeling, and inference.
5. **Storage:** SSD storage with a minimum of 512GB ensures fast access to datasets,

intermediate outputs, and trained models.

These tools and technologies were chosen for their proven effectiveness, seamless compatibility with our project requirements, and the support provided by extensive developer communities. They lay a robust and efficient foundation for implementing medAI Nexus, our doctor prescription project, by enabling precise image preprocessing, accurate text recognition, and advanced natural language processing. With these technologies, medAI Nexus aims to revolutionize the way healthcare professionals interact with prescription data—streamlining the extraction and analysis of critical information, enhancing diagnostic accuracy, and ultimately improving patient care.

# CHAPTER 2 LITERATURE SURVEY

The MedAI Nexus is poised to transform the handling of doctor prescriptions by harnessing advanced AI technologies to streamline data extraction and improve clinical workflows. Our innovative system leverages precise image analysis, accurate text recognition, and sophisticated natural language processing to extract critical information from prescriptions efficiently. However, dealing with sensitive healthcare data demands stringent security measures to ensure the integrity and confidentiality of patient information. With increasing cyber threats and evolving regulatory requirements, traditional data protection methods may not suffice. These challenges underscore the urgent need for a robust and secure framework tailored specifically to the management and safeguarding of prescription data in modern healthcare settings

## 2.1 Survey of Smart Event Timestamping

This literature survey for MedAINexus reviewed and analyzed key advancements in Optical Character Recognition (OCR) and Natural Language Processing (NLP) for healthcare applications, particularly focusing on their potential to digitize and interpret medical prescriptions. The survey includes recent research on text recognition accuracy, adaptability to diverse conditions, and the application of advanced deep learning techniques in biomedical contexts.

**Key Advances in OCR and NLP Models**

Research by Xiang et al. (2023) emphasized the development of the DBNet-CRNN framework for pill box text recognition. This approach integrates DBNet for text detection with CRNN for text recognition, eliminating the need for extensive preprocessing and achieving improved text localization and recognition accuracy. The framework significantly reduces labor-intensive manual entry of medical text, enhancing efficiency and usability.

Zhang et al. (2024) developed the Transformer-based OCR (TrOCR) model, leveraging Transformer architectures for superior performance in handling printed and handwritten medical text. Their study systematically evaluated TrOCR’s accuracy under various image distortions and emphasized its adaptability to challenging scenarios, such as blur, noise, and font variation, with character recognition accuracy ranging from 82% to 97%.

Li et al. (2023) provided insights into the application of TrOCR in text recognition, underscoring its convolution-free architecture and its ability to outperform traditional CNN-based methods. The integration of pre-trained Vision and Text Transformers enhanced recognition accuracy for both printed and handwritten datasets, demonstrating its potential in real-world medical applications.

Lee et al. (2020) introduced BioBERT, a pre-trained language representation model specifically designed for biomedical text mining tasks, such as named entity recognition (NER) and relation extraction. BioBERT, when fine-tuned for NER tasks, has shown state-of-the-art (SOTA) performance in extracting medicine names and dosages from biomedical corpora.

Sun et al. (2021) proposed utilizing BioBERT within a Machine Reading Comprehension (MRC) framework for biomedical NER. This model achieves SOTA performance across datasets like BC5CDR-Chem, with F1 scores exceeding 90%, and demonstrates its capability to adapt to complex text conditions.

**Challenges in Current Systems**

Despite significant advances, challenges persist in OCR and NLP for healthcare:

* **Handwritten Data Complexity**: Systems such as DBNet-CRNN rely heavily on high-quality input images, which limits their performance on noisy or degraded handwritten texts.
* **Variability in Handwritten Styles**: TrOCR and BioBERT-MRC face difficulties when interpreting heavily stylized or obscure handwritten text due to lack of adequate training data.
* **Domain-Specific Requirements**: Many existing models are not optimized for extracting specific entities like medicine names and dosages, necessitating further domain-specific adaptations.

**Insights and Future Directions**

The surveyed studies underscore the transformative potential of integrating OCR and domain-specific NLP models for healthcare applications. To advance the MedAINexus system, focus areas include:

* Improving robustness against low-quality and handwritten inputs.
* Utilizing pre-trained biomedical models such as BioBERT for better entity recognition.
* Developing scalable solutions that integrate OCR with advanced NER frameworks to accurately extract medicine names, dosages, and instructions.

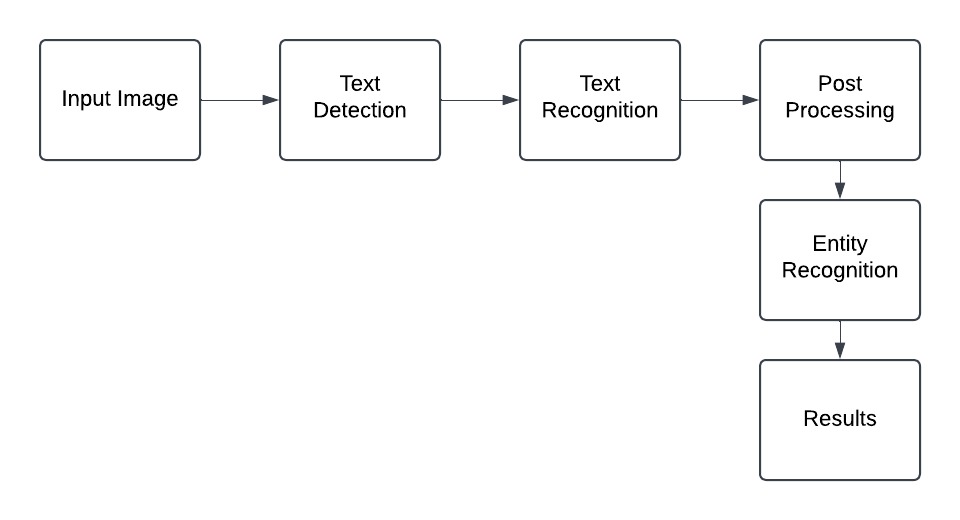
*Table 2.1 Survey of the project*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Yr. of Pub** | **Author** | **Technique** | **Summary** | **Limitation** |
| 1 | 2023 | Liuqing Xiang, Hanyun Wen, Ming Zhao | DBNet-CRNN | Developed a framework for pill box text recognition combining DBNet and CRNN for enhanced accuracy. | High dependency on clear and consistent image quality. |
| 2 | 2024 | Ray L. Zhang | TrOCR | Evaluated TrOCR’s robustness against image effects like blur and noise, achieving superior accuracy. | Accuracy decreases with historical or heavily stylized handwritten texts. |
| 3 | 2023 | Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, Furu Wei | Transformer-based OCR (TrOCR) | Introduced pre-trained Vision and Text Transformers for text recognition, surpassing CNN-based methods. | Performance  can decline  with noisy or  occluded  inputs. |
| 4 | 2020 | Jinhyuk Lee,  Wonjin Yoon,  Sungdong Kim,  Donghyeon Kim,  Sunkyu Kim,  Chan Ho So,  Jaewoo Kang | BioBERT | Adapted pre-trained BERT for biomedical text mining, excelling in NER tasks. | Requires  domain-specific fine-tuning for handwritten text and prescription datasets. |
| 5 | 2021 | Cong Sun,  Zhihao Yang,  Lei Wang, Yin  Zhang, Hongfei  Lin, Jian Wang. | BioBERT  MRC  Framework | Applied BioBERT in an MRC framework for entity recognition with superior accuracy. | Struggles with  highly noisy or  handwritten  data due to  limatations in  the training  corpus. |

# CHAPTER 3 SYSTEM DESIGN

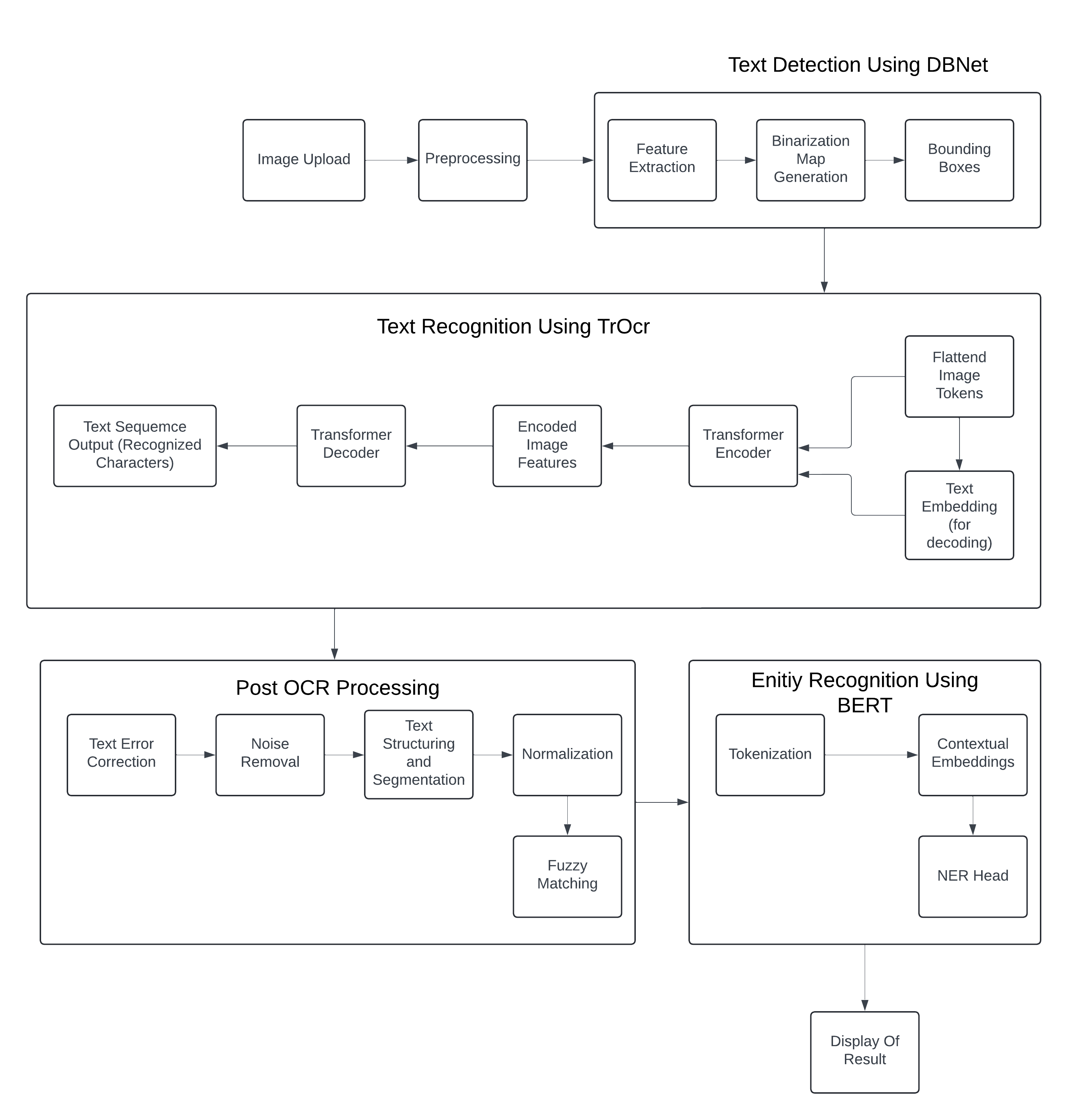
* 1. **Flow Chart**

The flowchart will serve as an intuitive visual representation of the innovative processes in our project, **MedAINexus: Advanced Handwritten Prescription Processing System**. This flowchart meticulously outlines the sophisticated workflow, starting from the input of a doctor's handwritten prescription image to the final extraction and display of crucial medical information. Each phase, encompassing advanced text detection, precise text recognition, intelligent post-processing, context-aware entity recognition, and comprehensive results presentation, is illustrated using modern symbols and dynamic arrows to clearly represent the logical flow and critical decision points. These visual tools enhance understanding of the system’s intricate operations, identify potential areas for optimization, and facilitate seamless communication among stakeholders, ensuring a unified and efficient development process. By providing a clear and detailed depiction of each process stage, the flowchart underscores MedAINexus's commitment to leveraging cutting-edge AI technologies for improving healthcare outcomes through accurate and efficient prescription interpretation.

 *Fig 3.1 Data flow*

## System Architecture

The system architecture is meticulously crafted to enable precise and efficient processing of handwritten prescriptions. It combines advanced machine learning and deep learning techniques to transform unstructured handwritten text into meaningful, structured information. The process is designed to provide a seamless experience for users by moving from image input to structured result display



*Fig 3.2 System Architecture*

## Preprocessing and Feature Extraction

## Purpose: This stage prepares uploaded images for text recognition by enhancing their quality and

## extracting meaningful visual features.

## Image Upload: Users upload a prescription image through the web interface.

## Preprocessing: The system applies techniques like binarization, noise reduction, and

## resizing to improve image clarity for subsequent tasks.

## Feature Extraction and Binarization Map Generation:

## Feature Extraction: Captures spatial information using convolutional layers.

## Binarization Map: Generates a map to highlight text regions, aiding in accurate bounding box detection.

## Bounding Boxes: Text regions are localized for further processing.

## Transformer-based Recognition and Encoding

**Purpose**: Converts localized text into structured, machine-readable formats using a transformer-based approach.

**Flattened Image Tokens and Text Embedding Preparation:**

* Detected bounding boxes are converted into image tokens, and embeddings are

initialized to guide the decoding process.

**Transformer Encoder:** Encodes visual features into contextualized representations.

**Encoded Image Features:** Output features retain both spatial and semantic details.

**Transformer Decoder:** Decodes the encoded image features into text sequences (recognized characters).

**3. Post-Processing for Text Structuring:**

**Purpose:** Refines the recognized text to improve readability and ensures accurate entity extraction.

**Text Error Correction:** Fixes errors introduced during OCR recognition.

**Noise Removal:** Eliminates unnecessary artifacts or misrecognized symbols.

**Text Structuring and Segmentation:** Groups recognized text into structured fields such as medicine names, dosages, and instructions.

**Normalization and Fuzzy Matching:** Ensures consistency in entity formats and boosts recognition accuracy using approximate string matching techniques.

**Tokenization and NER (Named Entity Recognition):**

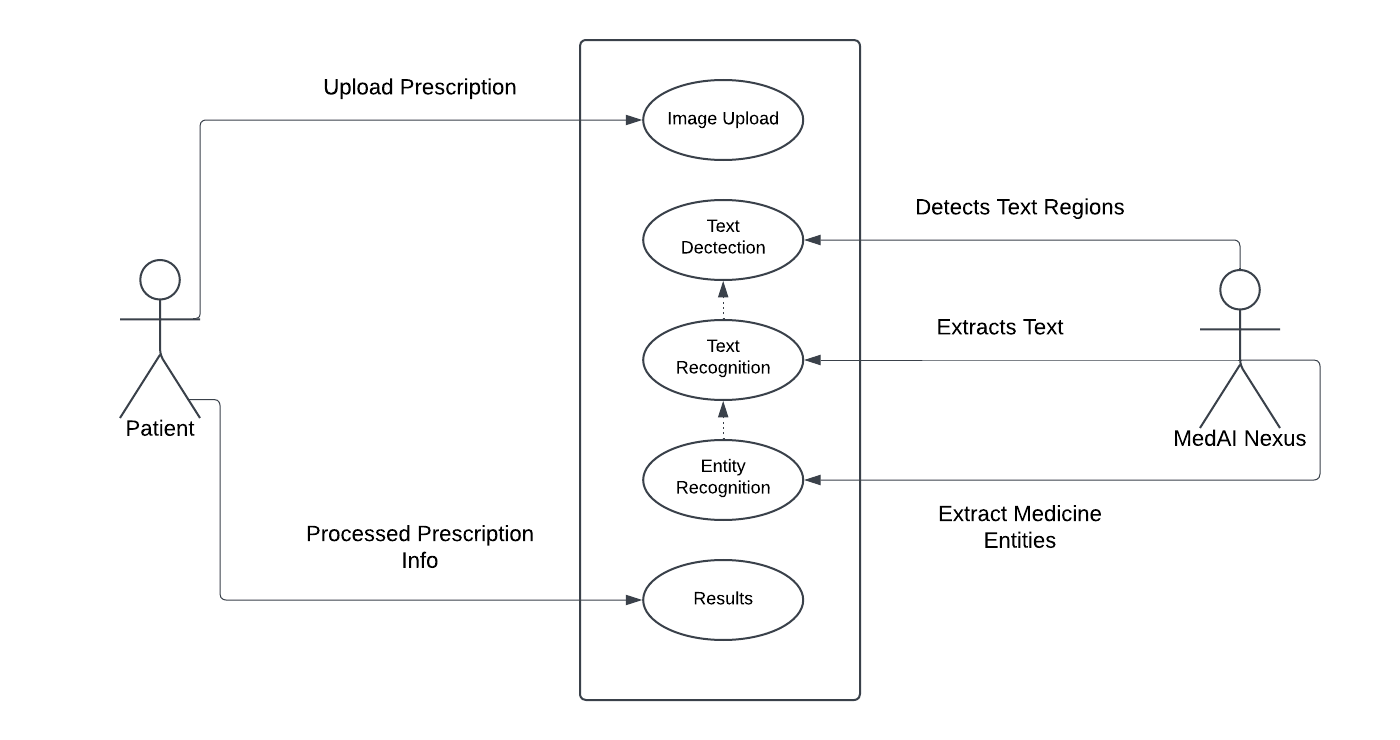
* Uses BERT to tokenize text and extract key entities (e.g., drug names, dosages).
* Generates contextual embeddings that aid in identifying the relationships

between entities.

## UML Diagrams

**Use case diagram:**

The use case diagram for the **Doctor Handwritten Prescription Project** visually illustrates user interactions and system functionalities. It showcases actors such as **Patients** uploading handwritten prescriptions and **MedAI Nexus** performing core backend operations. Essential use cases include **image upload**, **text detection**, **text recognition**, **entity recognition**, and the delivery of **processed prescription information**. This ensures a seamless and efficient workflow from prescription input to structured medical data output.



*Fig 3.3 Use Case Diagram*

## State diagram:

## The state diagram for the Doctor Handwritten Prescription Project illustrates the various states and transitions within the system. It encapsulates the system's behavior and the conditions under which it transitions between different stages of prescription processing. By visually representing the system's dynamic workflow, the state diagram provides insights into how the system responds to user inputs, preprocesses images, detects text regions, recognizes handwritten text, extracts relevant medical entities, and displays results to the user.

## Through a series of defined states and transitions, the diagram highlights error-handling mechanisms, such as managing upload failures, text detection errors, or entity recognition issues. This comprehensive representation offers an in-depth overview of the system’s operational flow, enabling a clear understanding of its functionality and behaviour in transforming handwritten prescriptions into structured, actionable data.

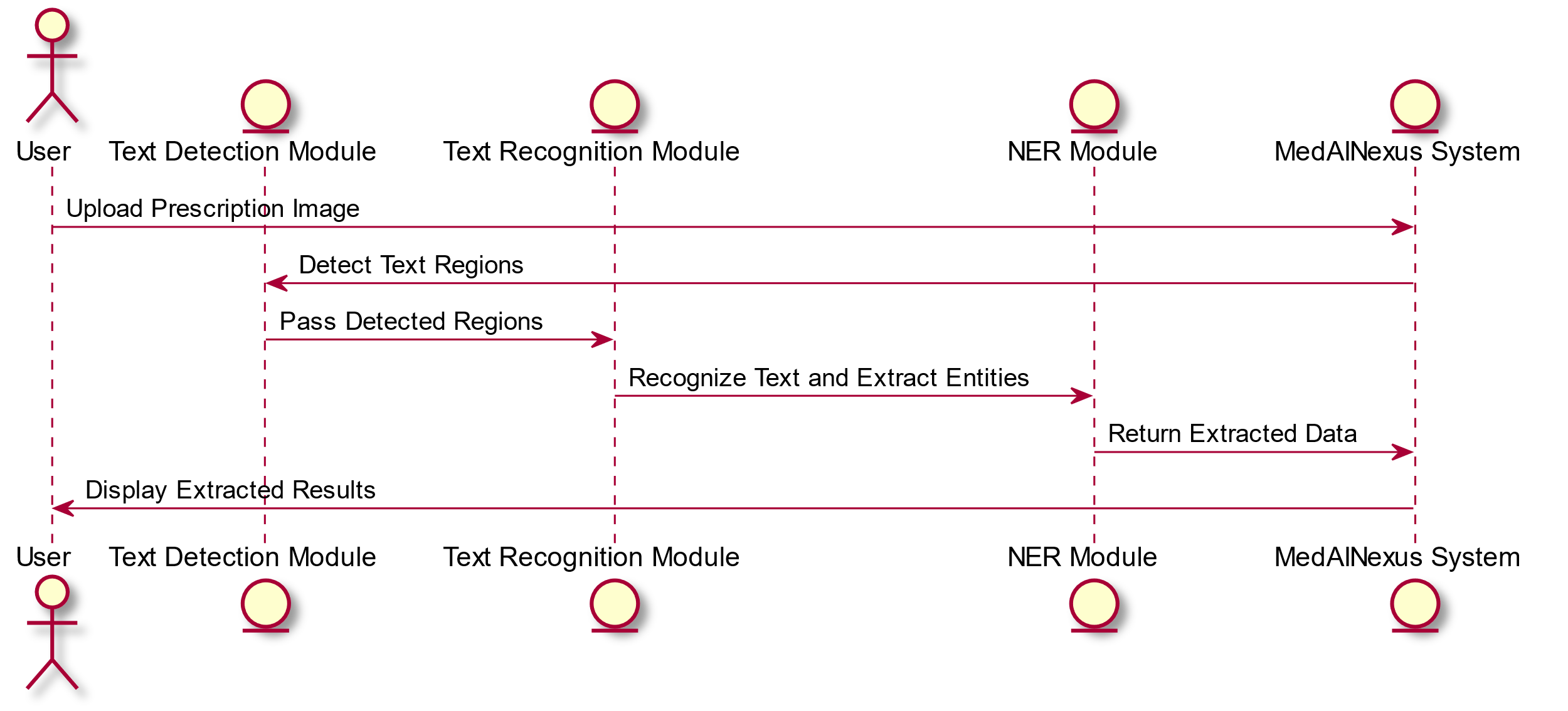
## 

*Fig 3.4 State Diagram*

## Sequence diagram:

A sequence diagram for the **Doctor Handwritten Prescription Project** visually represents the interaction between the system's components and actors. It starts with the **Patient** uploading a handwritten prescription. The system processes the input through the following steps:

the prescription image is received and analyzed for **text detection**, the detected text is passed to the **text recognition** module, and then extracted entities (e.g., medicine names) are processed through the **entity recognition** module. Finally, the **processed prescription information** is delivered back to the patient. The diagram includes lifelines for each component, depicting method calls, message exchanges, and data flows, providing a clear visualization of the system's operational workflow.



*Fig 3.5 Sequence Diagram*

* 1. **Project Plan**

Our 8-week project aims to develop the **MedAI Nexus**, an innovative solution designed to digitize and process handwritten medical prescriptions. The project will involve configuring and integrating state-of-the-art technologies like DBNet, TrOCR, and BERT for text detection, recognition, and entity extraction. We will organize and preprocess datasets, train the model to achieve high accuracy, and design a user-friendly interface. The system will ensure efficient recognition and extraction of key medical information, enabling seamless prescription digitization and improved medical data accessibility.

## Week 1: Project Initiation and Environment Setup

**Objective:** Establish the project foundation and prepare the development environment.

## Tasks:

* + - Conduct a project kickoff meeting to align the team on objectives, roles, and responsibilities.
    - Create a new Conda environment and install necessary dependencies from the `requirements.txt`.
    - Configure the development environment for GPU usage to ensure optimal performance.
    - Set up version control with Git and establish a repository for the project.

**Week 2: Dataset Collection and Preprocessing**

**Objective:** Gather and prepare data for training and testing.

## Tasks:

* + - Collect handwritten prescription images from diverse sources for robust model training.
    - Annotate the dataset with bounding boxes for text regions and labels for text content.
    - Perform data augmentation techniques (e.g., rotation, noise addition) to increase dataset diversity.
    - Split the dataset into training, validation, and test sets.

## Week 3: Text Detection Module Development

**Objective:** Implement and train the DBNet model for detecting handwritten text regions.

## Tasks:

* + - Configure and train the DBNet model on the preprocessed dataset.
    - Evaluate model performance and fine-tune hyperparameters to improve detection accuracy.
    - Test the text detection module on sample prescription images.

## Week 4: Text Recognition Module Development

**Objective:** Implement the TrOCR model to recognize detected handwritten text.

## Tasks:

* + - Train the TrOCR model using the detected text regions.
    - Validate the recognition accuracy using the validation set.
    - Optimize the model for faster recognition without compromising accuracy.

## Week 5: Named Entity Recognition (NER) Module Integration

**Objective:** Extract key medical entities such as medicine names, dosages, and instructions.

## Tasks:

* + - Fine-tune the BERT model for medical NER using annotated text data.
    - Integrate the NER module with the text recognition output.
    - Test the end-to-end pipeline for prescription image processing

## Week 6: System Integration and Backend Development

**Objective:** Develop Combine all modules into a cohesive system and build backend functionality.

## Tasks:

* + - Integrate the text detection, recognition, and NER modules.
    - Develop backend APIs to process prescription images and return structured data.
    - Ensure smooth communication between modules and resolve integration issues.

## Week 7: Frontend Development and User Interface Design

**Objective:** Create a user-friendly interface for prescription processing.

## Tasks:

* + - Design and implement a web-based interface for users to upload prescription images.
    - Display extracted prescription information in an organized and readable format.
    - Test the frontend for usability and responsiveness.

## Week 8: Testing, Deployment, and Project Wrap-Up

**Objective:** Finalize the project and deliver a fully functional system.

## Tasks:

* + - Perform end-to-end testing to ensure system reliability and accuracy.
    - Deploy the system on a cloud platform or local server for user access.
    - Conduct a final project review and prepare documentation for system usage.

**CHAPTER 4**

**SYSTEM IMPLEMENTATION & METHODOLOGIES**

**4.1 System Implementation**

With the growing need for accurate and efficient processing of handwritten medical prescriptions, **the Doctor Prescription Recognition System** has become an essential tool in modern healthcare. Manual transcription of prescriptions is not only labor-intensive but also susceptible to errors, which can adversely affect patient safety and clinical outcomes. Automating the conversion of handwritten prescriptions into structured digital records significantly enhances data accessibility, reduces manual workload, and improves overall treatment accuracy.

Our system is designed to bridge this gap by enabling healthcare providers to seamlessly transform handwritten prescriptions into digital, machine-readable text. By leveraging advanced deep learning-based sequence-to-sequence models, the system accurately extracts and interprets complex handwriting. Unlike conventional OCR solutions that often struggle with inconsistent handwriting styles, our approach integrates specialized modules for text detection, recognition, and semantic analysis to ensure high precision and contextual understanding.

In implementing this system, our focus is on developing a unified framework that effectively handles a diverse range of prescription formats, enhances recognition efficiency, and seamlessly integrates with electronic health record (EHR) systems. The following components form the foundation of our implementation:

**i. Prescription Image Preprocessing**

1. **Data Acquisition**
   * Collect prescription images from diverse sources (scanned documents, smartphone photos) to capture various handwriting styles and layouts.
   * Organize images into training, validation, and testing sets to ensure comprehensive model evaluation.
2. **Image Enhancement**
   * **Normalization:** Adjust brightness, contrast, and orientation to standardize image appearance.
   * **Noise Reduction:** Apply filters to remove background artifacts, enhancing text clarity.
   * **Segmentation (if required):** Isolate relevant sections (e.g., drug names, instructions) when multiple prescriptions or notes appear on a single page.

These steps ensure consistent, high-quality input for subsequent text detection and recognition stages.

**ii. Text Detection & Bounding Box Generation**

1. **DBNet for Text Detection**
   * **Feature Extraction**: DBNet processes the preprocessed image to generate feature maps.
   * **Binarization Map Generation:** The model produces a probability map to distinguish text pixels from the background.
   * **Bounding Boxes:** Using the binarization map, DBNet identifies and refines bounding boxes around each text instance, accurately localizing handwriting regions.
2. **Adaptability**
   * The text detection pipeline is robust to varied handwriting styles and prescription layouts.
   * DBNet’s dynamic thresholding helps it adapt to inconsistent pen strokes or image artifacts.

By isolating text regions early, the system reduces noise for the recognition model and focuses computation on relevant areas.

**iii. Handwritten Text Recognition**

1. **Transformer-Based Architecture (TrOCR)**
   * **Cropped Text Images:** The bounding boxes from DBNet are used to crop text regions, feeding them into the recognition model.
   * **Sequence-to-Sequence Learning:** TrOCR’s encoder-decoder structure captures spatial features (encoder) and generates character or word sequences (decoder).
   * **Contextual Understanding:** The transformer architecture helps handle variations in handwriting style, character spacing, and skew.
2. **Output**
   * **Digital Text Transcription:** The recognized text is output in a machine-readable format, preserving the prescription’s essential information.
   * **Confidence Scores:** Each recognized token is typically accompanied by a confidence score to indicate certainty, enabling potential post-processing or human review for low-confidence segments.

**iv. Medical Entity Extraction**

1. **Text Normalization & Tokenization**
   * **Cleanup:** Minor spelling errors or extraneous characters are removed.
   * **Token Splitting:** The recognized text is segmented into tokens (e.g., words or subwords) for entity recognition.
2. **BERT-Based NER**
   * **Contextual Embeddings:** A fine-tuned BERT model processes the tokenized text, capturing context for each token.
   * **Entity Classification:** A specialized NER head identifies medical entities such as drug names, dosages, frequencies, and administration instructions.
   * **Standardization:** Identified entities can be mapped to known medical terminologies (e.g., RxNorm) to ensure consistent naming conventions.

This semantic layer ensures the recognized text is clinically meaningful and structured, ready for downstream applications like pharmacy management or electronic health record (EHR) systems.

**v. Model Pretraining & Fine-Tuning**

1. **Initial Pretraining**
   * **Text Detection & Recognition Models:** DBNet and TrOCR may be pretrained on large-scale OCR datasets to build a foundational understanding of text structure.
   * **Language Model (BERT):** BERT is pretrained on extensive textual corpora for broad linguistic comprehension.
2. **Domain-Specific Fine-Tuning**
   * **Prescription Dataset:** Each model is further trained on handwritten prescription data, ensuring it adapts to medical terms and handwriting nuances.
   * **Continuous Updates:** As new prescriptions and medical abbreviations emerge, the models can be incrementally retrained or fine-tuned to maintain accuracy.
3. **Zero-Shot or Few-Shot Learning**
   * **Handling New Entities:** The system can leverage small amounts of new data to learn unfamiliar drug names or abbreviations with minimal overhead.
   * **Layout Variations:** By continuously refining the model, the system remains robust against different prescription formats and styles.
   1. **Dataset Utilization:**

In our Doctor Prescription Recognition System, we integrate a variety of datasets to train, fine-tune, and evaluate our text detection, recognition, and semantic analysis modules. This robust pipeline ensures that handwritten prescription content is accurately extracted and that detailed medicine information is reliably retrieved.

**i. Combined CORD + Handwritten Kaggle Dataset for Text Detection and Recognition**

**Overview:**  
This unified dataset harnesses the strengths of both the CORD dataset and a community-sourced Handwritten Kaggle dataset, offering a comprehensive resource for detecting and recognizing text in handwritten prescriptions.

**Contents:**

* **Images and Annotations:** Thousands of prescription images with detailed bounding box annotations, capturing diverse text orientations, complex backgrounds, and varying lighting conditions.
* **Handwritten Samples:** A wide range of handwritten text samples with corresponding ground-truth transcriptions, reflecting real-world variability.

**Characteristics:**

* **Diverse Handwriting Styles:** Covers various handwriting styles and noise conditions essential for robust performance.
* **Robust Annotations:** Provides detailed information for both text localization (detection) and conversion into machine-readable text (recognition).

**Utilization:**

* **Text Detection:** Trains our DBNet-based text detection module to accurately identify and localize handwritten text regions.
* **Text Recognition:** Serves as the primary resource for our text recognition pipeline, enhancing transcription accuracy even with complex handwriting.

**ii. BC5DAR Dataset for BERT-based Semantic Analysis**

**Overview:**  
The BC5DAR dataset is a domain-specific resource designed for biomedical semantic analysis. It contains detailed annotations of clinical entities, making it ideal for fine-tuning our BERT model.

**Contents:**

* **Annotated Clinical Data:** Includes prescription texts with labels for key medical entities such as drug names, dosages, frequencies, and administration instructions.
* **Diverse Medical Terminology:** Captures the complexity and variability of clinical language encountered in real-world prescriptions.

**Characteristics:**

* **Fine-Grained Annotations:** Provides precise labels necessary for effective medical Named Entity Recognition (NER).
* **Context-Rich Data:** Ensures the BERT model learns nuanced semantic relationships critical for clinical information extraction.

**Utilization:**

* **Semantic Analysis:** Enhances our BERT-based module’s ability to extract and standardize critical clinical information from transcribed prescriptions.
* **Clinical Information Extraction:** Improves the structuring and accuracy of the digital prescriptions by accurately identifying key medical details.

**iii. RxNorm Dataset for Medicine Information Retrieval**

**Overview:**  
Developed by the U.S. National Library of Medicine, the RxNorm dataset standardizes medicine names and provides detailed drug information, making it essential for validating and enriching recognized medicine names.

**Contents:**

* **Drug Information:** Includes medicine names, active ingredients, dosage forms, and standardized terminologies.
* **Interoperability Data:** Links across various drug vocabularies, ensuring consistent and reliable references.

**Characteristics:**

* **Authoritative Source:** Widely recognized and trusted in healthcare applications for its comprehensive drug information.
* **Clinical Standardization:** Ensures that the medicine data extracted from prescriptions conforms to established clinical norms.

**Utilization:**

* **Medicine Verification:** After text recognition and semantic analysis, the RxNorm dataset is used to verify and enrich extracted medicine names by providing detailed drug profiles.
* **Clinical Decision Support:** Facilitates processes like drug interaction checks and patient record management by ensuring digital prescriptions are clinically accurate.

**Integration and Impact**

By integrating these datasets in the specified order, our system achieves:

* **Accurate Localization and Transcription:** The combined CORD + Handwritten Kaggle dataset ensures robust text detection and recognition.
* **Enhanced Semantic Understanding:** The BC5DAR dataset empowers our BERT-based semantic analysis module to accurately extract and structure key clinical details.
* **Reliable Medicine Verification:** The RxNorm dataset enriches the recognized data with authoritative drug information, supporting clinical decision-making.
* **Overall Robustness:** This integrated approach prepares the system to handle varied prescription formats and handwriting styles, delivering a reliable digital prescription management solution.

**4.3 Handwritten Prescription Recognition and Data Structuring**

We categorize the handwritten prescription recognition process into three core components: image preprocessing and text detection, text recognition and semantic analysis, and data structuring and integration, each addressing different aspects of the problem.

**i. Image Preprocessing and Text Detection**

This stage focuses on preparing the prescription image and accurately localizing handwritten text.

The system performs:

* **Image Normalization:** Adjusting brightness, contrast, and orientation to standardize input images.
* **Noise Reduction:** Applying filtering techniques to remove artifacts and background interference, enhancing text clarity.
* **Text Region Detection:** Utilizing a DBNet-based module and a combined Kaggle dataset to identify and generate bounding boxes around handwritten text areas.

This process ensures that the subsequent recognition stages operate on clean, well-defined regions, minimizing ambiguity and maximizing detection accuracy.

**ii. Handwritten Text Recognition and Semantic Analysis**

Once text regions are isolated, the system converts them into digital text and extracts clinically relevant information. This process involves:

* **Text Recognition:** Leveraging a transformer-based model (TrOCR) trained on both the CORD dataset and Kaggle data for handwritten text recognition to transcribe text accurately.
* **Contextual Refinement:** Applying post-recognition processing to correct minor errors and standardize the transcribed text.
* **Semantic Extraction:** Employing a BERT-based Named Entity Recognition (NER) module to identify key medical entities such as drug names, dosages, frequencies, and administration instructions.

Together, these steps transform raw handwriting into accurate, contextually meaningful data, which is crucial for clinical interpretation and decision-making.

**iii. Medicine Information Extraction Using RxNorm**

Once the handwritten prescription has been transcribed and analyzed semantically, the extracted drug information is refined and enriched through the RxNorm dataset to produce a structured digital record. This process involves:

* **Data Structuring:** Organizing recognized medicine details into predefined schemas that align with clinical and electronic health record (EHR) standards.
* **Validation and Error Correction:** Leveraging the authoritative RxNorm dataset to verify, standardize, and augment the extracted drug information, ensuring accuracy and consistency in medicine names, dosages, and formulations.
* **System Integration:** Seamlessly interfacing with EHR and pharmacy management systems to enable real-time updates, retrieval, and clinical decision support based on enriched, structured prescription data.

By converting unstructured handwritten prescriptions into structured digital records, this stage enhances data accuracy, streamlines workflows, and ultimately improves patient safety.

* 1. **Pretraining for Handwritten Prescription Recognition**

The availability of large-scale datasets for handwriting detection and recognition has significantly advanced our ability to process handwritten prescriptions. Much like pretraining in vision-language models enhances multimodal understanding, large-scale pretraining in our system improves the accuracy and efficiency of detecting, transcribing, and semantically interpreting handwritten prescriptions.

**i. Large-Scale Pretraining with Handwriting and OCR Datasets**

Our approach begins by leveraging robust pretrained models such as DBNet and TrOCR. These models are initially trained on extensive datasets, including the combined CORD + Handwritten Kaggle dataset. This comprehensive dataset offers a rich collection of prescription images and handwritten samples, complete with detailed annotations. Pretraining on these diverse samples enables our system to:

* Learn to robustly localize handwritten text in varied real-world conditions.
* Develop accurate transcription capabilities across a wide spectrum of handwriting styles.
* Establish foundational visual representations that support subsequent semantic extraction.

In addition, we incorporate pretrained language models like BERT to build contextual embeddings that serve as a solid basis for our downstream semantic tasks.

**ii. Domain-Specific Fine-Tuning**

Handwritten prescriptions present unique challenges that require targeted fine-tuning:

* **Customized Training on Prescription Images:** We refine our detection and recognition models using annotated prescription datasets to capture the nuances of medical shorthand, specialized formatting, and domain-specific characteristics.
* **Medical Language Adaptation:** Our BERT model is further fine-tuned on the BC5DAR dataset, which is specifically curated with detailed clinical annotations. This step ensures that the semantic analysis module can accurately identify and extract critical information such as drug names, dosages, and administration instructions.
* **Augmentation and Self-Supervision:** Data augmentation and self-supervised learning techniques are employed to further enhance model robustness, enabling our system to generalize effectively across varied handwriting styles and environmental conditions.

**iii. Eliminating Dependency on Rule-Based Methods**

Traditional rule-based methods struggle with the variability and complexity of handwritten prescriptions. Our end-to-end deep learning pipeline eliminates the need for handcrafted rules by:

* **Direct End-to-End Processing:** Converting prescription images directly into structured digital records without relying on predefined templates.
* **Zero-Shot Adaptability:** Equipping the system to handle previously unseen handwriting styles and prescription formats, ensuring consistent performance in diverse real-world scenarios.
* **Unified Framework Integration:** Seamlessly combining text detection, recognition, and semantic analysis into a single processing pipeline, which minimizes the reliance on error-prone heuristic methods.

By leveraging large-scale pretraining, targeted domain-specific fine-tuning, and an integrated end-to-end architecture, our Doctor Prescription Recognition System achieves high accuracy and resilience. This approach not only streamlines the conversion of handwritten prescriptions into actionable digital records but also significantly enhances clinical workflow efficiency and patient safety.

**4.5 Unified Formulation**

Our Doctor Prescription Recognition System employs a unified approach to convert raw prescription images into structured, actionable clinical data. This formulation ensures consistent processing across diverse handwriting styles and document layouts.

**Towards Unified Prescription Representation**

Given an input handwritten prescription image (I) and a target medical data schema (M), our system processes I to generate a comprehensive digital record. We define three core elements for the unified formulation:

• **Image Feature Extraction (If):** Utilizes pretrained models (e.g., DBNet) to extract robust visual features from the prescription image. This step captures variations in handwriting style, layout, and image quality, establishing the foundation for accurate text localization.

• **Digital Transcription (Dt):** Converts the detected text regions into machine-readable text using transformer-based models like TrOCR. This process ensures that handwritten content is accurately transcribed into digital text, preserving essential details such as dosage instructions and drug names.

• **Medical Entity Mapping (Me):** Processes the transcribed text through a BERT-based module to identify and standardize key medical entities. This includes mapping entities such as drug names, dosages, and administration frequencies to a structured medical schema, ensuring clinical relevance and consistency.

Together, these components form an integrated pipeline that seamlessly transforms handwritten prescriptions into structured digital records, enhancing data accuracy and streamlining integration into electronic health systems.

**4.5.1 Revisiting Various Prescription Recognition Tasks**

Our unified framework addresses the challenges inherent in processing handwritten prescriptions, ensuring accurate extraction and structuring of vital medical information.

**i. Scalable Document Understanding:** To enhance the system’s adaptability, we employ advanced pretraining and domain adaptation techniques that enable the system to:

* **Dynamically Recognize Diverse Formats:** Adapt to variations in layout, handwriting style, and image quality across different prescription types.
* **Map Handwritten Content to Medical Data Fields:** Automatically align text segments with corresponding fields such as drug names, dosages, and administration instructions.
* **Generalize Across Data Sources:** Leverage large-scale pretraining to maintain robust performance even with previously unseen prescription styles.

**ii. Inferring Missing or Ambiguous Information:** Handwritten prescriptions can sometimes be incomplete or unclear. Our system addresses these challenges by:

* **Utilizing Contextual Cues:** Drawing on neighboring text to infer missing details like dosage frequency or measurement units.
* **Applying Medical Constraints:** Incorporating standard medical guidelines and common abbreviations to resolve uncertainties.
* **Leveraging Historical Patterns:** Using insights from previously processed prescriptions to predict and correct unclear elements, ensuring a complete and coherent transcription.

**4.5.2 Prescription Categorization and Labelling**

To optimize the recognition process, our system categorizes prescriptions based on their complexity. This tailored approach allows for efficient processing and improved accuracy.

**i. Simple Prescriptions (Single Medication Entry)**

**Example:** “Take 1 tablet of Aspirin 100mg daily.”

* **Characteristics:**
  + Clear, legible handwriting with a straightforward layout.
  + Minimal annotations or extraneous notes.
* **Processing:**
  + Direct detection and transcription with basic entity extraction, resulting in quick and accurate digitization.

**ii. Complex Prescriptions (Multiple Medications with Detailed Instructions)**  
**Example:** “Take 2 tablets of Paracetamol 500mg every 6 hours, and 1 tablet of Ibuprofen 200mg as needed.”

* **Characteristics:**
  + Multiple entries with varied formatting and additional notes.
  + May include layered instructions and dosage adjustments.
* **Processing:**
  + Requires advanced segmentation and contextual analysis to extract and organize multiple data points accurately.

**iii. Advanced Prescriptions (Ambiguous or Corrected Handwriting)**  
**Example:** Prescriptions containing overlapping text, corrections, or ambiguous annotations.

* **Characteristics:**
  + Complex layouts with potential ambiguities or inconsistent handwriting.
  + Often feature handwritten corrections or smudged text.
* **Processing:**
  + Involves sophisticated post-processing, including error correction and context-based inference, to resolve ambiguities and produce a reliable digital record.

By categorizing prescriptions into simple, complex, and advanced types, our system ensures that each document is processed with the appropriate level of analysis, ultimately delivering structured, actionable medical data that seamlessly integrates into clinical workflows.

* 1. **Unified Model**

We introduce our unified Doctor Prescription Recognition model, which integrates image preprocessing, text detection, text recognition, and semantic analysis into one cohesive framework to accurately transcribe and extract clinical data from handwritten prescriptions.

**4.6.1 Overview**

As depicted in Fig. 4.3, our model comprises three main components:

* **Text Detection Module:** Utilizes a DBNet-based approach trained on a combined dataset that merges the CORD dataset with a Kaggle-sourced handwritten text dataset. This composite dataset provides a rich variety of prescription images and authentic handwritten samples, ensuring precise localization of handwritten regions.
* **Text Recognition Module:** Employs a transformer-based sequence-to-sequence architecture (akin to TrOCR) that has been pre-trained on the integrated Kaggle and CORD datasets. This training strategy enables the model to accurately convert handwritten text from prescription images into digital transcriptions.
* **Semantic Analysis Module:** Leverages a BERT-based framework that is fine-tuned on the BC5DAR dataset—a domain-specific resource enriched with clinical annotations. This fine-tuning allows the module to extract and standardize critical medical entities such as drug names, dosages, and administration frequencies.

Given an input prescription image (I), the model processes it through the following stages:

**i. Image Encoding & Text Detection**

* **Preprocessing:** Standardizes images through normalization, contrast enhancement, and noise reduction to prepare them for analysis.
* **Feature Extraction:** Applies DBNet to detect handwritten text and generate precise bounding boxes around regions of interest.
* **Region Proposal:** Extracts candidate areas that likely contain meaningful handwritten content for further processing.

**ii. Cross-Modal Interaction & Text Recognition**

* **Text Recognition:** Processes the cropped text regions using our transformer-based model, converting handwriting into machine-readable text while preserving the contextual integrity of the content.
* **Error Correction:** Implements post-recognition refinement to address ambiguous or unclear handwriting elements, thereby improving transcription accuracy.

**iii. Semantic Extraction & Data Structuring**

* **Tokenization & Embedding:** Converts recognized text into tokens and produces rich contextual embeddings via our fine-tuned BERT module.
* **Medical Entity Extraction:** Identifies and extracts essential clinical details such as medication names, dosage instructions, and scheduling information.
* **Structured Output:** Organizes the extracted information into a standardized schema that seamlessly integrates with electronic health record systems.

This integrated approach, underpinned by large-scale pretraining on diverse datasets and targeted domain-specific fine-tuning, ensures that our Doctor Prescription Recognition System reliably converts handwritten prescriptions into structured digital records. The result is a robust, efficient solution that significantly enhances clinical workflow efficiency and patient safety.

**4.6.2 Pretraining Objectives**

To achieve robust performance and generalization, our model is pre-trained with targeted objectives across its core components:

**i. Text Detection Loss**

* **Objective:**  
  Minimize the discrepancy between predicted and ground-truth bounding boxes (using metrics like Intersection-over-Union) to ensure precise localization of handwritten text.
* **Training:**  
  The text detection module is trained on a combined dataset that merges the CORD dataset with a dedicated handwritten text dataset from Kaggle. This blend exposes the model to a wide range of handwriting styles and image conditions, ensuring robust detection in real-world prescription images.

**ii. Text Recognition Loss**

* **Objective:**  
  Utilize cross-entropy loss at the character level to achieve accurate transcription of handwritten content into digital text.
* **Training:**  
  The text recognition module is pre-trained on the same combined CORD and Kaggle handwritten text dataset. This diverse training set enables the model to handle variations in handwriting, noise levels, and other real-world challenges effectively.

**iii. Semantic Extraction Loss**

* **Objective:**  
  Optimize the identification of key medical entities through specialized loss functions tailored for named entity recognition.
* **Training:**  
  The BERT-based semantic analysis module is fine-tuned on the BC5DAR dataset—a domain-specific resource enriched with detailed clinical annotations. This fine-tuning penalizes incorrect or missed entity extractions, enhancing the clinical relevance and accuracy of the output.

**iv. Optimization and Inference Acceleration**

* **Runtime Optimization:** Implement techniques such as model quantization and leverage optimized inference frameworks (e.g., ONNX Runtime, TensorRT) to reduce processing time.
* **Adaptive Refinement:** Optionally incorporate reinforcement learning strategies to further refine transcription and extraction accuracy based on real-world feedback.

By leveraging these pretraining objectives, our Doctor Prescription Recognition System is equipped to robustly detect and transcribe handwritten text from prescriptions, while accurately extracting critical clinical information.

**4.7 Inference**

During inference, our Doctor Prescription Recognition System processes an input prescription image to produce structured, actionable clinical data. The system follows a systematic pipeline to convert handwritten content into digital text, verify its accuracy, and format the output for clinical applications.

**Step 1: Image Acquisition & Preprocessing**

* **Image Input:** The system accepts scanned or photographed prescription images.
* **Preprocessing:**  
  Standardizes images by applying normalization, contrast enhancement, and noise reduction.
* **Region Preparation:** Segments the image to identify areas most likely to contain handwritten text.

**Step 2: Text Detection & Recognition**

* **Text Detection:** Utilizes a DBNet-based module, trained on the combined CORD and Kaggle handwritten datasets, to locate text regions and generate bounding boxes.
* **Region Extraction:** Crops the detected text regions from the preprocessed image to isolate areas for analysis.
* **Text Recognition:** Processes the cropped regions using a transformer-based recognition model (similar to TrOCR) to convert handwritten text into digital form.

**Step 3: Semantic Extraction & Validation**

* **Semantic Analysis:** The transcribed text is processed by a BERT-based module, fine-tuned on medical prescription data from the BC5DAR dataset, to extract key clinical entities such as drug names, dosages, and administration schedules.
* **Validation Checks:** Contextual validation is performed to ensure that the extracted information is clinically plausible (for example, verifying dosage units and common medication names).
* **Error Correction:** Post-processing techniques are applied to resolve ambiguities or correct transcription errors.

**Step 4: Structured Data Output & Integration**

* **Data Structuring:** The extracted clinical information is organized into a standardized format that conforms to clinical data schemas.
* **Integration:** The structured prescription data is prepared for seamless integration into electronic health record (EHR) systems.
* **Feedback Loop:** When confidence scores are low or inconsistencies are detected, the system flags the output for human review, ensuring high reliability in critical applications.

**Task-Specific Inference Strategies**

* **Simple Prescriptions:** For prescriptions with a single, clearly written medication entry, the system provides quick and direct transcription with minimal need for contextual disambiguation.
* **Complex Prescriptions:** For prescriptions containing multiple medications with detailed instructions, advanced segmentation and semantic analysis are employed to accurately separate and label each data point.
* **Ambiguous or Corrected Handwriting:** In cases of overlapping text or ambiguous annotations, the system leverages contextual inference and error correction mechanisms. If confidence levels fall below a defined threshold, it prompts for human intervention to ensure accuracy.

This comprehensive inference process allows our Doctor Prescription Recognition System to reliably convert handwritten prescriptions into structured digital records, enhancing clinical workflow efficiency and patient safety.

* 1. **Integration and Deployment**

The integration and deployment of our Doctor Prescription Recognition System involve several key steps to ensure smooth operation across model inference, backend services, and user interaction. Our objective is to transform uploaded prescription images into structured, actionable clinical data in real time.

**Step 1: Setting Up the Environment**

* **Virtual Environment Creation:** Establish an isolated environment using tools like virtualenv or Conda to manage dependencies effectively.
  + python -m venv prescription\_env
  + source prescription\_env/bin/activate # For Linux/macOS
  + prescription\_env\Scripts\activate # For Windows
* **Dependency Installation:** Install required libraries (e.g., PyTorch, Transformers, OpenCV, Flask) via a requirements file.
  + pip install -r requirements.txt
* **Verification:** Run a simple script to confirm that all necessary libraries and modules for model inference are installed correctly.

**Step 2: Preparing the Dataset**

* **Dataset Organization:** Arrange the datasets used for training and fine-tuning—including the combined CORD and handwritten Kaggle dataset for text detection and recognition, and the BC5DAR dataset for semantic analysis—into dedicated directories.
* **Preprocessing Pipeline:** Apply image normalization, noise reduction, and region segmentation techniques to ensure high-quality inputs for the model.
* **Data Augmentation:** Optionally augment the dataset to improve model robustness and performance across various handwriting styles and prescription formats.

**Step 3: Backend & Model Inference Setup with Flask**

* **Backend Framework:** Develop the backend using Flask to serve model inference endpoints.
* **API Endpoints:** Implement Flask routes for:
  + **Image Upload:** Handling incoming prescription images from the frontend.
  + **Model Inference:** Processing the image through the text detection, recognition, and semantic extraction pipelines.
  + **Structured Output:** Returning the organized digital prescription data as JSON.
* **Error Handling & Logging:** Incorporate robust error handling to capture low-confidence predictions or ambiguous outputs, and log these events for troubleshooting and model improvement.

**Step 4: Model Deployment & Optimization**

* **Deployment Service:** Deploy the fine-tuned models using Flask, ensuring that the inference service is accessible via RESTful API endpoints.
* **Performance Optimization:** Optimize model inference by integrating frameworks such as ONNX Runtime or TensorRT to reduce latency, and configure the deployment to scale on cloud platforms (e.g., AWS, GCP) for high availability.
* **Monitoring:**  
  Implement monitoring tools to track model performance, API response times, and system resource usage in production.

**Step 5: Frontend Integration**

* **User Interface Development:** Build a user-friendly interface with React that allows users to:
  + **Upload Prescription Images:** Enable drag-and-drop or file selection functionality.
  + **View Results:** Display the extracted and structured prescription data in an intuitive format.
* **API Communication:** Integrate the React frontend with the Flask backend through API calls to:
  + Initiate model inference.
  + Retrieve structured prescription data.
  + Handle error notifications and request reprocessing if necessary.
* **User Feedback Loop:** Optionally include mechanisms for users to provide feedback on the extracted information, facilitating continuous improvement of the system's accuracy.

**CHAPTER 5**

**TESTING AND RESULTS**

* 1. **Evaluation Metrics**

The effectiveness of our Doctor Prescription Recognition System is evaluated across three key tasks using multiple datasets and well-established metrics. For text detection, we assess performance using Precision, Recall, and F-score metrics on a combined dataset comprising the CORD and Kaggle handwritten prescriptions, which captures diverse handwriting styles and imaging conditions. For text recognition, transcription accuracy is measured using similarity metrics such as Jaro Similarity and the Levenshtein Ratio, ensuring that the system faithfully converts handwritten content into digital text. Finally, for semantic extraction, we evaluate the extraction of clinical entities (e.g., drug names, dosages, frequencies) using Precision, Recall, and F1-score, with the BC5DAR dataset serving as our reference standard.

Our system employs a unified deep learning pipeline that integrates a DBNet-based text detection module, a transformer-based text recognition module (similar to TrOCR), and a BERT-based semantic analysis module. The architecture is optimized through extensive pretraining on large-scale datasets, followed by targeted domain-specific fine-tuning. Pretraining experiments are conducted using high-performance GPUs to accelerate convergence, while downstream tasks are optimized for real-time inference on clinical data.

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Label | Samples | Domain |
| Combined CORD+ Kaggle Handwritten | Text Detection/Recognition | 5k+ | Handwritten Prescriptions |
| BC5DAR | Medical NER | 2k+ | Clinical/Prescription Data |

*Table 5.1 Dataset Details*

For evaluation, the same input prescription images are processed through the entire pipeline, and the results are compared against ground-truth annotations. The testing framework ensures consistency across the tasks by applying established benchmarks for each module. The evaluation metrics and results provide a comprehensive insight into the system's capability to accurately detect handwritten text, transcribe complex handwriting, and extract clinically relevant information, thereby streamlining clinical workflows and enhancing patient safety.

* 1. **Performance Comparison**

**Text Detection**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Precision** | **Recall** | **F1 Score** |
| **DBNet** | **0.7911** | **0.9420** | **0.8822** |
| **PANet** | **0.7050** | **0.6912** | **0.7790** |

*Table 5.2 Text Detection Comparison*

The DBNet-based module achieves a high recall rate, indicating that most handwritten regions are successfully detected. With an F-score of 0.8822, the module demonstrates robust performance in accurately localizing text despite challenging image conditions.

**Text Recognition**

|  |  |  |
| --- | --- | --- |
| **Model** | **Jaro Similarity** | **Levenshtein Ratio** |
| **CRNN-based Model** | **0.917** | **0.8879** |
| **Transformer-based Model ( TrOCR )** | **0.934** | **0.9113** |

*Table 5.3 Text Recognition Comparison*

The transformer-based recognition approach (TrOCR) outperforms the CRNN-based model, achieving higher similarity metrics. This indicates a more accurate transcription of handwritten text, which is critical for ensuring the fidelity of digital prescriptions.

**Medical NER**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Precision** | **Recall** | **F1 Score** |
| **BioBERT Model** | **0.8775** | **0.9086** | **0.8928** |
| **Biom-Transformer Model** | **0.8475** | **0.8545** | **0.8509** |

*Table 5.4 Medical NER Comparison*

The BioBERT-based semantic analysis module demonstrates superior performance in extracting clinical entities, with an F1 score of 0.8928 compared to 0.8509 for the biom-transformer. This results in more accurate and clinically relevant data extraction, which is essential for creating structured digital prescriptions.

**5.3 Test Cases**

**i. Test Case I**

**Objective:**  
Verify the OCR engine's ability to extract accurate and complete text from a scanned handwritten prescription.

**Description:**  
This test case ensures that all key fields — patient name, age, date, medicine names, and signature — are properly extracted from the uploaded image.

**Step 1: Upload/Link Prescription Image**

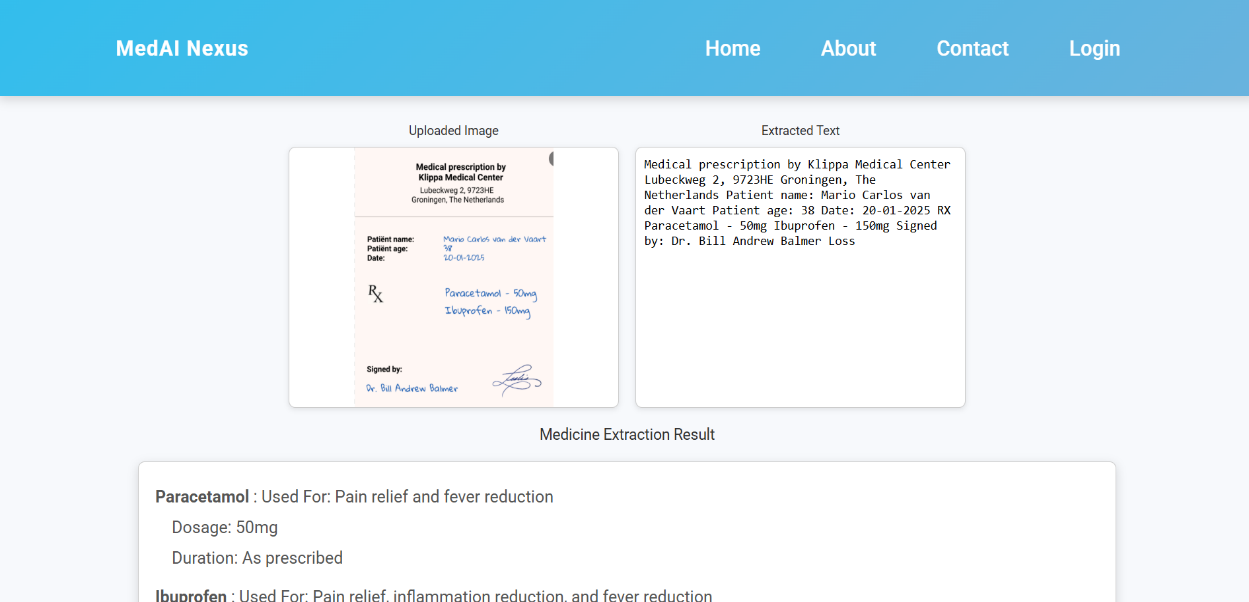
* **Action:** Ensure the prescription image is displayed in the system’s image viewer.
* **Expected Result:** The image appears, clearly showing the handwritten prescription.

**Step 2: Run CR Extraction**

* **Action:** Click the "Convert" button.
* **Expected Result:** A text box shows the complete extracted content.

**Step 3: Verify Extracted Data**

* **Action:** Compare the extracted text with the actual image content.
* **Expected Output:** Extracted fields include:
  + Patient name: Mario Carlos van der Vaart
  + Age: 38
  + Date: 20-01-2025
  + Medicines: Paracetamol - 50mg, Ibuprofen - 150mg
  + Doctor: Dr. Bill Andrew Balmer



*Fig 5.1 Test Case I*

**ii. Test Case II**

**Objective:**  
Test the system's capability to identify and map recognized medicine names to medical usage descriptions.

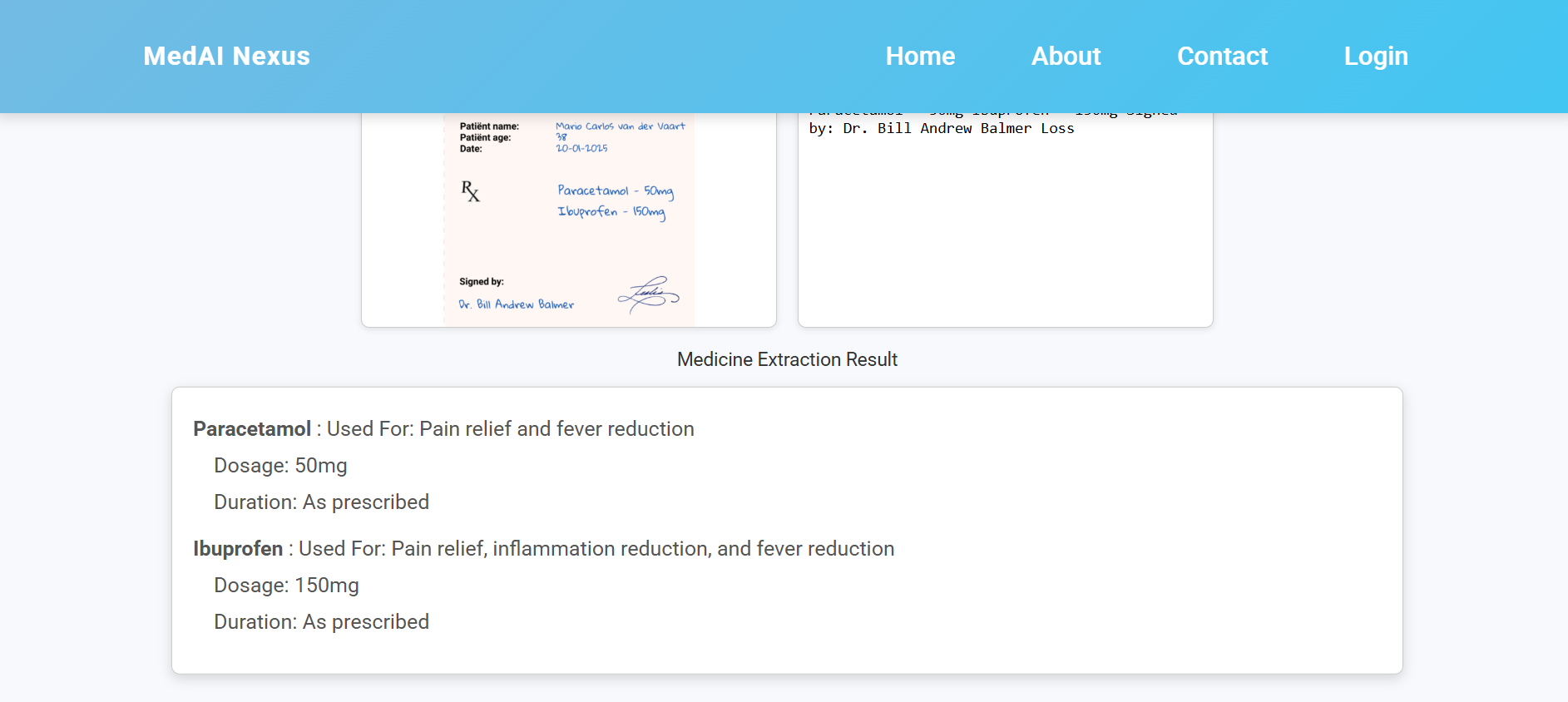
**Description:**  
This verifies if named medicines in the text (like Paracetamol and Ibuprofen) are detected and described accurately using internal knowledge mapping or a medical database.

**Step 1: Ensure OCR Output Contains Medicine Names**

* **Action:** Use the extracted text to identify medicine entities.
* **Expected Result:** Entity recognizer detects "Paracetamol" and "Ibuprofen".

**Step 2: Check Mapping to Medical Use**

* **Action:** Confirm that descriptions for each medicine are correct.
* **Expected Result:**
  + **Paracetamol →** *Used for: Pain relief and fever reduction*
  + **Ibuprofen →** *Used for: Pain relief and inflammation reduction*



*Fig 5.2 Test Case II*

**iii. Test Case III**

**Objective:**  
Validate that the system extracts and formats dosage information correctly for each medicine.

**Description:**  
This test checks whether the system can associate correct dosage values and durations with their respective medicines (e.g., Augmentin, Enzoflam) and format them for clear display.

**Step 1: OCR Extraction Complete**

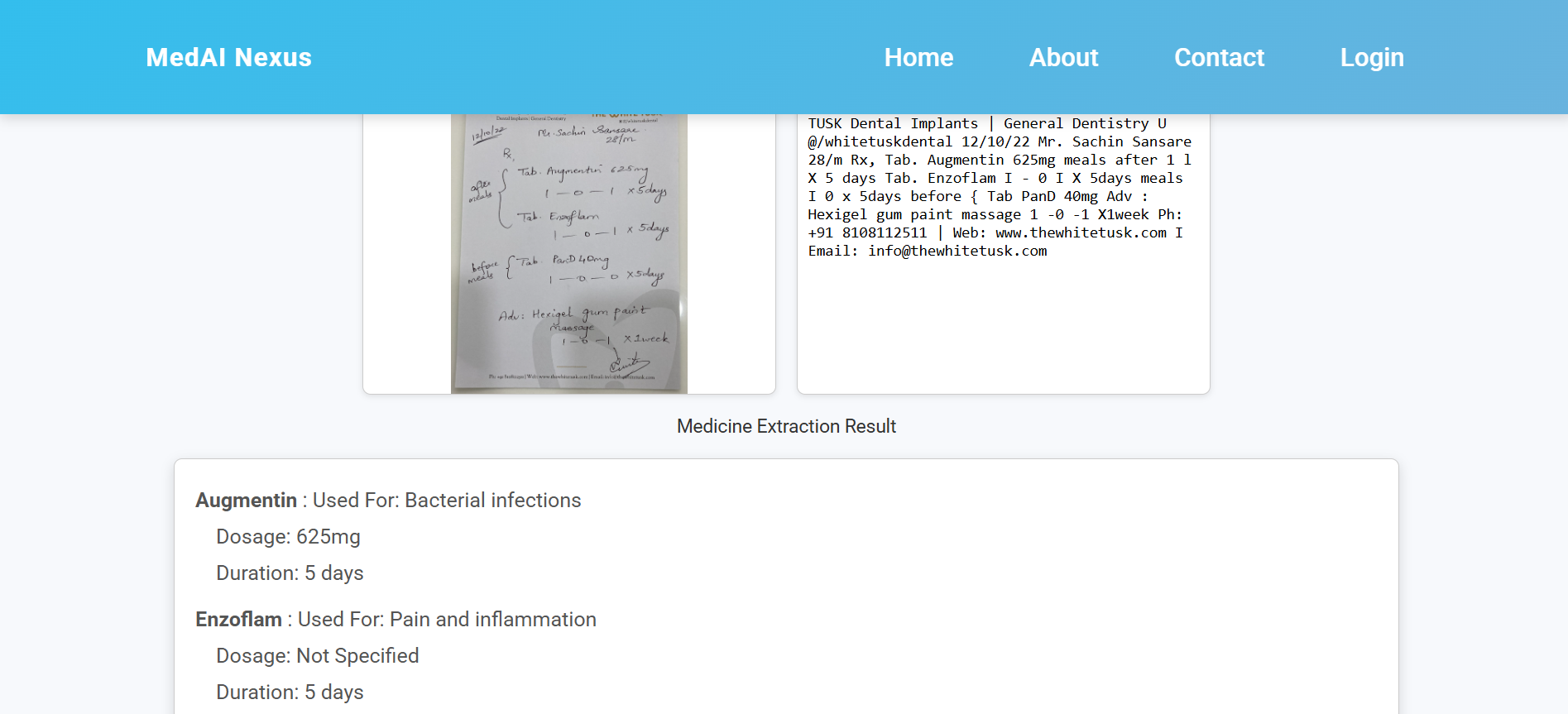
* **Action:** Ensure the OCR output includes medicine names and their relevant dosage and duration information.
* **Expected Result:** Medicines like “Augmentin” and “Enzoflam” are visible in the OCR text.

**Step 2: Dosage Association Logic**

* **Action:** Validate that:
  + Augmentin → correctly mapped to 625mg
  + Enzoflam → shown as “Not Specified”
* **Expected Result:** Dosage values are associated with the correct medicines.

**Step 3: Duration Mapping**

* **Action:** Confirm duration is extracted properly:
  + Augmentin → 5 days
  + Enzoflam → 5 days
* **Expected Output:** Duration values are accurately extracted and shown.



*Fig 5.3 Test Case III*

**Test Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Objective** | **Expected Output** | **Our Output** | **Pass/Fail** |
| OCR Text Extraction Accuracy | Verify the OCR engine's ability to extract accurate and complete text from a scanned handwritten prescription. | Extracted fields include: Patient name: Mario Carlos van der Vaart Age: 38 Date: 20-01-2025 Medicines: Paracetamol - 50mg, Ibuprofen - 150mg Doctor: Dr. Bill Andrew Balmer | Extracted fields include: Patient name: Mario Carlos van der Vaart Age: 38 Date: 20-01-2025 Medicines: Paracetamol - 50mg, Ibuprofen - 150mg Doctor: Dr. Bill Andrew Balmer | Pass |
| Medicine Entity Recognition and Mapping | Test the system's capability to identify and map recognized medicine names to medical usage descriptions. | Paracetamol → Used for: Pain relief and fever reduction Ibuprofen → Used for: Pain relief and inflammation reduction | Paracetamol → Used for: Pain relief and fever reduction Ibuprofen → Used for: Pain relief and inflammation reduction | Pass |
| Dosage Extraction and Formatting | Validate that the system extracts and formats dosage information correctly for each medicine. | Augmentin: 625mg, 5 days Enzoflam: Not Specified, 5 days Final Output Format: Augmentin: Used For: Bacterial infections Dosage: 625mg Duration: 5 days Enzoflam: Used For: Pain and inflammation Dosage: Not Specified Duration: 5 days | Augmentin: 625mg, 5 days Enzoflam: Not Specified, 5 days Final Output Format: Augmentin: Used For: Bacterial infections Dosage: 625mg Duration: 5 days Enzoflam: Used For: Pain and inflammation Dosage: Not Specified Duration: 5 days | Pass |

*Table 5.5 Test Results*

**CHAPTER 6**

**CONCLUSION AND FUTURE ENCHANCEMENTS**

The development of this **handwritten doctor prescription recognition system** marks a significant advancement in the field of medical document processing and automation. The system successfully extracts and interprets handwritten prescriptions, converting them into structured text for further analysis and integration.

By leveraging **deep learning models such as DBNet for text detection, TrOCR for text recognition, and BERT for context-based correction**, the system has demonstrated high accuracy in recognizing complex handwriting patterns. The integration of **Flask-based backend processing** and a **user-friendly interface** enables seamless prescription scanning, improving efficiency in medical record management.

This project has **streamlined the prescription digitization process**, reducing manual errors and enhancing accessibility for both healthcare professionals and patients. By **automating text extraction and interpretation**, the system contributes to **faster data entry, reduced prescription-related errors, and better healthcare record management**.

**Future Enhancements**

Looking forward, several enhancements can further improve the system’s accuracy, functionality, and real-world applicability.

**1. Multi-Language Support**

Currently, the system primarily supports English-language prescriptions. Expanding support to multiple languages, including regional scripts, would enhance its usability for **diverse healthcare settings**. This would involve integrating **multilingual OCR models and language-specific fine-tuning** for accurate recognition across different writing styles.

**2. Improved Handwriting Recognition**

Doctors' handwriting can be highly inconsistent, making recognition challenging. Future improvements could include:

* **Training on larger, diverse datasets** to improve accuracy for various handwriting styles.
* **Context-based correction models** that refine recognition using medical terminology databases.
* **Adaptive learning techniques** that allow the system to improve over time based on user feedback.

**3. Integration with Electronic Health Records (EHRs)**

Enhancing the system to **directly integrate extracted prescriptions into hospital EHR systems** would improve workflow automation. This would allow **real-time synchronization of prescriptions with patient records**, reducing manual data entry for healthcare providers.

**4. Real-Time Prescription Processing**

Currently, the system processes uploaded prescription images. Introducing **real-time recognition through mobile applications or live camera scanning** would provide instant prescription digitization. This could be useful for **pharmacies and clinics**, enabling **faster medicine dispensing and reducing processing time**.

**5. Error Detection and Correction**

To ensure the highest accuracy in prescription recognition, **an intelligent validation system** could be integrated to:

* **Detect potential misinterpretations** by comparing results against a **medical terms database**.
* **Flag incorrect drug names or dosages** for manual verification.
* **Use AI-based suggestions** to improve recognition accuracy based on historical prescriptions.

**6. Offline Functionality**

Enabling offline processing would make the system accessible in **rural or low-connectivity areas**. This could involve **on-device OCR and NLP processing**, allowing doctors and pharmacists to use the system without internet access and later sync results when online.

By focusing on these future enhancements, the project can **evolve into a comprehensive and intelligent prescription digitization tool**, significantly benefiting the healthcare industry by **reducing errors, improving efficiency, and enhancing accessibility**.

# REFERENCES

[1] Jiahui Li, Huiying Ru\* and Chao Li. Text Detection Technology for Complex Natural Scenes Based on Improved DBNet. In IEEE, pages 311-315, 2023.

[2] Huiyang Chen, Jing Liu and Weimin Zhou. Natural Scene Text Detection Algorithm Based on Improved DBNet. In IEEE, pages 186-190, 2022.

[3]Liuqing Xiang, Hanyun Wen and Ming Zhao. Pill Box Text Identification Using DBNet-CRNN. In MDPI.

[4] Saurabh Dome and Asha P Sathe. Optical Character Recognition using Tesseract and Classification. In IEEE, pages 153-158, 2021.

[5] Muhammad Ateeque Zaryab and Chuen Rue Ng. Optical Character Recognition for Medical Records Digitization with Deep Learning. In IEEE, pages 3260-3263,2023.

[6] Minghao Li, Tengchao Lv, Lei cui and Yijuan Lu. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. In Research Gate,2021.

[7] Swapnil Ghosh, Vedang Pokharkar, Sanjeevani Parida, Saiem Pathan, Prof. Palash Sontakke. Medical Prescription Scanner using Optical Character Recognition, In IRJET, pages 1208-1214, 2023.

[8] Xintong Zhao, Jane Greenberg, Yuan An and Xiaohua Tony Hu. Fine-Tuning BERT Model for Materials Named Entity Recognition, In IEEE, pages 3717-3720,2021.

[9] Sulaiman Aftan and Habib Shah. A Survey on BERT and Its Applications. In IEEE, pages 161-166,2023.

[10] Ning Liu, Quian Hu, Huayun Xu, Xing Xu, Mengxin Chen.Med-BERT: A Pretraining Framework for Medical Records Named Entity Recognition. In IEEE, pages 5600 – 5608,2022.

[11] Aditya Yadav, Shauryan Singh, Muzzamil Siddique, Nileshkumar Mehta and Archana Kotangale. OCR using CRNN: A Deep Learning Approach for Text Recognition. In IEEE, pages 1-6,2023.

**APPENDIX**

**Backend :**

from flask import Flask, request, jsonify, send\_from\_directory

from flask\_cors import CORS

import cv2

import numpy as np

import tempfile

import os

from doctr.models import detection\_predictor

from doctr.io import DocumentFile

from doctr.utils.geometry import detach\_scores

import logging

app = Flask(\_\_name\_\_)

CORS(app)  # Enable CORS for all routes

# Define the path for static files

UPLOAD\_FOLDER = os.getenv('UPLOAD\_FOLDER', 'static/images')

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

# Load the detection model

det\_predictor = detection\_predictor(arch="db\_resnet50", pretrained=True)

@app.route("/process", methods=["POST"])

def process\_image():

    if "image" not in request.files:

        return jsonify({"error": "No image uploaded"}), 400

        # Run Doctr detection

        doc = DocumentFile.from\_images([input\_filename])

        results = det\_predictor(doc)

        # Process detection results and draw bounding boxes

        for doc\_img, res in zip(doc, results):

            img\_shape = doc\_img.shape[:2]

            logging.debug(f"Image shape: {img\_shape}")

            # Extract words

            words = res.get("words", [])

            logging.debug(f"Words detected: {words}")

        # Save processed image to the static folder

        output\_filename = os.path.join(app.config['UPLOAD\_FOLDER'], "processed\_image.jpg")

        cv2.imwrite(output\_filename, image)

        with open(input\_filename, "rb") as image\_file:

            response = textract.detect\_document\_text(Document={"Bytes": image\_file.read()})

        extracted\_text = " ".join([item["Text"] for item in response["Blocks"] if item["BlockType"] == "LINE"])

        # Return the URL for the processed image

        return jsonify({"processedImageUrl": f"http://localhost:5000/static/images/processed\_image.jpg", "extractedText": extracted\_text})

from PIL import Image

import torch

from transformers import TrOCRProcessor, VisionEncoderDecoderModel

    # Load the processor and model from the specified local directory

    processor = TrOCRProcessor.from\_pretrained(model\_local\_path)

    model = VisionEncoderDecoderModel.from\_pretrained(model\_local\_path)

    # Preprocess the image to get the pixel values

    pixel\_values = processor(image, return\_tensors="pt").pixel\_values

    # Decode the token IDs to convert them into human-readable text

    predicted\_text = processor.batch\_decode(generated\_ids, skip\_special\_tokens=True)[0]

    return predicted\_text

def \_to\_absolute(geom, img\_shape):

    h, w = img\_shape

    # Convert [x\_min, y\_min, x\_max, y\_max] to absolute coordinates

    x\_min, y\_min, x\_max, y\_max = geom

    return [[x\_min \* w, y\_min \* h], [x\_max \* w, y\_min \* h], [x\_max \* w, y\_max \* h], [x\_min \* w, y\_max \* h]]

@app.route('/static/<path:filename>')

def serve\_static(filename):

    return send\_from\_directory(app.config['UPLOAD\_FOLDER'], filename)

if \_\_name\_\_ == "\_\_main\_\_":

    port = int(os.getenv('PORT', 5000))

    app.run(port=port, debug=True)

**Frontend :**

import React, { useEffect, useState } from "react";

import { useLocation } from "react-router-dom";

import axios from "axios";

import { motion } from "framer-motion";

import "./Extract.css";

const Extract = () => {

  const { imgLink, extractedText } = useLocation().state || {};

  const [medicineExtraction, setMedicineExtraction] = useState("");

  useEffect(() => {

    if (extractedText) {

      axios

        .post("http://localhost:5003/medicine-extraction", { prescriptionText: extractedText })

        .then((response) => {

          setMedicineExtraction(response.data.extraction);

        })

        .catch((error) => {

          console.error("Error during medicine extraction:", error);

          setMedicineExtraction("Error during medicine extraction.");

        });

    }

  }, [extractedText]);

  if (!imgLink) {

    return <div>Loading...</div>;}

return (

    <motion.div

      className="extract-container"

      initial={{ opacity: 0 }}

      animate={{ opacity: 1 }}

      transition={{ duration: 0.8 }}

    >

      <div className="sections-wrapper">

        {/\* Uploaded Image Section \*/}

        <div className="section">

          <h3>Uploaded Image</h3>

          <div className="image-box">

            {imgLink && <img src={imgLink} alt="Uploaded" />}

          </div>

        </div>

        {/\* Extracted Text Section \*/}

        <div className="section">

          <h3>Extracted Text</h3>

          <textarea

            className="textarea-box"

            value={extractedText || "No text extracted"}

            readOnly

          />

        </div>

      </div>

      {/\* Medicine Extraction Result Heading \*/}

      <h3 className="result-heading">Medicine Extraction Result</h3>

      {/\* Medicine Extraction Result Section \*/}

      <div className="result-box">

        <div style={{ whiteSpace: "pre-line" }}>

          {medicineExtraction

            ? medicineExtraction.split("\n\n").map((block, index) => {

                const lines = block.split("\n").map((line) => line.trim());

                if (lines.length >= 3) {

                  let medicineName = lines[0].replace("Medicine Name:", "").trim();

                  medicineName = medicineName.replace(/:$/, ""); // Remove trailing colon if present

                  return (

                    <div key={index} style={{ marginBottom: "15px" }}>

                      <p>

                        <strong>{medicineName}</strong> : {lines[1]}

                      </p>

                      <p style={{ marginLeft: "20px" }}>{lines[2]}</p>

                      <p style={{ marginLeft: "20px" }}>{lines[3]}</p>

                    </div>);}

                return <p key={index}>{block}</p>;

              })

            : "No extraction result available"}

        </div></div>

    </motion.div>);};

export default Extract;