Faculty of Engineering - Cairo University
Credit Hour System Programs

Communication and Computer Engineering

CCE-C

CMPN481

Graduation Project Report

Spring/2022

SCAN-MED

Prepared by:

Aya Hossam ElDien Hanafy

Kareem Hossam El Din

Supervised by:

Dr. Michael Nawar

Cairo University
Faculty of Engineering
Department of Computer Engineering

SCAN MED

A Graduation Project Report Submitted

to

Faculty of Engineering, Cairo University

in Partial Fulfillment of the requirements of the degree

of

Bachelor of Science in Computer Engineering.

Presented by

Aya Hossam ElDien Hanafy Kareem Hossam El Din

Supervised by

Dr. Michael Nawar

June 20, 2022

Project Code	GP-N481-2022	
Project Title	SCAN-MED	
Keywords	Machine Learning, Natural Language Processing, Convolution Neural Network, Computer Vision, Optical Characters Recognition	
Students	**Name:** Aya Hossam ElDien Hanafy  **E-mail:**aya.hanafy95@eng-st.cu.edu.eg **Phone:** +2 01010654510 **Address:** New Cairo, Egypt	**Name:** Kareem Hossam El Din  **E-mail:** kareem.elsokary97@eng-st.cu.edu.eg **Phone:** +2 01118082444 **Address:** Giza, Egypt
Supervisor	**Name:** Dr. Michael Nawar	**E-mail:** michael.nawar@eng.cu.edu.eg
	Signature:	**Phone:** +2 01205536262
Project Summary	This Project is divided into 2 main parts: Digitizing Doctors' Handwritten Prescriptions, and Digitizing Arabic Medical Forms. The first part can take any image scanned as an input and extract the correct medicine with the right dosage, and the second part extracts the important information from an admission medical form and the output of either parts is to be kept in our database.	

Abstract

Not all of us a have perfect handwriting or professional typing skills, due to various circumstances. Maybe growing up in a world full of technology and smartphones had its impact on many of us yet, it shall have more and more impact on the generations to come with respect to data storage and interpreting.

Having to read handwritten documents can sometimes be hard, we all face that struggle, governmental workers converting thousands of data into digitized documents and typing excel forms, teachers who have to read thousands of MCQ exam answers and mark them accordingly, and the most struggle for the pharmacists and hospital check-desk officers who have to deal with the doctor’s prescriptions handwritings whether in serving patients with medicine prescribed or inputting the medicines info in their database system.

Our objective is to provide health-care workers with a clear digital Document/Form that can of course be read & understood by anyone concerned and related to the medicine field, and secondly, promoting the data recording process in hospitals, pharmacies and medicine companies’ warehouses.

Dealing with digitized documents/forms out of scanned hardcopies/photocopies along with an automated recording process will make check-outs and daily routines quite easier, faster, more efficient and less erroneous, not only because it will be more readable, but also because we can use those digitized documents for further statistical analysis as a future work, for example: how many patients ask for antibiotic X under the prescription of Doctor Y so we can counter-fight Antibiotic-Abuse rising issue lately.

Our project will simply require the user to take a picture of/electronically scan the document/ form/ prescription receipts -in a specific format- that he/she wants to record into the system, and then our project aims to firstly perform some image enhancing preprocessing techniques on the input image, decreasing any noise/shadows if detected, crop the image for our page scope if needed, and deal with as many as skew/ rotation/imperfections. Afterwards, take the input enhanced image will pass from the preprocessing stage to the segmentation stage in which we will have an output of each character separated, and then finally, aiming to recognize the characters / numbers (English using our optical character recognition technique, hence an output text file with the recognized characters will be available and automatically fed into a record database system as our final end result of the project.

الملخص

نحن لا نمتلك جميعًا مهارات الكتابة اليدوية أو الكتابة الاحترافية ، نظرًا لظروف مختلفة. ربما كان للنشأة في عالم مليء بالتكنولوجيا والهواتف الذكية تأثيرها على الكثيرين منا حتى الآن ، سيكون لها تأثير متزايد على الأجيال القادمة فيما يتعلق بتخزين البيانات وترجمتها.

قد يكون الاضطرار إلى قراءة المستندات المكتوبة بخط اليد أمرًا صعبًا في بعض الأحيان ، فنحن جميعًا نواجه هذا الصراع ، حيث يقوم العاملون الحكوميون بتحويل آلاف البيانات إلى مستندات رقمية وكتابة نماذج Excel ، والمعلمين الذين يتعين عليهم قراءة الآلاف من إجابات امتحان الاختيارات المتعددة ووضع علامة عليها وفقًا لذلك ، وأكثر صعوبة من أجل الصيادلة وموظفو مكاتب الفحص بالمستشفى الذين يتعين عليهم التعامل مع كتابات وصفات الطبيب سواء في خدمة المرضى بالأدوية الموصوفة أو إدخال معلومات الأدوية في نظام قاعدة البيانات الخاصة بهم.

هدفنا هو تزويد العاملين في مجال الرعاية الصحية بمستند / نموذج رقمي واضح يمكن بالطبع قراءته وفهمه من قبل أي شخص معني ومرتبط بمجال الطب ، وثانيًا ، تعزيز عملية تسجيل البيانات في المستشفيات والصيدليات وشركات الأدوية. المستودعات.

التعامل مع المستندات / النماذج الرقمية من النسخ الورقية / النسخ الممسوحة ضوئيًا جنبًا إلى جنب مع عملية التسجيل الآلي سيجعل عمليات السحب والروتين اليومي أسهل وأسرع وأكثر كفاءة وأقل خطأ ، ليس فقط لأنه سيكون أكثر قابلية للقراءة ، ولكن أيضًا لأننا يمكن استخدام هذه المستندات الرقمية لمزيد من التحليل الإحصائي كعمل مستقبلي ، على سبيل المثال: كم عدد المرضى الذين يطلبون مضادًا حيويًا "س" بموجب وصفة الطبيب "ص" حتى نتمكن من مكافحة مشكلة تعاطي المضادات الحيوية المتزايدة مؤخرًا.

سيتطلب مشروعنا ببساطة من المستخدم التقاط صورة / مسح ضوئي إلكترونيًا للمستند / النموذج / إيصالات الوصفات الطبية - بتنسيق محدد - يريد تسجيله في النظام ، ومن ثم يهدف مشروعنا أولاً إلى إجراء بعض تحسينات الصورة. تقنيات المعالجة المسبقة على صورة الإدخال ، وتقليل أي ضوضاء / ظلال إذا تم اكتشافها ، واقتصاص الصورة لنطاق صفحتنا إذا لزم الأمر، وتعامل مع العديد من الانحرافات / التدوير / العيوب. بعد ذلك ، ستنتقل الصورة المحسّنة المدخلة من مرحلة المعالجة المسبقة إلى مرحلة التجزئة حيث سيكون لدينا إخراج لكل حرف مفصول ، ثم في النهاية ، بهدف التعرف على الأحرف / الأرقام (اللغة الإنجليزية باستخدام تقنية التعرف الضوئي على الأحرف لدينا ، ومن ثم سيكون الملف النصي الناتج مع الأحرف المعترف بها متاحًا ويتم إدخاله تلقائيًا في نظام قاعدة بيانات السجلات كنتيجة نهائية نهائية للمشروع.

ACKNOWLEDGMENT

First, we would like to thank, and to express our gratitude to **GOD** for guiding us and providing us with the patience and perseverance to implement this project.

Then, we would like to thank our parents for their support, prayers and patience they gave us from birth up till now.

Furthermore, we would like to express our utmost gratitude to ***Dr. Michael Nawar*** for his guidance, patience, and his unmatched expertise.

We hope and looking forward that our project would be a successful initiative to honor Cairo University, Faculty of Engineering, Computer Engineering Department and ultimately, be a benefit to our community for better life.

Aya, and Kareem

Table of Contents

List of Figures

Contacts

Team Members

Name	Email	Phone Number
Aya Hossam ElDien Hanafy	aya.hanafy95@eng-st.cu.edu.eg	+2 01010654510
Kareem Hossam ElDin	kareem.elsokary97@eng-st.cu.edu.eg	+2 01118082444

Supervisor

Name	Email	Number
Dr. Michael Nawar	michael.nawar@eng.cu.edu.eg	+2 01205536262

List of Abbreviations

CNN	Convolutional Neural Network
RNN	**Recurrent Neural Network**
CTC	**Connectionist Temporal Classification**
LSTM	**Long-Short Term Memory Cell**
OCR	**Optical Character Recognition**
SVM	**Support Vector Machines**
HMM	**Hidden Markov Models**
KNN	**K-Nearest Neighbor**
PNN	**Probabilistic Neural Network**
BDD	**Background Directional Distribution**

List of Tables

This page is left intentionally empty

SCAN MED

Chapter 1: Introduction

1. Introduction

Since the mid-20th century, we have been living in the Digital Age or the Information Age. We can say that everything is now smart, from the tiniest headphone to the biggest self-driving car. It’s the right era for us to think of digitizing our hard copied documents, and make it easier for all of us to manipulate and store our data.

The focus of this project is creating a tool with which you can easily convert a handwritten documents into a text file that can be viewed on any device. Extracting the text from an image can be very useful even as an input too many other applications, such as document analysis, text-based image indexing, etc.

1.1. Motivation and Justification

Many of us spend a lot of time and effort searching for a specific word in a hard-copied document, having to re-write dozens of words by hand, then consuming a lot of time typing all of this back again on the computer moreover, such exhaustive process is prone to many human errors.

Having a hard copy of any document can sometimes be necessary, but what should we do if we want to easily convert it into a digitized version? We cannot deny the fact that a computer version of our written document is way more readable than ours. We aim to save time for others by converting their documents into a soft copy that can be easily edited, stored, or even let it pass by a summarizer and modify it.

1.2. The Essential Question

Our Solution resolves around one main questions; how can we make both the patient and the pharmacist’s life easier? Answering that question is how we address the problem of the hassle the patient is put through to read a doctor’s prescription, and the time wasted between physically going to a pharmacy, putting all the load on the pharmacist and then placing the medicine order.

Our mission is to provide an automated method to digitize medical documents, whether it’s a prescription, a medical administration form, or even just an image of a medicine. We are targeting methods that digitize scanned documents or images. Our application will be essential for any medical corporation. The project allows us to learn and apply concepts of Natural Language Processing as well as Software Development. Our Project will give us the chance we needed to apply what we have learnt in a real life problem and get to experience the real engineering market orientation.

1.3. Project Objectives and Problem Definition

Having to read and interpret a careless handwritten prescriptions/receipts can be a bit of a challenge, and since ages everyone suffers from reading doctors' handwriting, and pharmacists suffer from the same thing too, the only difference between a pharmacist and everyone else while reading the prescription, is that the pharmacist got used of such handwriting, so he automatically predicts the written medicine from the first and second letter and also from the given dose.

According to the studies, people can sometimes suffer from Typo glycemia, which is the principle when readers can comprehend a text despite the fact it has spelling errors or misplaced letters in the words. Having that said and with the knowledge of knowing that catching an error or a typo in a handwritten document is like looking for a needle in a haystack. For this specific matter, another local survey was conducted to endorse our studies.

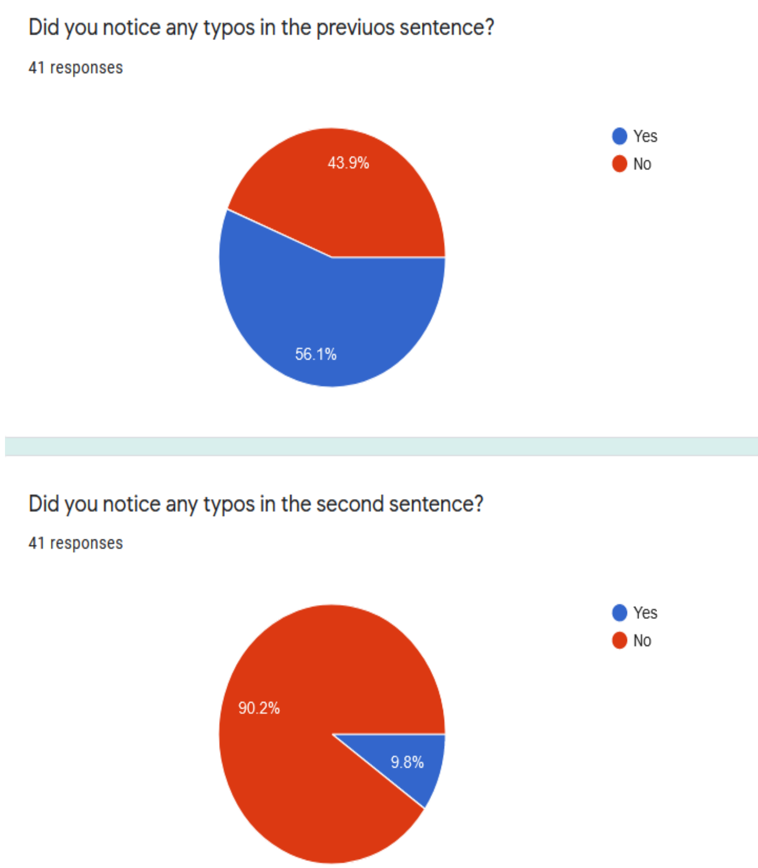


Figure1.31: Survey made to see whether human eye can catch tricky typos

Here in this survey, we asked multiple questions that are dependent on the previous one, and by "the second sentence" we meant in the question that was "Did you notice any typos in the previous sentence?", and as we see from the figure above, 82% of the people didn't notice that the word "previous" is misspelled in the question. But we know that a typo like this can be easily caught if it was a digital document.

Handwritten document errors can lead to a series of bad predictions, but this is not the only issue that we are seeking to overcome, but also gathering information about the required medicine from the corresponding location in store, and giving it out to patients can be a bit of a challenge too. However, spending hours and hours on a daily basis only for the purpose of doing such routine for dozens and dozens of patients queuing at the checkout is such a hassle. Moreover, in hospital or clinics the process of gathering some personal information and medical history of patients' through forms is a repeated time consuming process, specially while afterwards entering this whole data to the system word by word and checking by the human eye to check for the similarity or whether this patient visited the hospital before or not, and at the end of the day, it will not please the patients nor the employers in terms of the slow process and errors occurrence in such a critical field

Moreover, after doing some research, we found out that a typical worker spends more than 2 hours a day only trying to find and recall information in handwritten documents.

In order to overcome those issue, our project objectives are to create a windows application that:

- Converts and digitize English hand-written forms into a digital recognized file.
- Prediction of a handwritten medicine from a handwritten prescription, and check for its availability in our database.
- Records the recognized meaningful data records into a database system, collecting related information such as (name, product id, dosage, expiry date, and other important remarks).
- Saves time and work load, our application shall seek real-time processing.
- Eliminates errors in data records or retrieval, our application shall check for spelling errors in a logical way from a dataset of medicines.
- Stands as database system and a repository of records where we can statistically analyze for trends or correlations-relationship in a time period.
- Keeps a user-friendly GUI for various targeted front end users and system admins.

1.4. Project Outcomes

Moving from a paper-based document commutation routine to a digital-automated platform is a big leap for any company or organization. And for this reason, this project will not only contribute to healthcare institutions, but will also contribute to other commuters-employed organizations. Users can benefit from digitizing their own handwritten documents reducing time wasted reading, comprehending, re-writing and filling it into a digital form on a computer-based system.

Some countries went through an easier path of using an electronic medical prescription form in which the doctor prescribes the medicine on the computer and a medical prescription is generated and ready to be printed. However, this method has not been applied yet in Egypt. Accordingly, the proposed system is going to admittedly breakthrough in Egypt due to its high demand and importance as a survey was conducted to clarify whether the application would be beneficial.

Lastly, our project can be expanded and exploited in a computer-interactive robot-based system where the recording, storing, prescribing, retrieving from stores and serving medicines to patients at hospitals/pharmacies checkout windows in a completely automated system.

By the end of this project, we expect to have a tool with which every doctor, patient, hospital can benefit from it. Hospitals will use our tool to easily retrieve data from a given patients' forms, our system will automatically save the data of each patient in our database, our output database can be used to see whether this patient has a medical history and needs immediate care or not (Checking patients' status).

Our application also, accepts handwritten doctors' prescriptions', and even though that the prescription can suffer from a bad handwriting, or a misspelling word, our application can overcome those issues through our prediction module, that automatically predicts the medicine from some letters and from the given dose. Prediction of any word is done whether the input is a medical form or a handwritten prescription, so we will not worry about any typos or misspelling.

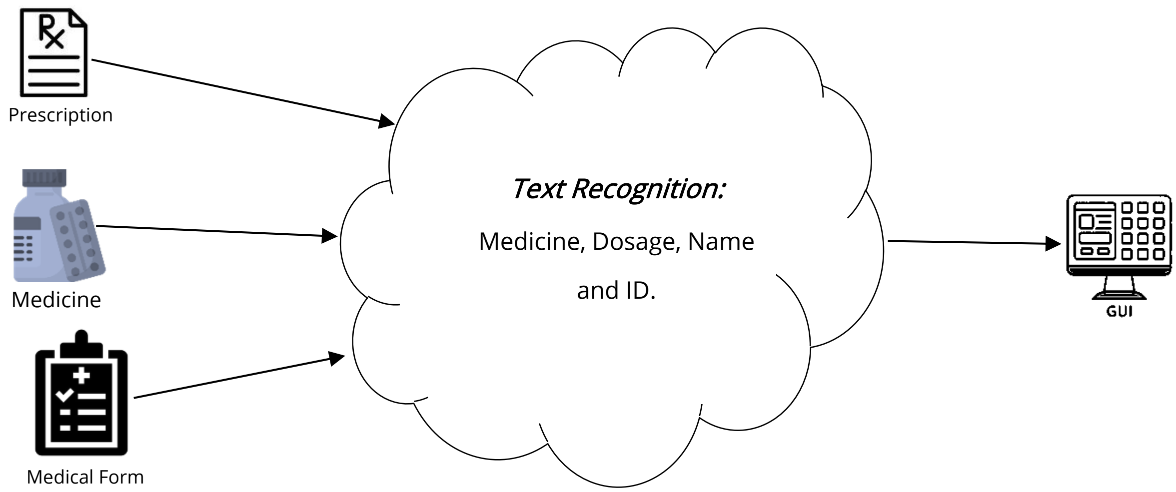


Figure 1.2: Project Outcome Overview

Our application can be easily integrated to any hospital/pharmacy system to have the best accuracy of search and prediction.

Our application can be a prototype and enhanced later on to have more features, it can be available for pharmacies, and make it easier for the public to easily scan prescriptions, and have their medicine delivered to their doorstep.

The main contribution of this project is proposing a solution system for both the pharmacist and patients through providing an application that recognizes and reads doctors' handwriting in the medical prescription and returns a readable digital text of the medicine and its dose, and with just one click that medicine can be delivered to the patient's doorstep.

1.5. Document Organization

The purpose of this document is to provide business and technical details related to the project. The document is divided into 6 chapters, each of which has its own functionalities and importance. However, the chapters are organized in logical manner in order to walk both technical and non-technical users through the project without the need to do further readings. An appendix is added to the document for side-readings.

In this chapter, we gave an introduction to the project where the reader is expected to have a general overview for the whole project, and we also justify the motivation behind it and give the reader a sense of what to expect from the rest of the document.

The second chapter discusses the business aspect of the project where market analysis and surveying are explained in details. We start by giving the analysis for the potential market pool by discussing the survey we conducted to evaluate the need for the project with respect to different market segments. The result of the analysis define the steps we need to take to target specific segments of the market and define their needs that we need to focus on. We follow that by analyzing the current market with respect to competitors; we define what significant competitors are and how they address the market. We conclude the chapter with business case study upon which our plans to compete in the market and win profit are discussed. The next three chapters focus on the technical aspects of the chapter.

The third chapter focuses on the literature analysis; we give a comprehensive technical background on the topics we used. We start by giving some non-technical background on the history of the problem. We start with the history of NLP as a whole. We follow that by giving the technical background required to understand the project implementation; we explain some statistical machine learning concepts relevant to our project, then we give some deep learning concepts where we discuss some artificial neural network architectures mainly, and other important notions. We then give some background related to the services as a product where we discuss the architecture of networking upon which our project is implemented.

By now the reader should have the sufficient business and technical background to understand the models we tested and implemented. Section 3.3 discusses the comparative study between the approaches that we investigated in attempt to solve our problem. We discuss their intuition and give model architecture details then conclude each of them with a small critique stating the pros and drawbacks of each. Finally, we conclude the chapter with the approaches we chose to implement and justify the choices.

In Chapter four, we discuss the system architecture. We start by giving a very abstract overview of the system stating the high-level architecture, assumptions we made and how the architecture was designed. We then give the details of the designed architecture; we explain the architecture in top-down approach where we start by explaining the big picture in the block diagram, then we explain each module on its own. For each module, we give the full functional description, the modular decomposition and the design constraints enforced by this model.

The fifth Chapter discusses the system testing and verification pipeline we used to ensure the quality of the system. This chapter is of significant importance to the user; it gives the details of the performance and reliability of the approaches we used in a way, and the final product in another way. We start by stating the setup required to test the project. We then give the detailed plan we designed and followed in order to test the system in an efficient way. We then give detailed test scenarios for the pipeline modules and for the levels of integration as well. We discuss the schedule we followed for testing, then conclude with comparing the performance of our product with other approaches. Since, as far as we researched, we are the first platform to combine multiple solutions, we perform the comparative study of the modules on their own.

The sixth Chapter is the conclusion and future work. We state the overall self-critique of the system stating what we believe to be strengths of our platform and the weakness points. Then we state what we plan for future work to be added to the project. The appendix contains the software tools and libraries we used to develop the project, the use case diagram, the user guide and the feasibility study of the project.

Chapter 2: Market Feasibility Study

2. Market Feasibility Study

In this chapter, we discuss the market visibility analysis for the proposed project; the intended customers, the targets market segments, and the competitor analysis. The first step to evaluate a product is to conduct an accurate and detailed market analysis to identify how one can position himself in the market. Then, we discuss the outcomes of the investigation; we start by analyzing potential customers, and try to anticipate how the market would expand in both the best and worst cases.

We have conducted a quite representative and reliable market analysis to evaluate our proposed project. In this chapter, we will discuss the outcomes of inquiries; we start by analyzing targeted potential customers and then other who may have interests in such project approach. Lastly, we follow by discussing the similar approaches that are currently present in the market, and how we will try to distinguish and differ ourselves from them.

The survey we have conducted in section 2.2 is a local market survey that was uploaded in WhatsApp, Facebook Medical groups for doctors, medicine and clinical pharmacy under-graduates, google forms and several other social media platforms related to our concerned field.

2.1. Potential Market Evaluation

In this section, we discuss the market survey we conducted to evaluate the potential market acceptance of the idea. The goal of this survey is to help us define the next steps; target customers and the market tackling plan. The survey was conducted online through social media.

The targeted pool consisted of 236 respondents distributed as shown in Figure 2.1 below. The distribution of industry related respondents includes that all the respondents are doctors to ensure realistic insights.

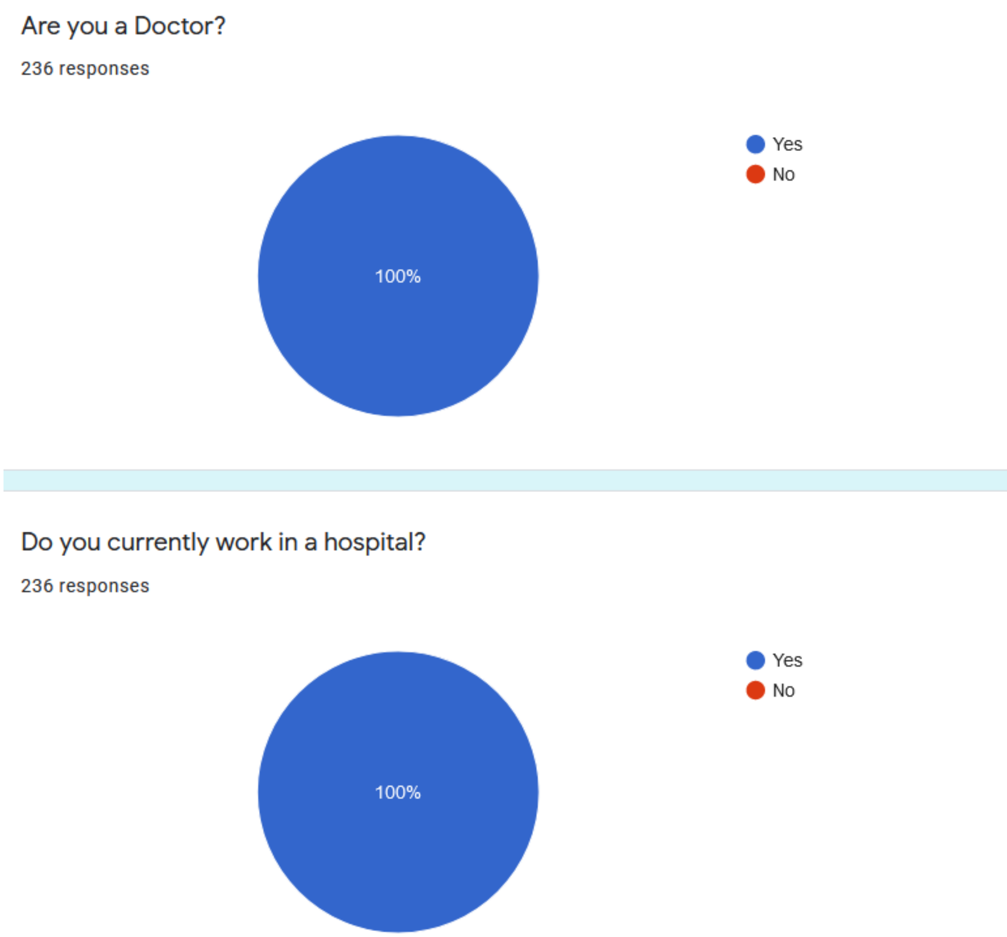


Figure 2.1: Survey to take responses from only doctors whom work in hospitals

2.1.1. The Need for the Platform Idea

In this survey section, we wanted to know the potential need for the abstract idea. So we asked if it's important for patients to record their medical history in forms, and as we can see in figure 2.2 we received almost 84% saying that yes, it's important. And even thought that 14% voted a maybe as a response, but that doesn't mean that disagree, but it means that sometimes hospital have old patient records, or sometimes patients only want to perform some medical examinations that don't require any knowledge of information, but that is not a common case, and this is the doctors' response when we asked what "maybe" means in their responses.

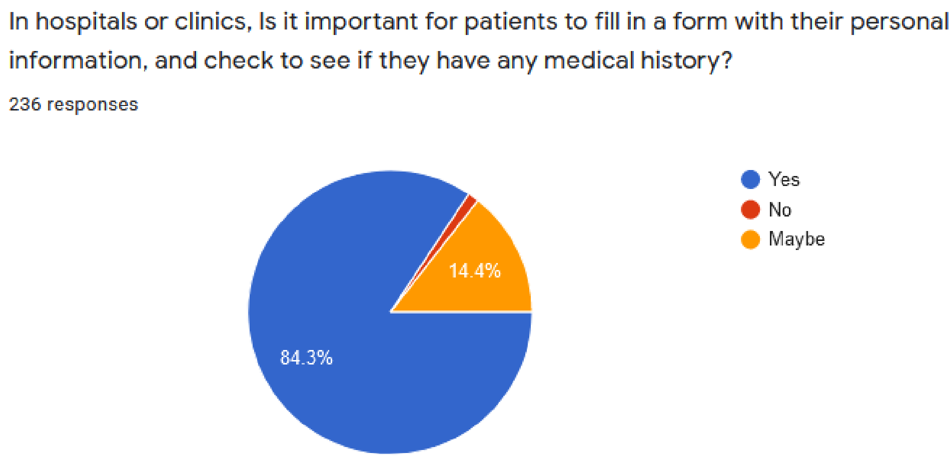
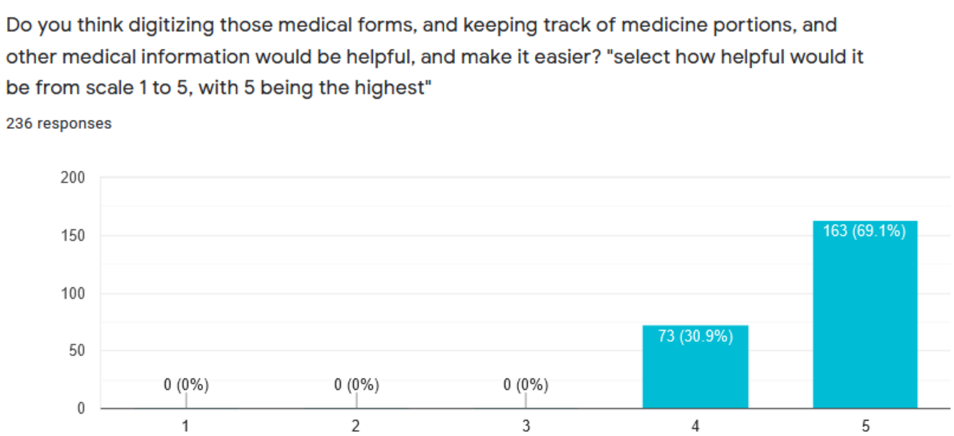


Figure 02.2: Doctors' response on how important patients filling forms are

2.1.2. The Need for Digitizing Scanned Medical Documents

These questions group includes in it one main important purpose which is to evaluate the potential need for a platform that helps patients/pharmacists/doctors extract the useful information of a medicine or a prescription from a scanned image with the intent of fastening a part of their day-to-day work routine. This purpose defines the potential strength of our automatic document digitization module.

We asked doctors to rate having their documents digitized would be helpful or not from scale 1 to 5, and as we can see in figure 2.3 we received most of the voted for that it would be really very helpful, and make their life extremely easier.



Not-Easy Slightly-Easier Fairly-Easier Easier Extremely Easier

Figure 02.3: Responses for the importance of digitizing medical forms

2.2. Targeted Customers

The project nature implies that interest in it can be gained from both fields of individuals and industry. We start by dividing our customers into three categories based on their interests: individuals and employees who work for health care institution. The second category, industry customers, includes medical organizations, entities, and companies. The third category, academic customers represented in research labs, university departments.

We used the survey results from the previous section to apply a narrower segmentation of our target market. As shown in section 2.1, the results agree highly with our intuition of the project; the most interested segments in our product are the management of industrial medical entities, individuals who work for health care institution, and people in academia. We focus on their special needs as the backbone of our product.

2.2.1. Individuals

At the top, individuals and employees who works for health care institution:

- Hospital check-out windows workers.
- Pharmacists.
- Medicine Warehouses admins.
- Doctors.
- Health-care Inspectors.
- Nursing Personnels.

2.2.2. Industry Entities

The second cluster of potential users we focus on are the medical industry entities, specifically those in management levels. Those include:

- Governmental institutions for Inspection & Safety.
- Health-care Ministry.
- Hospitals.
- Clinics.
- Medicine Companies.
- Pharmacies.
- Import and Export Companies.

2.2.3. Academic Entities

Lastly, individuals who may be interested in such project foundation, of which including:

- University Academic Departments.
- Other Governmental institutions Commuters.
- Airports Check Desk Workers.

2.3. Market Survey

The need for current market research did not drive from the sole fact of positioning oneself in market. However, it was a major step in the product development process; we needed to assess popular approaches to similar projects, investigate their points of superiorities as well as weaknesses. We represent the functional and non-functional specifications of each product as well.

2.3.1. Vezeeta

Vezeeta is one of the leading digital healthcare booking platform and practice software. The application/website has over 15k doctors [6].

Features:

- Booking appointments with doctors.
- Scan Doctors' prescription.

Downside:

- No real time OCR conversion of the prescription.

Figure 2.4: Vezeeta Platform

Our Strength and Goal:

We are hoping to implement a real time processing application that outputs the data right away within a non-significant delay during use.

2.3.2. OCR Medical Claims Systems Capture

The OCR medical and healthcare is a document scanner and imaging software.

Features:

- Capture any type of document from any source.
- Eliminate manual data entry.
- Identify and sort documents using supervised machine learning.
- Extract the metadata from documents.
- Claim related data extraction with improved accuracy/quality.

Figure 2.5: OCR Medical Claims Platform

Downside:

- Only online trial is available, and requires verification that you are with a medical institution.
- Available in Florida, USA [5].

2.4. Business Case Plan

Following the normal-premium user scheme, our business plan is based on two major phases: Market Attention and Market Absorption. The platform will be available for free usage during both phases, while using the APIs will be only for premium subscribed users. Subscriptions are billed according to requests per day with minimum daily fare of 1 USD per license. Licenses are issued for single user acquisition, cancellation and refund policies follow the standard MIT rules. For free users, they are granted a free clouding service to save their work up to 1 GB for life. Users can upgrade their storage with 2 different subscriptions:

1. Pro License: user has 10 GB for 2.99 USD/month

2. Ultimate License: user has 250 GB for 9.99 USD/month

Upon user cancellation or pause, their storage will be archived for 30 Days before re-marketing.

Chapter 3: Literature Survey

3. Literature Survey

In this chapter, a literature survey will be conducted. In the next subsection, we give a brief background on the non-technical aspects in the project; defining important concepts that are important for the full understanding of the project's idea. Then, we give a brief background on the technical aspects; discussing important algorithms and methods that are useful to know, to understand how we built the project.

We will focus on introducing Machine Learning and NLP since the system hugely depends on understanding these two sciences.

3.1. Non-Technical Background

In this section, we will discuss some general concepts and ideas that will illustrate the non-technical aspects in our project, providing a clear and bright understanding of the ideas discussed throughout the project.

3.1.1. Natural Language Processing

Natural language processing (NLP) is a branch of linguistics, computer science, and artificial intelligence that studies how computers interact with human language, particularly how to design computers to process and analyze massive amounts of natural language data. As a result, a computer can "understand" the contents of documents, including the intricacies of the language used within them. The system can then extract accurate information and insights from the papers, as well as categorize and organize them.

There are countless applications of NLP ranging from simple text processing like string matching and semantic search, up to voice recognition and speech-controlled systems. NLP is one of the most active and required fields in both industry and academia.

Applications of NLP

- Text and speech processing.
- Morphological analysis.
- Syntactic analysis.
- Lexical semantics (of individual words in context).
- Relational semantics (semantics of individual sentences).
- Discourse (semantics beyond individual sentences).
- Higher-level NLP applications.

3.1.2. Machine Learning

Machine learning (ML) is the study of computer algorithms that improve themselves over time as a result of experience and data. It is considered to be a component of artificial intelligence. Machine learning algorithms create a model based on sample data, referred to as "training data," in order to make predictions or judgments without being explicitly programmed.

The learning agent, called model, attempts to generalize from its experience, and tries to classify new data based on past examples it learned. The more training data there is to learn from, the more accurate the model becomes at classifying. The complexity of the hypothesis should match the complexity of the function underlying the data for the optimal generalization results. The model has under fitted the data if the hypothesis is less complex than the function. The training error lowers when the model's complexity is increased in response. However, if the hypothesis is too complicated, the model will be prone to over fitting, resulting in poor generalization.

3.2. Technical Background

In this section, we will discuss some techniques and algorithms that will illustrate and help in the understanding of the approaches and methods used throughout the project.

3.2.1. Artificial Neural Networks

Artificial neural networks, commonly referred to as connectionist systems, are computer systems based on organic neural networks found in animal brains. Such systems learn to perform tasks by analyzing examples, frequently without the use of task-specific rules. An artificial neural network (ANN) is a model made up of "artificial neurons," which are connected units or nodes that are generally modelled after neurons in a biological brain. Each link may convey information, or a "signal," from one artificial neuron to the next, similar to the synapses in the human brain. An artificial neuron can receive a signal and process it before sending it to other artificial neurons.

Classification of data is a key fundamental of artificial intelligence. The traditional solution is to have some kind of decision function which is always a higher order polynomial function. However, when the data become complex, construction such decision function becomes harder and non-intuitive; it is then subjected to cases of under fitting where your function is not performing well as a class separator, and sometimes overfitting where the function is tailored on the formerly observed data and misses the fact of generalizing the boundary functions. Even when you have a reasonable fitting of your data, having higher accuracy of classification is not always the result.

The idea suggests hence that we need some kind of universal mapping structure where simple changes to the structure result in more complex decision boundaries. If we considered the ultimate intelligent thing man has ever known; the brain, we would then see how we can mimic its structure for machines.

The human brain consists of billions of neurons connected together forming multiple complex networks working together in parallel and in series to learn new behaviors. Each neuron consists of three main parts as shown in Figure 3.1 below: the input axon where pulses are sent to this neuron from other neurons, the body where processing of the pulse signal is applied, then activation where the neuron transmits new signals to other cells.

Likewise, the ANN (Artificial Neural Network) consists of many neurons, each of which has input, output and an activation function; which maps the input into more complex shape in the output.

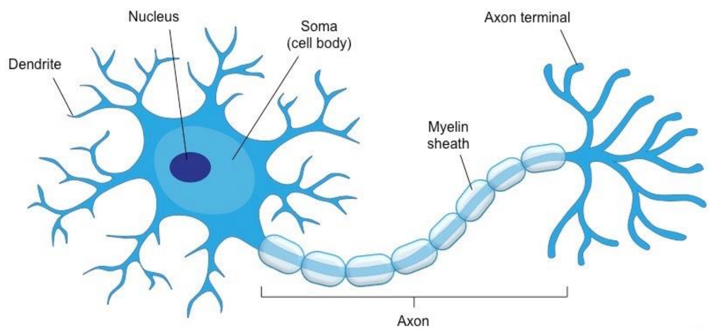


Figure 3.01: Neuron Cell Structure

Neurons of the same level form a layer. Layers are divided into 3 types; input layer which accounts for the input data features we want to learn to classify, output layer which produces the final decisions of the network, and hidden layers where actual learning occurs. Figure 3.2 below shows an example of two-hidden-layer neural network.

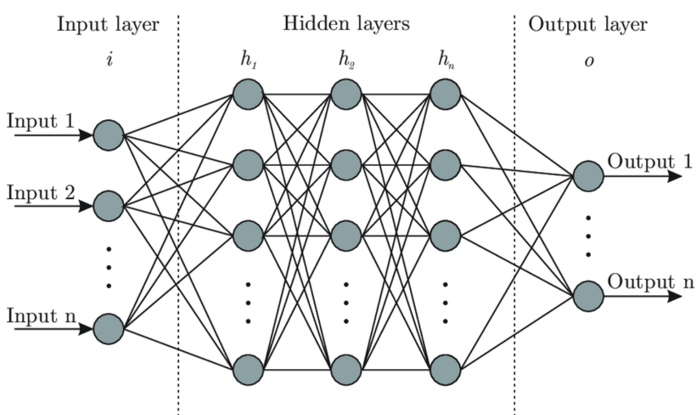


Figure 3.02: Artificial Neural Network Structure

Using an algorithm called Back Propagation, the neurons, also called nodes, learn to efficiently process their inputs into advanced representations that help making final decisions. This allows for building very complex classifiers in a systematic way with high accuracy and less error-prone algorithms known as optimization algorithms.

The ANN algorithm consists of three phases: the forward pass, loss calculation, and the backward pass. The forward pass, also called forward propagation, is where the input features of the input layer are propagated through each layer until the final decision is made at the output layer. During this pass, each neuron, or node, must map its inputs to an output signal, adding another piece of information useful for making the decision.

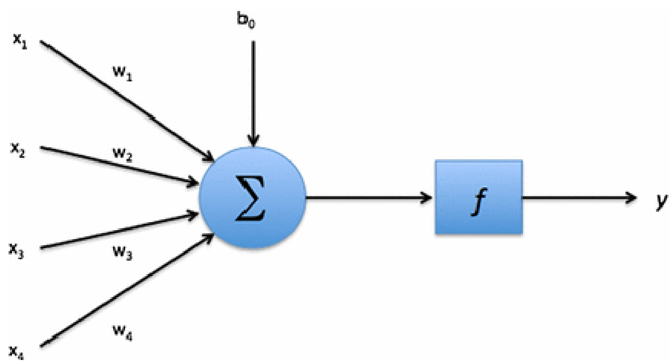


Figure 3.3: Neural Node

The input-output flow of a single node l of layer L is shown in Figure 3.3. Each input $Xi[l-1]$ to the node is multiplied by corresponding weight matrix Wi whose values are learned during training of the network. The products of this multiplication are then aggregated and a bias term bi is added to force minimal needed information to be gained from the node as shown. Finally, we apply an activation function, f(X), to the final value in order to force non-linearity mapping of the signal. Figure 3.4 below shows six different activation functions along with their corresponding equations and graphs.

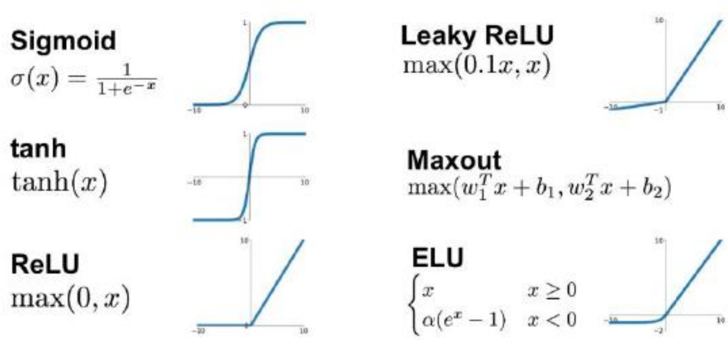


Figure 3.004: Activation Functions

The equations of calculating the response of node '**i**' of layer **'l'** during a single pass of forward propagation is:

$$Z_{il} = X_i * W_{il-1} + b_i$$

$$a_{il} = f(Z_{il})$$

Once a forward propagation pass is completed, predictions are made and the loss is calculated according to a defined loss function. The loss function is used to calculate how well the classifier is performing, as well as calculating the effect that needs to be applied to the weights to make better predictions, this is known as the error signal.

In order for the error signal to make updates to the weights of each node, it should be correctly passed in reverse way of the forward propagation. This process is known as the backward propagation algorithm which is the key strength of the ANNs. The back-propagation algorithm is based on the computational graph theorem, where differential equations are used to calculate the updates for the nodes. Figure 3.5 below shows the general formula of back-propagation equations.

$$\frac{\partial J_{\text{net}}}{\partial \mathbf{W}^{[l]}} = \frac{1}{m}\boldsymbol{\Delta}^{[l]}\left(\boldsymbol{A}^{[l-1]}\right)^{\mathbf{T}} + \frac{\lambda}{m}\mathbf{W}^{[l]}$$

$$\frac{\partial J}{\partial \boldsymbol{b}^{[l]}} = \frac{1}{m}\sum_{i=1}^{m}\Delta_{ji}^{[l]}$$

$$\boldsymbol{\Delta}^{[l]} = \left\{\boldsymbol{\nabla}_{\mathbf{A}^{[l]}}J\right\} \odot g^{[l]\prime}\left(\mathbf{Z}^{[l]}\right)$$

$$\boldsymbol{\nabla}_{\mathbf{A}^{[l]}}J = \mathbf{W}^{[l+1]^{T}}\boldsymbol{\Delta}^{[l+1]} \quad , \qquad l \neq L$$

Figure 3.511: Back Propagation Equations

3.2.2. Deep Learning Based Models

In this section, we discuss the technical background needed to understand the deep learning models we used. We give the full information relevant for our topics so that readers do not need further knowledge to understand how to re-implement the models.

3.2.2.1. Convolutional Neural Networks

A famous class of ANN is the convolution based neural networks. They use a step before the ANN known as convolution layer. Convolution is the processing of signal using some filter system where the filter performs a series of summations and multiplications over the signal in order to interpret it or map it into another form. Convolution over digital signals like 2D matrices using 2D filters can be simplified into shifting the filter over the matrix, and perform element-wise multiplication between the filter and its overlap with the matrix, resulting in a new form of the digital signal that can be useful to extract deeper, local, and more advanced information about the signal.

As shown in Figure 3.6 below, layers of a CNN (Convolutional Neural Network) can be classified into:

1. Convolutional layers: each layer consists of a filter of size f * f * c; where f is the filter size, and c is the channel size of the filter. The filter is applied to the input signal of size x * x * c, resulting in a completely new signal of size y * y * 1, where: $y = \frac{x-f\_2p}{s} + 1$[1]. Each layer is followed by a pooling layer which selects the most relevant output value of the resulting signal using maximum pooling, or taking their averages using average pooling.
2. Fully connected layers: which is an ANN of multiple layers that account for actual classification.

[1] x = input signal/image size, y = new signal/image size, f = filter size, p = padding of filter, s = stride step.

To understand how each process in the CNNs works, we need to go through them one at a time:

I. Convolution:

In the convolutional layer, a filter (kernel) slides to the right with a certain stride value till it parses the complete width. After scanning the row, it hops down to the next rows starting from the

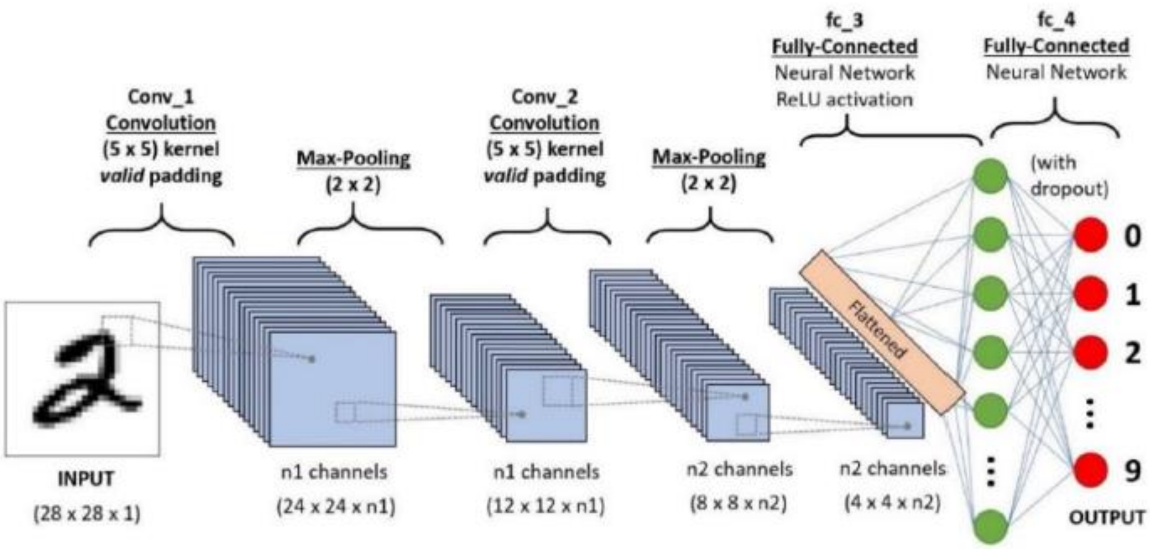


Figure 3.6: Convolutional Neural Network[2]

left of the image with the same stride value and repeats the process until the entire image is traversed. Each time convolving the window to create a single value in the output convolved feature.

In the case of images with multiple channels, the kernel has the same depth as that of the input image. This is known as volume convolution. Matrix multiplication is performed between k and In stack ([K1, I1]; [K2, I2]; [K3, I3]), where 'K' is the kernel and 'I' is the input image, and all the results are summed with the bias to give a squashed one-depth channel convoluted feature output.

The objective of the convolution operation is to extract the high-level features, such as edges, from the input image. Figure 3.7 below shows an example of a convolution operation where a kernel (filter) matrix that is applied to an input image matrix.

2 Kernel is another name for filters.

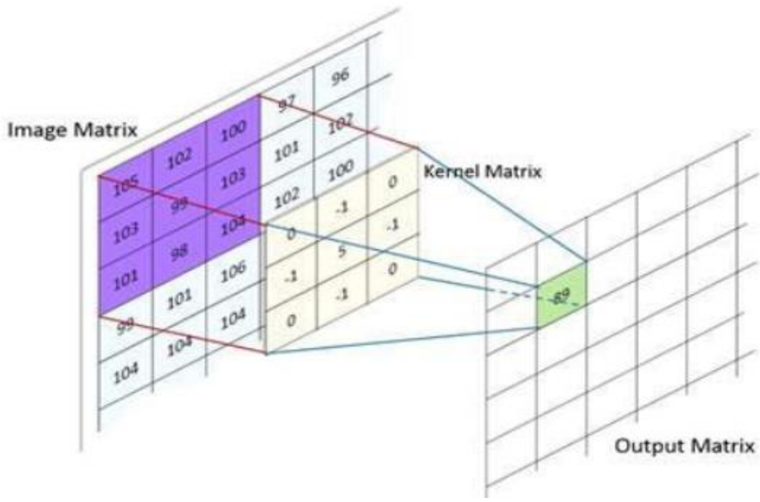


Figure 3.7: Convolution of an Image by a Kernel

II. Pooling:

After each convolution layer, a pooling layer is used which is responsible for reducing the spatial size of the convolved features. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training the model.

There are two ways to apply pooling of features: “Max Polling” and “Average Pooling”. Max pooling returns the maximum value from the portion of the image covered by the kernel. On the other hand, average pooling returns the average of all the values from the portion of the image covered by the kernel. Figure 3.8 below shows the difference between max pooling and average pooling operations. The choice of the pooling type is domain dependent where some problems tend themselves towards averaging, others may require maximum pooling. Max pooling is used more frequently than averaging.

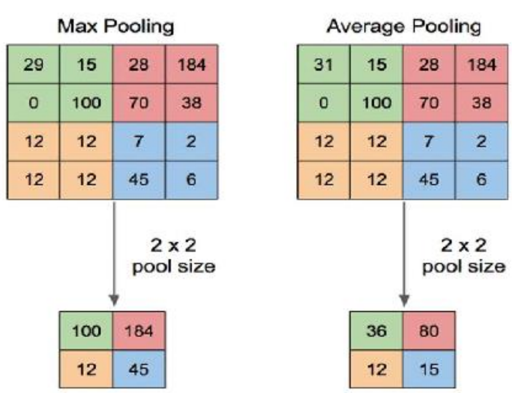
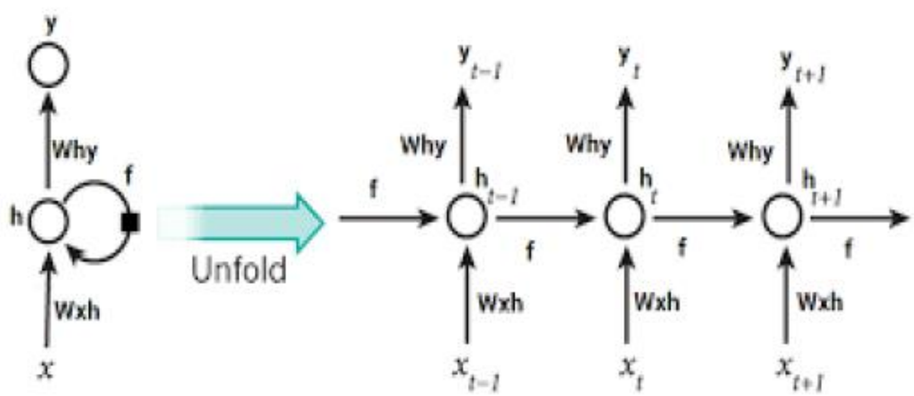


Figure 3.8: Max Pooling vs. Average Pooling

3.2.2.2. Recurrent Neural Network

RNN (Recurrent Neural Network) is a type of neural networks where the output from previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence or generating sequences of text like in the case of summarization, the previous words are required and hence there is a need to remember the previous words. Thus, the idea of RNNs came into existence, which solved this issue with the help of ‘hidden layers. The main and most important feature of RNN is the hidden state/layer, which remembers some information about a sequence of words which will be useful in our case or problem.

RNNs take the previous output or hidden states as inputs. The composite input at time (t) has some historical information about the happenings at time (T) where T < t. RNNs are useful, as their intermediate states can store information about past inputs for a time that is not fixed a priori. Figure 3.9[3] below illustrates how the RNN is modeled (unfolded). Figure 3.10 below shows the equations, for one-time step, of the RNN. As shown below in both figures, there are two steps: forward propagation where the inputs enter and move forward at each time step and



backward propagation through time where, the RNN weights are updated through time to make the network learn during the training phase.

Figure 3.9: RNN Network Structure

[3] Where ‘x_t’, ’x_{t-1}’ and ’x_{t+1}’ are input word vectors at the current, previous and next time step (t) respectively.’y_t’, ‘y_{t-1}’and ‘y_{t+1}’ are output word vectors at the current, previous and next time step (t) respectively.’h_t’, ’h_{t-1}’ and ‘h_{t+1}‘are the hidden state vectors at the current, previous and next time step (t) respectively. ‘W_{xh}’, ‘W_{hy}’ and ‘Why’ are the weights that are used with the input, output and hidden state vectors respectively. ’f’ is the activation function that is applied.

3.2.2.3. Long-Short Term Memory Cell

LSTM (Long-Short Term Memory Cell) cells network is a special kind of RNN, capable of learning long-term dependencies. Remembering information for long periods of time is essential and needed in many cases where the input source text is very long. Imposing a structure and learning long-term dependencies over this text can't be learned by the traditional RNN network and the need of something more advanced like the LSTM cells network is a must in such cases.

All RNNs networks have the form of a chain of repeating modules of neural network. In traditional RNNs, this repeating module will have a very simple structure, as shown before in the previous subsection. LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four unique ones, interacting in a very special way. Figure 3.10 below illustrates and shows the LSTM network structure; composed of multiple LSTM cells.

LSTMs are designed to remember the input state for a longer time than an RNN, hence allowing long sequences to be processed accurately. The next steps show how the LSTM process

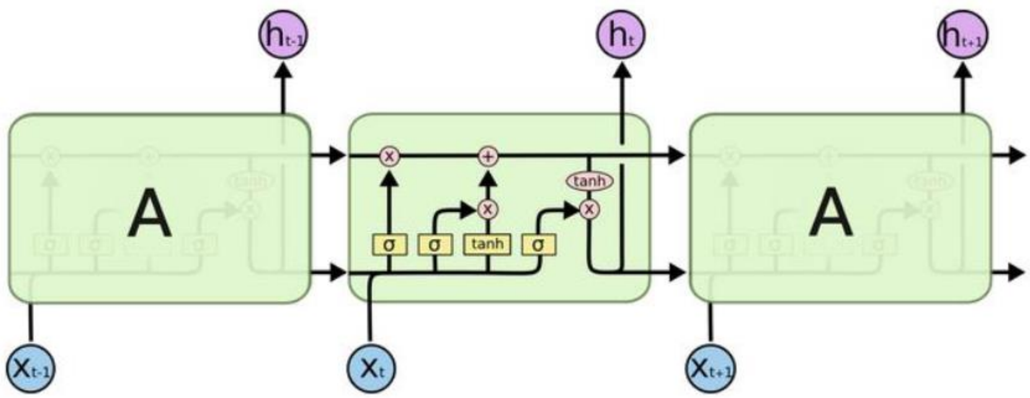


Figure 3.10: LSTM Network Structure

the input sequences and achieve a long-term dependency using three unique 'Gate Layers':

1. The LSTM decides what information that it is going to throw away from the cell state using the 'forget gate layer'.

2. The LSTM decides what new information that it is going to store in the candidate cell state using the 'input gate layer'.

3. The LSTM updates the old cell state into the new cell state by the help of the candidate cell state and both the 'forget gate layer' and 'input gate layer'.

4. Finally, the LSTM decides the output using the 'output gate' and the cell state.

3.2.2.4. Beam Search Mechanism

Beam Search is an approximate search strategy that tries to solve this in an efficient way. In its simplest representation, 'beam width' is the only tunable hyper-parameter for tweaking translation results. Beam width in-general decides the number of words to keep in-memory at each step to permute the possibilities. Beam Search can be described in the following steps:

1. Start decoding the target sequences using the decoder network, and collect the outputs that achieve high probabilities. Take only the top ('beam width' size) probabilities and keep them in-memory.

2. Hardwire the Y1 in the decoder network and try to predict the next probable word if Y1 is already occurred. The target is to maximize the probability of Y1 and Y2 occurring together. P (Y1, Y2 | X) = P (Y1| X) * P (Y2 | X, Y1) Here X = x1, x2 till Xn (all words in the input). 'X' are the input words, 'Y' are the words generated from the input and 'P' is probability.

3. Take top ('beam width' size) probable (Y1, Y2) pairs, that are kept in-memory in the previous step, and hardwire them in the decoder network and try to find conditional probability of Y3, i.e.: P (Y3 | X, Y1, Y2). Similar to previous two steps we again find top ('beam width' size) probable three-word sequences and so on. We keep this process till '<End>' is reached. If we set 'beam width' size = 1 then the technique is the same as 'Greedy Search' which will only take the current output sequence token that has the highest soft-max probability without taking care of the below tokens that has less probability. Larger the 'beam width' size, good chances for better output sequences, but it would consume more resources and computation power. Smaller the 'beam width' size, not so good results, but much faster and memory efficient.

3.3. Comparative Study of Previous Work

In this section we give background information needed for understanding how our system is made, and then we review academic papers and their different methodologies that have been published on the subject of “Handwritten Text Recognition”. Various methods have been discussed, with many different specializations.

3.3.1. Background on Handwritten Text Recognition

There are two approaches for text recognition; feature extraction, classification, and pattern matching.

3.3.1.1. Types of Text Recognition

Feature Extraction, and Classification

Feature extraction breaks down or decomposes the glyphs into features such as lines, closed loops, line direction, and line intersections. It then uses these features to find the best match or the nearest neighbor among its various stored glyphs.

Classification is a technique where we categorize data into a given number of classes. The main goal of a classification problem is to identify the category/class to which a new data will fall under.

Pattern Matching

Pattern matching works by isolating a character image, called a glyph, and comparing it with a similarly stored glyph. Pattern recognition works only if the stored glyph has a similar font and scale to the input glyph. This method works well with scanned images of documents that have been typed in a known font.

3.3.1.2. Methods of Feature Extraction

This section gives a brief description of the methods and algorithms used for Feature Extraction.

3.3.1.2.1. Zoning

The frame containing the character is divided into several overlapping or non-overlapping zones and the densities of object pixels in each zone are calculated. Density is calculated by finding the number of object pixels in each zone and dividing it by total number of pixels.

3.3.1.2.2. Projection Histogram Features

Projection histograms count the number of pixels in specified direction [4]. There are three types of projection histograms: Horizontal, Vertical, Left diagonal, and right diagonal. These projection histograms for a 3*3 pattern are depicted in figure 3.11.

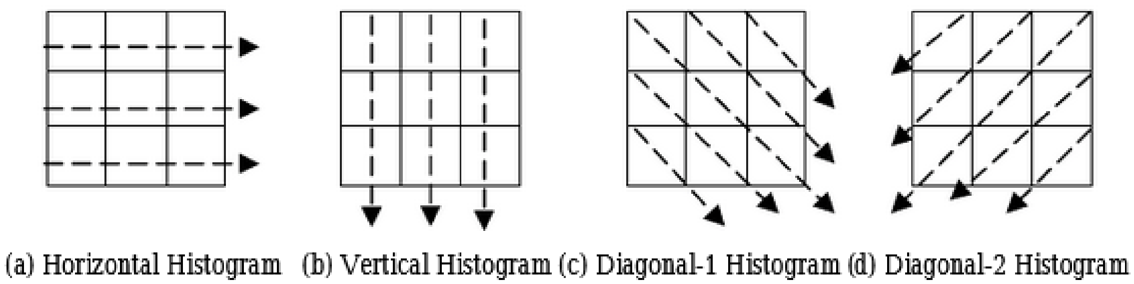


*Figure 3.11: Evaluation of 4 types of Projection Histograms on 3*3 patterns*

3.3.1.2.3. Distance Profile Feature

Profile counts the number of pixels (distance) from bounding box of character image to outer edge of character. In this approach, profiles of four sides left, right, top and bottom were used [4].

3.3.1.2.4. Background Directional Distribution (BDD) Features

To calculate directional distribution values of background pixels for each foreground pixel, we have used the masks for each direction shown in figure 3.12. The pixel at center 'X' is foreground pixel under consideration to calculate directional distribution values of background. The weight for each direction is computed by using specific mask in particular direction depicting cumulative fractions of background pixels in particular direction [4].

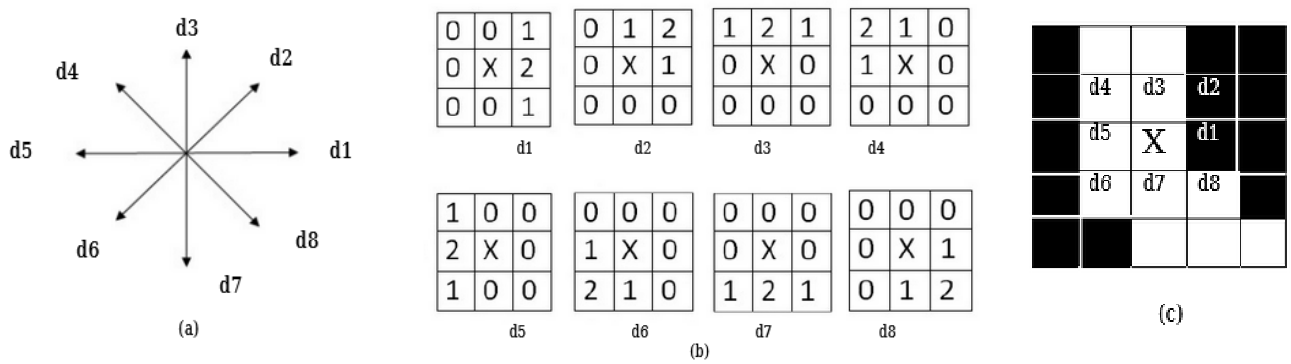


Figure 3.12: (a) 8 directions used to compute directional distribution, (b) Masks used to compute directional distribution in different directions.

3.3.1.3. Methods of Classification

Classification determines the region of feature space in which an unknown pattern falls.

3.3.1.3.1. K-Nearest Neighbor

In k-nearest neighbor algorithm (k-NN) [4], [5] is a method for classifying objects based on closest training examples in the feature space. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor.

Generally we calculate the Euclidean distance between the test point and all the reference points in order to find K nearest neighbors, and then arrange the distances in ascending order and take the reference points corresponding to the k smallest Euclidean distances. A test sample is then attributed the same class label as the label of the majority of its K nearest (reference) neighbors.

3.3.1.3.2. SVM (Support Vector Machines)

Support vector machines (SVM) [4], [5] are a group of supervised learning methods that can be applied to classification. A classification task usually involves separating data into training and testing sets. The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes. The standard SVM classifier takes the set of input data and predicts to classify them in one of the only two distinct classes. SVM classifier is trained by a given set of training data and a model is prepared to classify test data based upon this model. For multiclass classification problem, we decompose multiclass problem into multiple binary class problems, and we design suitable combined multiple binary SVM classifiers. Different types of kernel functions of SVM: Linear kernel, Polynomial kernel, Gaussian Radial Basis Function (RBF) and Sigmoid (hyperbolic tangent).

3.3.1.3.3. Probabilistic Neural Network (PNN) Classifier

A probabilistic neural network (PNN) [4] is a classifier which maps any input pattern to a number of classifications. If the probability density function (pdf) of each of the populations is known, then an unknown, X, belongs to class 'i' if:

$$fi(X) > fj(X), all\, j \neq i$$

fk is the pdf for class k.

3.3.1.4. Methods of Pattern Recognition

Matrix matching involves comparing an image to a stored glyph on a pixel-by-pixel basis; it is also known as "pattern matching", "pattern recognition", or "image correlation".

3.3.1.4.1. The Dictionary Approach

This approach improves the character results obtained by the actual OCR processing. This approach uses a given alphabet for word matching, which makes this approach dependent, and so we assume that the language used in the document is known.

3.3.1.4.2. The Independent Approach

IF the language used in the document is unknown or if the purpose of passing the document through the OCR is to find a new language, then this approach is the perfect fit. Specially that this approach creates a new dictionary for the document, which can correct a document written in an unknown language.

3.3.2. Deep learning approaches for Text Recognition

The availability and ease of access of huge amounts of text data on the web presents both an opportunity as well as a challenge. Increased accessibility of data has led to the information overload problem. An important task in the domain of natural language understanding is text recognition, and one of the most used methods for text recognition and OCR is using deep learning.

Due to the success of modern deep neural network architectures, it is envisaged that state-of-the-art handwriting recognition systems will be either hybrid systems (deep networks with some segmentation and features extraction) or pure neural recognizers with deep architectures [36]. There are many deep learning systems and techniques used for handwriting recognition. Some of these systems and architectures are presented in the following subsections.

3.3.2.1. Hidden Markov Models (HMM) and Hybrid Approaches

HMMs have been developed in the 1960s and have since been successfully applied to sequential data such as speech or handwritten text. A HMM consists of observable variables:

$xt \in X$ and internal (hidden) variables: $zt \in Z$, as we can see in figure 3.13.

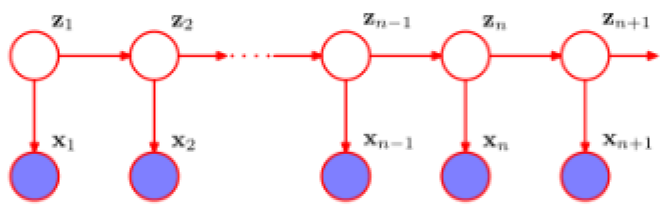


Figure 3.13: Hidden Markov Model

A HMM model is a generative model that means it can be used to generate new data. When using HMMs for HTR, a separate HMM is created for each possible character. Multiple character HMMs are used in parallel to account for all possible characters at this location. A series of such parallel character models is used to recognize a complete line of text. An illustration for such a HMM to recognize a text line can be seen in Figure 3.14.

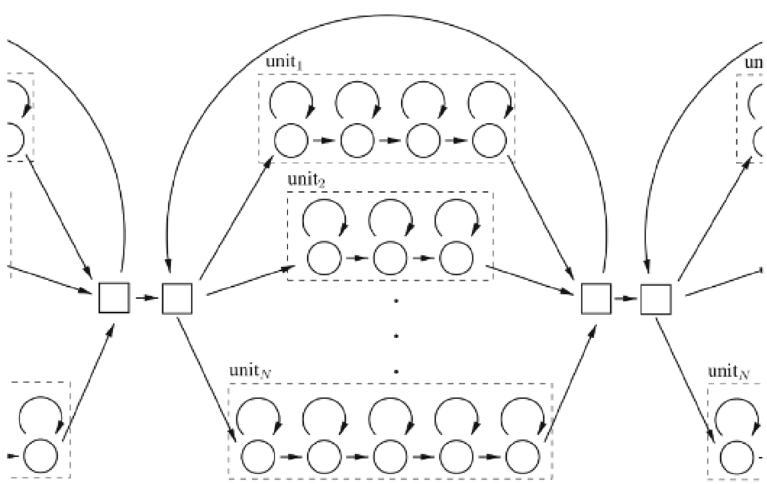


Figure 3.14: Illustration for such a HMM

3.3.2.2. Convolutional and Recurrent Neural Networks

Figure 3.15 shows the main components of an end-to-end trainable ANN for handwritten recognition proposed by Shi et al [6]. The input to the ANN is a gray-value image containing text. A stack of convolutional layers maps the input image onto feature maps. The output of the final layer of the CNN can be regarded as a sequence of length T with F features. Information is then propagated along this sequence with a stack of RNNs. The RNNs map the sequence to another sequence of same length T, assigning probabilities to each of the C different classes. Finding the most probable labeling in this C × T sized matrix is called decoding and is done with a Connectionist Temporal Classification (CTC) output layer. While training, the CTC loss is used to calculate a loss value for a training batch which is then back propagated to the output layer of the RNN. The integration of a language model (LM) can be done directly in the CTC decoding algorithm or as a post processing step which modifies the predicted labeling.

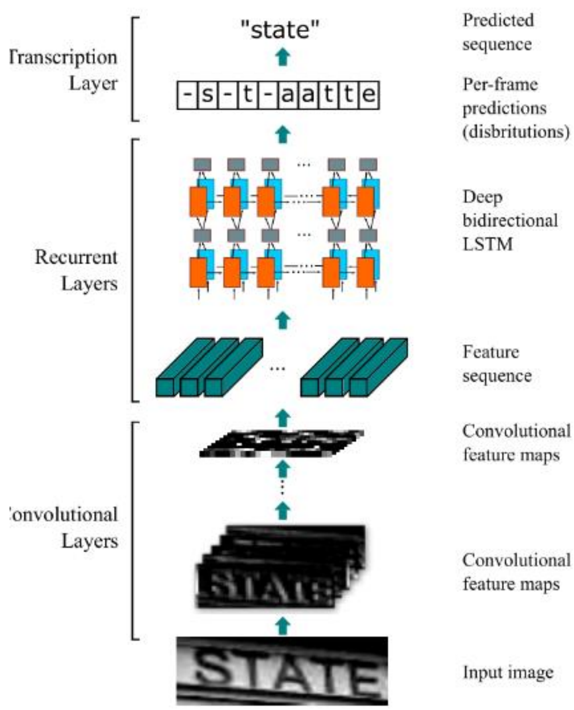


Figure 3.15: The network architecture called CRNN

3.3.2.3. Long-Short Term Memory (LSTM)

It is also a sequential model starts with embedding layer then LSTM layer and a dense layer with soft-max activation. LSTM layer helps us to encode all information of the text in the final output of recurrent neural network before running the feedforward network for classification. The LSTM layer with 100 memory units.

3.3.2.4. CONNECTIONIST TEMPORAL CLASSIFICATION (CTC)

In the context of RNN, the use of Connectionist Temporal Classification (CTC) enables recognition without prior segmentation [17]. This has enabled the use of neural network classifiers for offline recognition of handwritten texts, which has become more popular in recent years [17], [24], [25]. The segmentation process for a text, printed or handwritten, is a highly error-prone task. The Connectionist Temporal Classification (CTC) layer can transcribe the data without prior segmentation. Graves et al. [11] introduced the approach of CTC which was initially designed to recognize speech [14], and then the idea was extended to recognize handwriting [17]. The CTC layer predicts the transcription for the test images in the recognition phase. The out transcriptions are matched with the ground truth of the line image using Levenshtein edit distance.

3.3.2.5. TOKEN PASSING DECODER

Graves et al. [43] introduced the token passing algorithm which restricts its output to a dictionary. The token passing decoder solves the problem of word decoding, i.e. the most likely word in the input sequence, where the word output is limited by vocabulary. For single words, the token passing decoder expects a sequence of letter probabilities of length t as provided by the neural network, along with a word w as a sequence of ASCII characters, and returns a corresponding score, i.e. the probability that the input to the neural network was indeed the given word [14]. This means that, the probability $p\ (l \mid x)\ =\ \sum \beta(\pi) = l\ p(\pi|x)$ could be approximated by:

$$p\ (w \mid x)\ =\ max\pi, B(\pi) = wp(\pi)$$

3.3.2.5. WORD BEAM SEARCH (WBS) DECODER

Word Beam Search (WBS) decoding is an algorithm used to decode the output matrix of the CTC layer. The WBS decoder is placed just following the CTC layers for output decoding. The main advantages of the WBS decoder [44] over token passing decoder are:

- It is faster than token passing.
- It allows an arbitrary number of non-word characters between words (numbers, punctuation marks).
- Words constrained by dictionary.

Scheidl et al. [44] presented a WBS decoder which is a modification of the Vanilla Beam Search (VBS) decoding algorithm [44] that has the following characteristics:

- Words are limited to words in the dictionary.
- Any non-word character number between words is permitted.
- Optionally, we can integrate an LM word-level bigram.
- A better runtime (in terms of time-complexity and real-time on a computer) than token passing.

The iterative beam search decoding creates and scores the text (beams). The beam search decoding algorithm allows arbitrary character strings, which are required to decode numbers and punctuation marks while token-passing limits its output to words in the dictionary to avoid spelling mistakes. Of course, all words to be recognized in the dictionary must be included.

3.4. Implemented Approach

In this section, we show the chosen approaches and methods for each module; justifying the reasons behind our choices.

3.4.1. WHOLE-WORD Modeling (HOLISTIC RECOGNITION)

Arabic words or PAWs were used by some researchers as modeling units long ago. There are many problems with this approach. One such problem is that it usually leads to a massive model set as each word will be represented by a separate class. There may be hundreds or thousands or more words in a lexicon. This means that more training data are required as each word should have enough samples in the training set to train model parameters adequately. It may only be appropriate for text recognition tasks with a very small lexicon size [16], [21].

3.4.2. Using Characters as Modeling Units

In this modeling option, the basic characters (28 characters, ten digits, and seven ligatures) are used as modeling units. However, because Arabic has many position-dependent models, modeling English/Arabic handwriting, using characters as models are not the best way, although it was used in the early days of Arabic text recognition. Many Arabic character shapes look quite different visually and mapping them to one model may not be the best option.

3.4.3. Using the Character Shapes as Modeling Units

The use of a character shapes as distinct models is the most successful and widely used approach [25]. Although this approach generally works well, it still has a problem. The number of modeling units rises four times from 28 to about 100, with the exclusion of numbers and special characters. This creates a big model set with a large recognizer. Moreover, a wide range of modeling units requires extensive training data to ensure that each model is properly trained.

3.4.4. Time Distributed Convolution Recurrent Neural Network Approach

As we have discussed in the previous section, the most promising approach, will be discussed in-details in the next chapter, was the CNN based model. Its key strength is that it addresses the problems of all other approaches.

In all the papers we have reviewed, we found similar techniques that have been used, and when it comes to deep learning the CNN is very popular. The accuracy of the results depends on many factors: the character count, how noisy the original image is, the dataset that is used, and most importantly, choosing the right technique for the desired language.

It caught our attention a scientific paper that used more than one technology to implement the OCR, and compared the accuracies. It was found that CNN when compared to SVMs, and Nearest Neighbor, it can perform far better than other techniques, and that is due to the convolutions, and multiple filters [3].

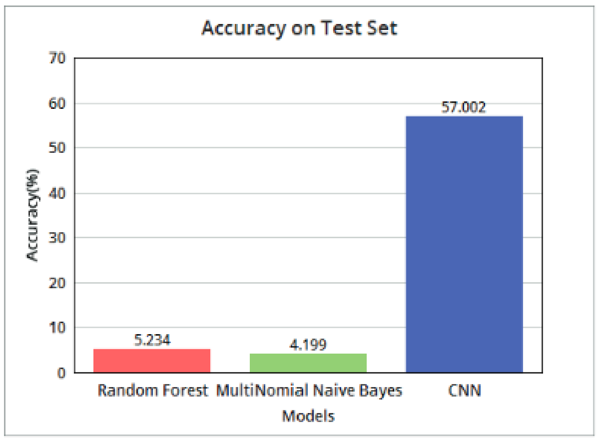


Figure 3.16: Accuracy of Different Classifiers

SCAN MED

Chapter 4: System Design & Architecture

4. System Design and Architecture

The third phase of building our solution is to establish initial system design, and agree on the architectural paradigms and patterns we follow. Since we use Agile Process for development and maintenance of the product, this architecture should be flexible enough to allow for changes, and be well planned to minimize the dramatic changes that could be raised. We discuss the details of the application architecture in this chapter, giving reasonable justifications for each decision we agreed on starting by giving an overview for our plan, as well as assumptions made before designing process. We follow that by detailed design for the complete solution, and break down each module to its finest detail.

4.1. Overview and Assumptions

The model is developed using tensorflow keras framework, as it was GPU optimized, and easy to handle. When trying to develop an individual neural network with feed forwarding, back propagation and weight adjustment in word embedding, it was too slow and near impossible to be developed.

4.2. System Architecture

This section explains the system architecture of SCAN-MED. The block diagram and the overall design will be discussed with the implementation details of each module, including its exact inputs and outputs.

4.2.1. Modules Overall Block Diagram

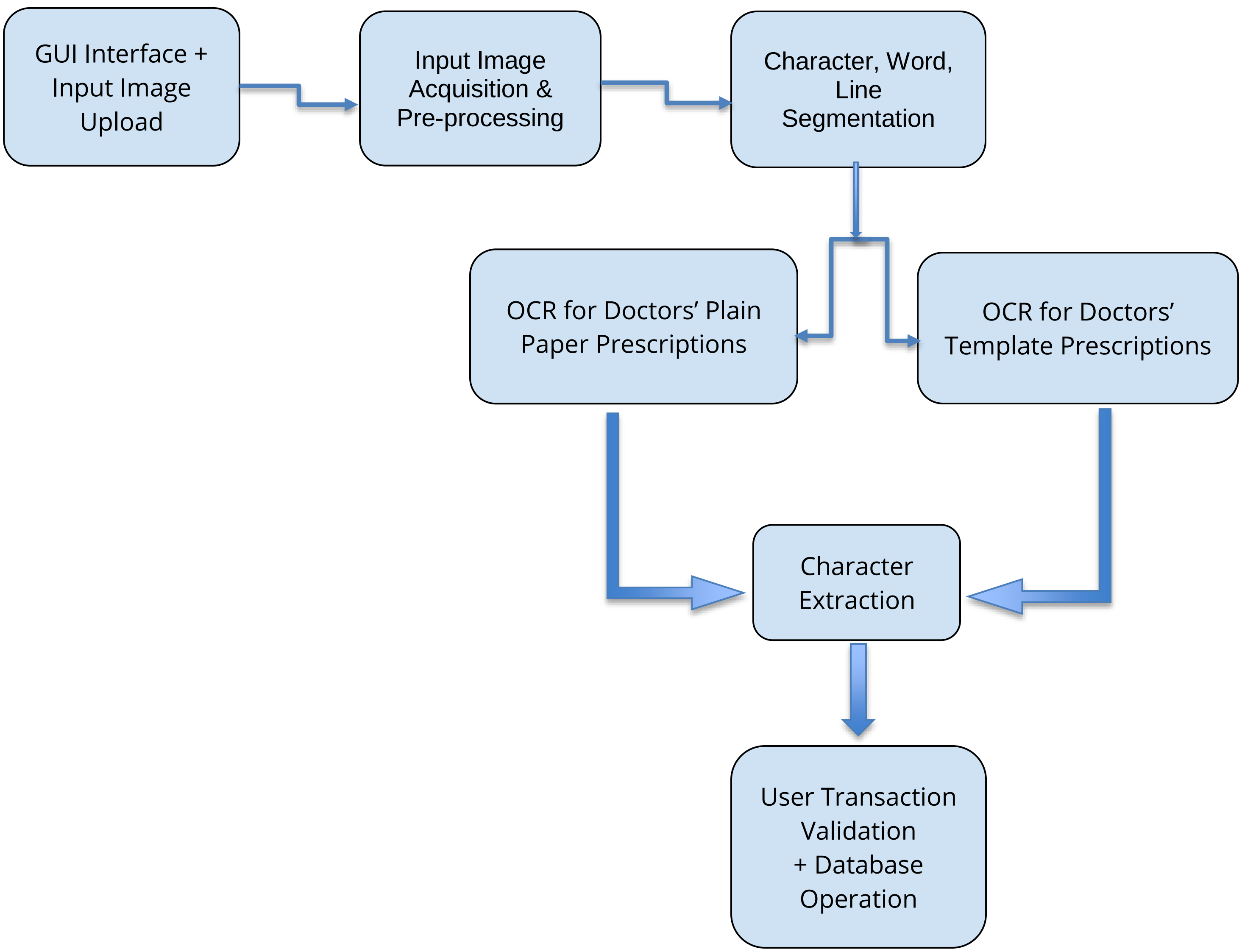


Figure 4.1: Block Diagram

The system consists of 6 main modules, and will be divided on us equally.

1st Module: Graphical User Interface (GUI):

- Responsible for signing in and out, check if user want to input a medical form or a doctor's prescription, and check patient's status, and more.

2nd Module: Image Acquisition and Preprocessing

- Responsible for acquiring images from the user, enhancing the image, correct it from any rotation if need it, remove any noise, and enhance the brightness.

3rd Module: Image Segmentation:

- We will extract our desired characters and save each of them in an image or in an array, and in this module, it will be different in case it's a medical form or a doctors' prescription.

4th Module: Character Recognition for Doctors' Prescriptions:

- We will extract our matched characters and put the input in a text file.

5th Module: Character Recognition for Patients' Medical Form:

- We will extract our matched characters and put the output in a text file.

6th Module: Database:

- We will insert and update the organized text files into a database.

We will discuss in the next subsections the clear details of each module, and explain the work flow and functionalities in a proper way to make the reader better understand the implementation of our system.

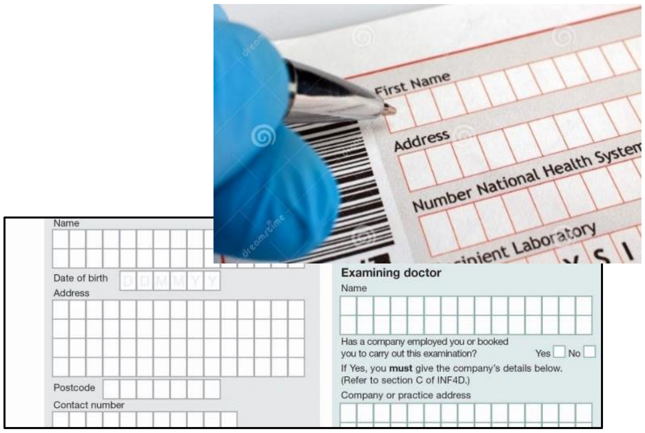


Figure 4.2: An input can be a predefined template for Prescription"

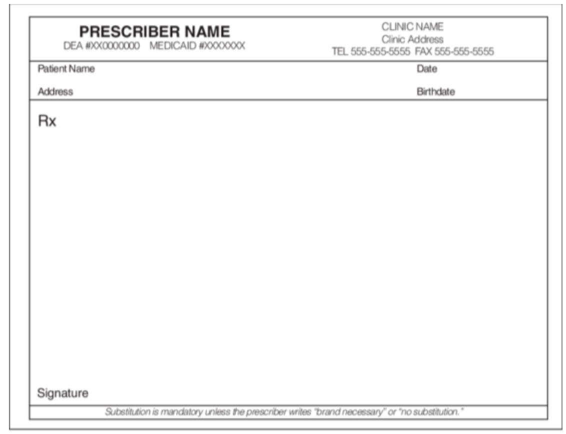


Figure 4.3: Input can be Doctor's Prescription as a plain Paper

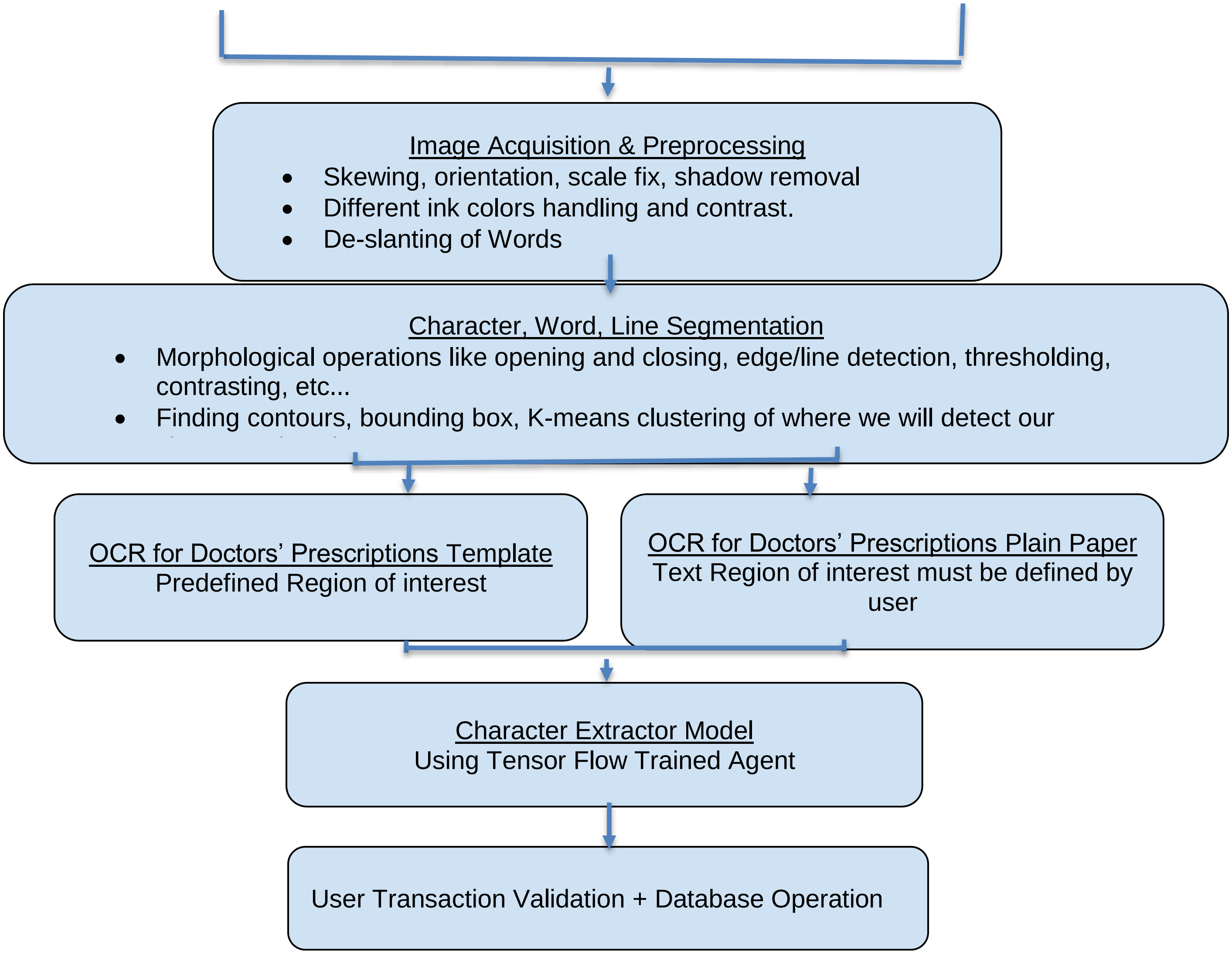


Figure 4.4: Detailed Block Diagram

4.3. Module 1: GUI

In this module, SCAN-MED will have a fancy interactive interface for the user in which will he will be prompted to multiple options, and features before the pre-processing phase. As our method concern is to save time and effort in an automated way, we have developed a simple GUI using C# and dot net windows applications with. It consists of some buttons for program functionalities like: a button to (Choose a Prescription) allows the user to open a directory-pane or open file-explorer which asks the user to choose a sample image file of the prescription and a Start Processing button for which the classification work is executed in the back-end.

4.3.1. GUI Functional Description

A Users is prompted to choose to input (upload) a scanned doctors' prescription whether it is a medical form - "a template" – or a "Plain Paper" prescription with medicine and dosage to be extracted. Both can be in English or Arabic, thus the user is prompted to choose at the beginning the language.

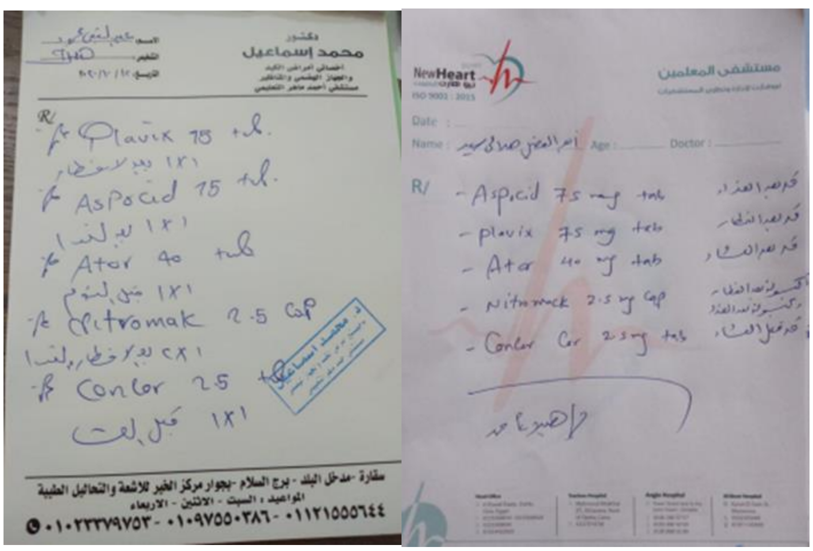
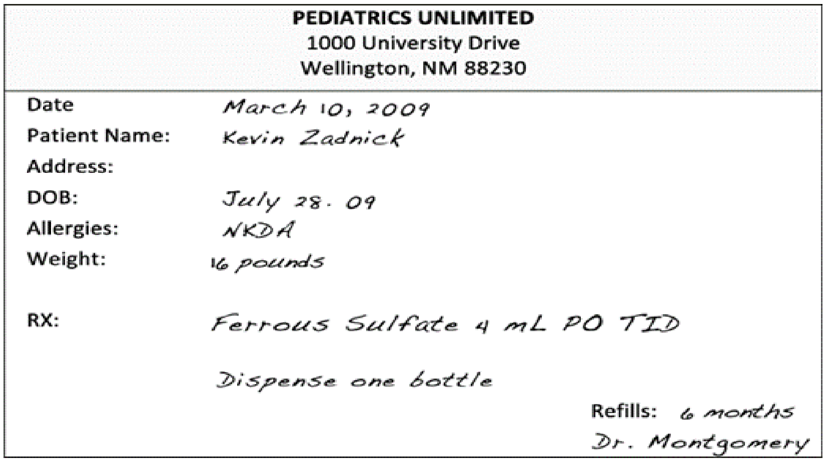


Figure 4.5: Example on plain paper



PEDIATRICS UNLIMITED
1000 University Drive
Wellington, NM 88230

Date	March 10, 2009
Patient Name:	Kevin Zadnick
Address:	
DOB:	July 28. 09
Allergies:	NKDA
Weight:	16 pounds
RX:	Ferrous Sulfate 4 mL PO TID
	Dispense one bottle

Refills: 6 months
Dr. Montgomery

Figure 4.6: Example on template

A plain paper prescription is a document where there is no expected arrangement or pattern of hand-written text. The area of interest – the one which contains the prescription writing and dosage can be randomly anywhere on the paper. In that case, the user is prompted to highlight the area of concern manually, where a preview of the cropped area will be displayed on the right side to facilitate the user's process. The cropping process can be re-taken as per the user preferences and there are no limits on trials. The only limitation is the gentle assumption that the user crops a reasonable size of the document and the cropped area will be containing some hand-writings to extract, otherwise, the user will be prompted to apply for another trial.

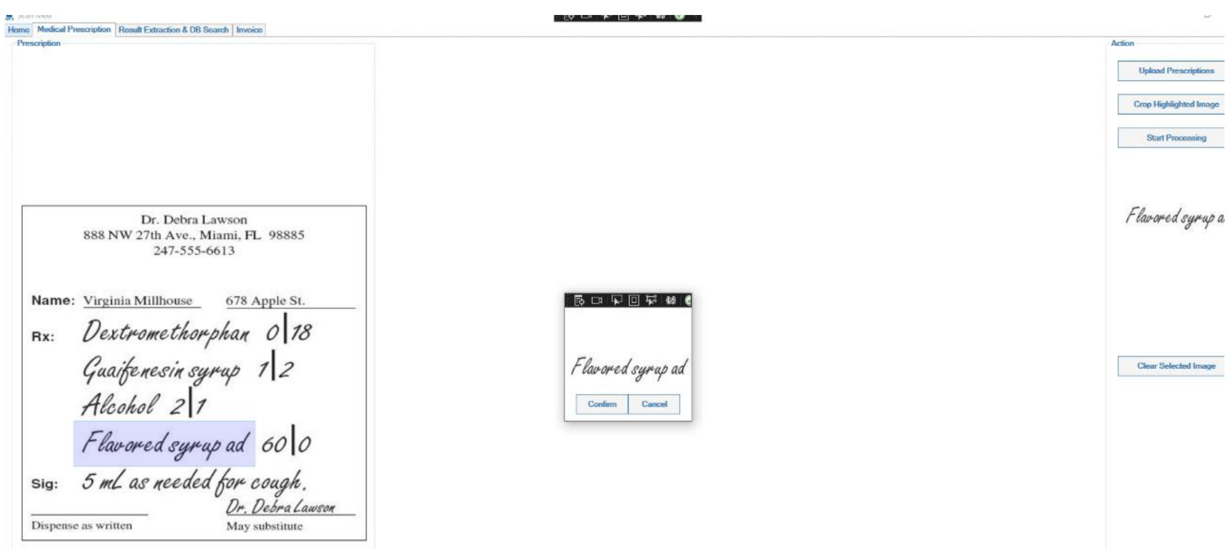


Figure 4.713: Example 1 on input plain paper

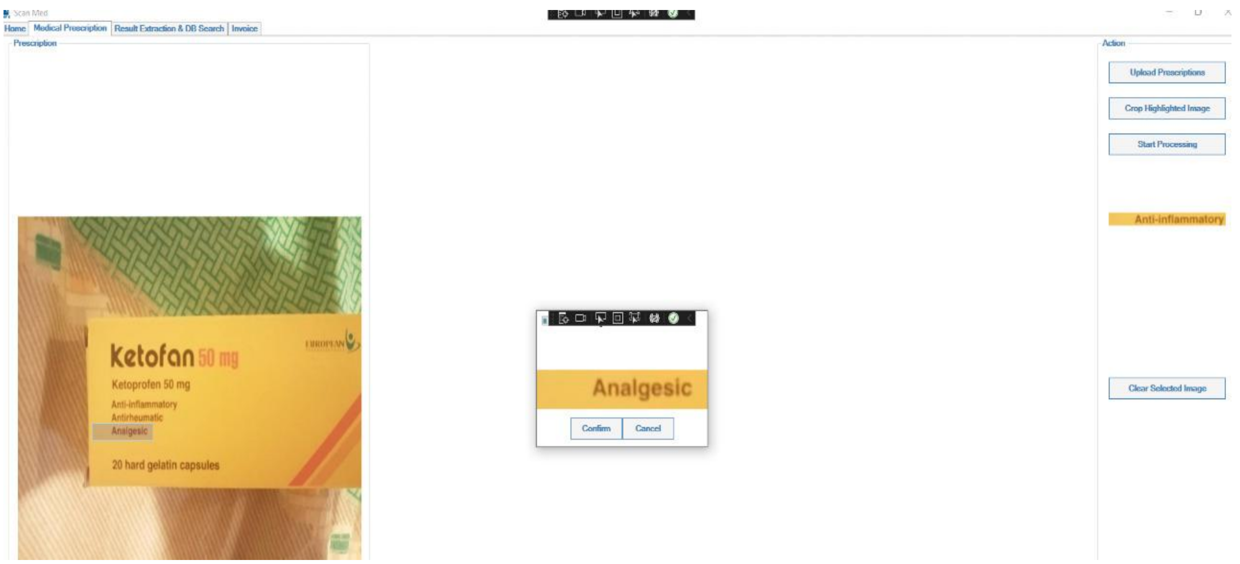


Figure 4.8: Example 2 on input plain paper

A template is a pre-determined document format layout consisting of a header, a body and a footer. The area of concern in this case will be pre-defined in the document body where it is expected to find the info of the medicine and dosage for extraction.

After the confirmation of user's input, the user shall press on start processing to advance into the next phases automatically and the next user interaction needed is the output validation for the database operation.

4.4. Module 2: Image Acquisition and Preprocessing

After we have acquired the desired image from the user, and make sure it is cropped is meaningful, the image shall pass through preprocessing phase in order to get enhanced and ready for the following modules with minimized errors in results.

The Tensorflow model was trained on the IAM-dataset. The model not only trained to learn how to read a hand-written text, but it also learns how the dataset samples are expected look like. Throughout the IAM dataset, word images have generally high contrast and are tightly cropped (well separated). An input image compromising those criteria will also compromise the accuracy of the extraction module. For this reason, a pre-processing module was implemented in this project for better performance.

4.4.1. Preprocessing Module Functional Description

Our proposed preprocessing flow diagram as follows in figure 4.9 below:

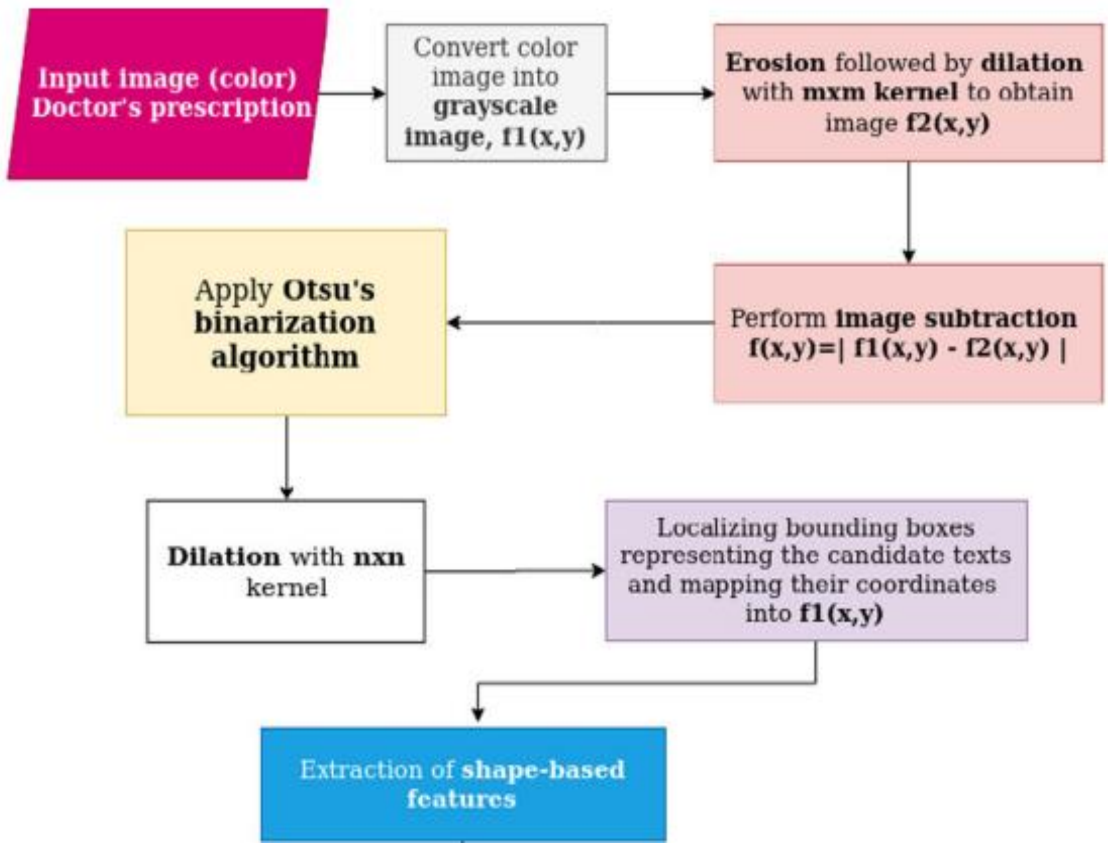


Figure 4.9: proposed preprocessing flow diagram

The image is first converted into single channel grayscale image f1(x, y). The text pixels and character pixels have a very low value, whereas the pixel value of the background is very high. Comparing from different images about the width of the characters, a 11×11 zero-valued square sized morphological structuring element b (x, y) is taken as kernel.

Using this kernel, erosion followed by dilation is operated on the grayscale image f1(x, y). Due to the shape and size of the characters and their region stability throughout their position, by using the square morphological kernel there is no risk about modifying their original shape. The characters will remain in lower pixel value after this operation, and both the printed and handwritten texts will be characterized with lower-valued pixels.

The next step was determining the difference of the corresponding pixel values of the grayscale image f1(x, y) and the resultant image f2(x, y), a new image is obtained as output which contains lower-valued pixels for the text part, and lower valued pixels for the background part because the background has higher pixel values for both of the two operand parts, As shown by Equation Formula:

$$f2(x, y) = (((\overline{f1 \oplus b})(x, y)) \oplus b)(x, y)$$

Following, we looked for higher-valued pixels for the patches because they appear on the second part only. And finally lower-valued pixels are observed for the text parts inside the patches as the characters of the texts possess very low pixel value in both of the images in operand parts, As shown by Equation Formula:

$$f(x, y) = abs(f1(x, y) - f2(x, y))$$

Consequently, also the lines are removed from the text due to these morphological operations.

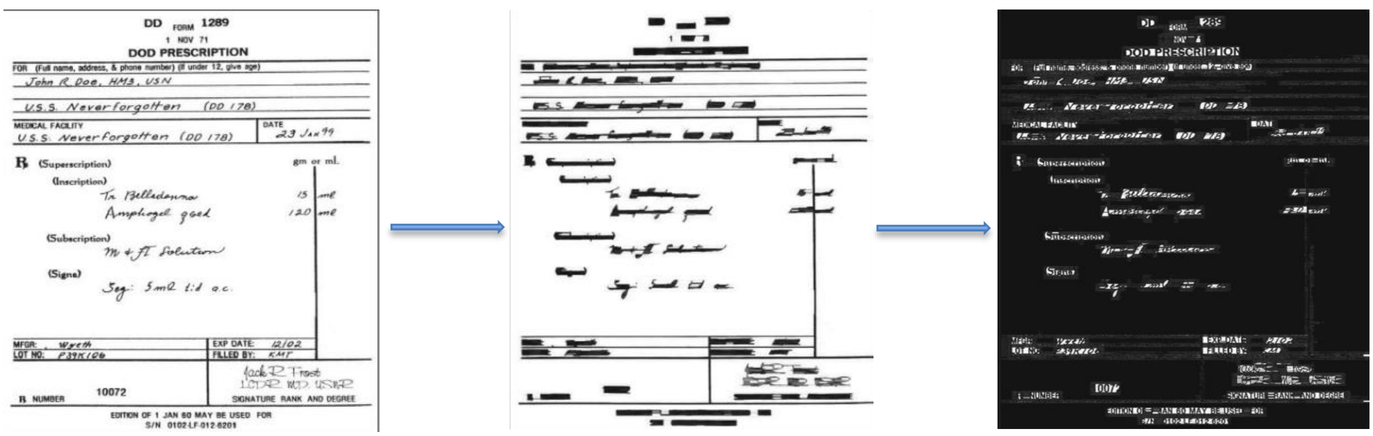
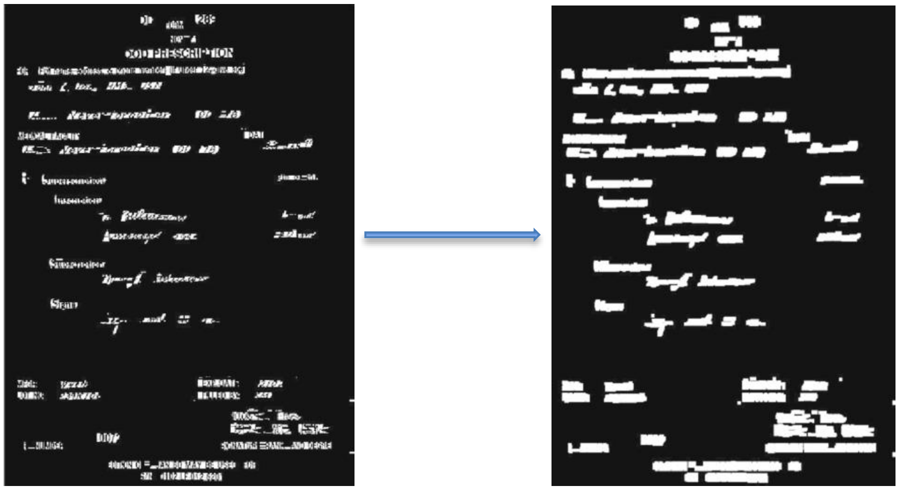


Figure 4.10: Diagram of morphological operation flow

The next step in pre-processing module is Otsu’s binarization. Otsu’s thresholding is performed on the output image, followed by dilation operation with a square structuring element c(x, y) having size 5×5. It is done in order to reshape the regions to get them in their actual size and also to remove the unnecessary dots and points scattered throughout the image. The resultant image after performing Otsu’s binarization is shown below as well as the output image produced after applying dilation operation.

The process can be expressed by the equation formula:

$$r(x, y) = (f[Otsu] \oplus c)(x, y)$$



Now, the bounded box of each of the regions is determined and these parts are implicitly extracted (cropped) out from the original image, and five shape-based parameters are calculated from that cropped binarized image. These features are defined as follows.

- Height (hp)
- Width (wp)
- Aspect Ratio (ar)
- Height Text Ratio (HTR):
- Transition Count Ratio (TCR):

$$\frac{\sum_{i=1}^{hp} \frac{\textit{Total number of black pixels in } i^{th} \textit{ column}}{hp}}{wp}$$

$$\frac{\sum_{i=1}^{wp} \frac{\textit{Total number of transition of pixels in } i^{th} \textit{ row}}{wp}}{hp}$$

Where, Transition means ‘change in pixel value from white to black and vice versa’ as a parameter measuring modifications throughout the morphological process.

The last method of enhancing the input image was De-Slanting (which will be thoroughly implemented as future work to our preprocessing module of our project), “The slant is the angle between the vertical direction and the direction of the strokes that, in an ideal model of handwriting, are supposed to be vertical”.

An illustration of slant and slope is shown:

Figure 4.11: illustration of slant and slope

An example for an input image and the de-slanted output image is shown:

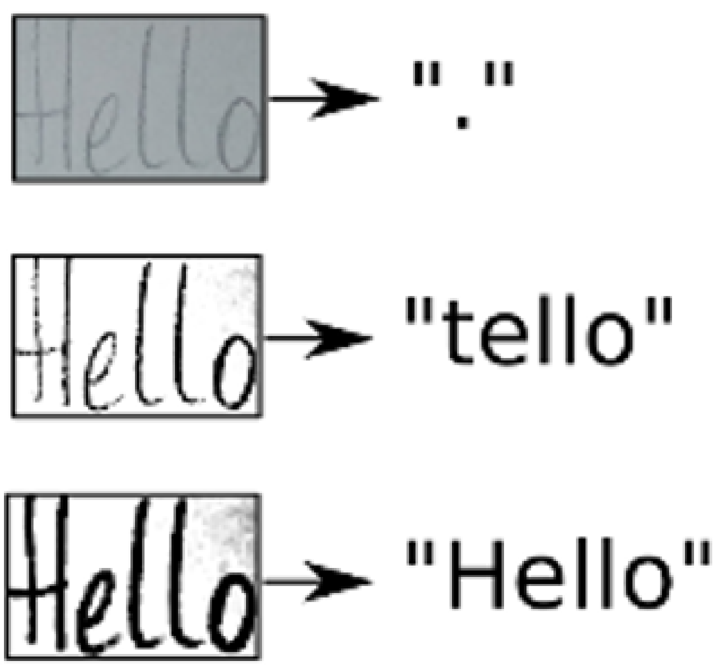
Figure 4.12: Example for an input image and the de-slanted output image

De-slanting is the process of transforming the handwritten text so that the slant angle gets minimized. The de-slanting technique proposed by builds on the hypotheses that an image containing text is de-slanted if the number of image columns with continuous strokes is maximal.

The Preprocessing Module showed remarkably better results than if we neglected it and increases the accuracy of word detection and characters recognition.

As noticed, the word “Hello” was scanned by a mobile phone camera, and tested for extraction, better results were not shown unless improving the image input quality by passing through a preprocessing module first.

An illustration is shown below:



The following figures are also some samples tested on SCAN-MED:

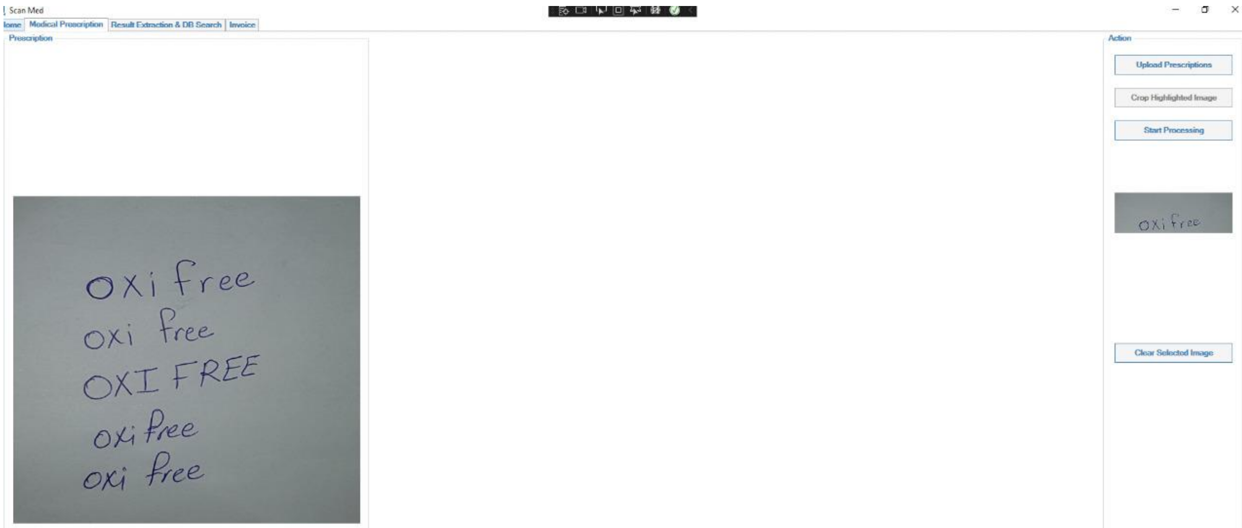


Figure 4.13: Sample tested on SCAN-MED

The result -after training the agent- showed inaccuracy to some extent:

Figure 4.14: Result of the sample tested on SCAN-MED

However, after Applying the preprocessing for enhancement of the input compared to the dataset the agent was familiar with:

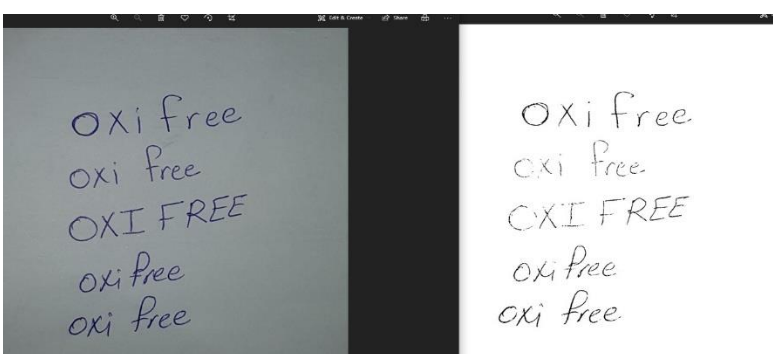


Figure 4.15: Same sample but with preprocessing

The Results have shown a greater accuracy and matching:

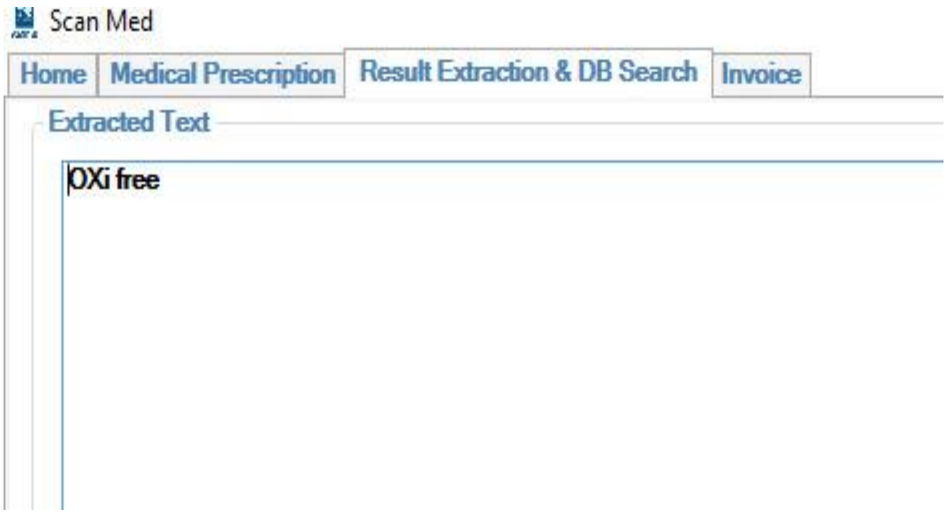


Figure 4.16: Output of sample with preprocessing on SCAN-MED

4.5. Module 3: Character Segmentation

The output of this stage will go directly to the last stage which is optical character recognition, the output here should be just letters and numbers that can be handled separately.

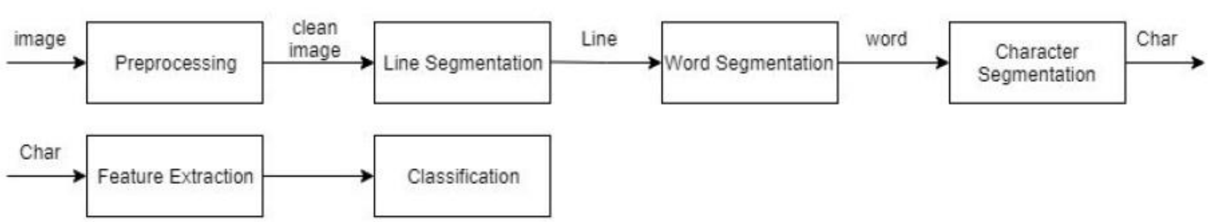


Figure 4.17: Character Segmentation block diagram

4.5.1. Functional Description

Line Segmentation: we make a horizontal projection and detect the gaps and from the gaps we segment the lines into partitions.

To get the approximated lines count we need to calculate the single line width after that the count is easily calculated by dividing the width of the segment by the width of the single line.

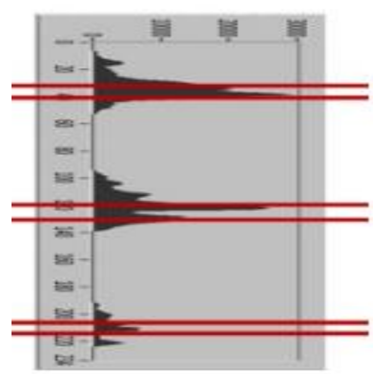
To estimate the line width, horizontal projection method is applied to the segment to find the global maximum peak and its location. The location of the global maximum peak performs the baseline for single line in the segment; the baseline is then marked by marking the pixels pass through it with white color.

Line segmentation is done using Image Axis Profile method that calculates the horizontal axis profile for the binarized text image. The horizontal axis profile matrix Ij is calculated by summing pixels values$P(i,j)$ along the X-axis for each y value:

$$I_j = \sum_{i=0}^{i=n} \left(255 - P(i,j)\right)$$

This profile has information about the text lines that are indicated by the regions with the black intensities. The blank lines appear as a drop in the black intensities. The text lines can be extracted by comparing the profile with a pre-defined threshold. This is done using the horizontal axis profile on two stages. The first one is to locate each connected group of dark regions in the profile. The other one is to decide which dark region(s) can be considered as a separate line.

Line Segmentation, thus, involves the detection of position of the local maxima, the base-line is the space around the maximum value in the horizontal profile, so to avoid false alarms and remove noise, Gaussian filter (low pass filter) is used to smooth the image.



Word Segmentation: the next step and by the same analogy, we segment line into separate words by using Vertical projection and deciding on the values between characters and blanks (gaps) using a threshold and ratio of about 1/3 of the connected strokes.

The text line is segmented, beginning from the left side to the right, into connected parts. These connected parts are clustered to the corresponding word.

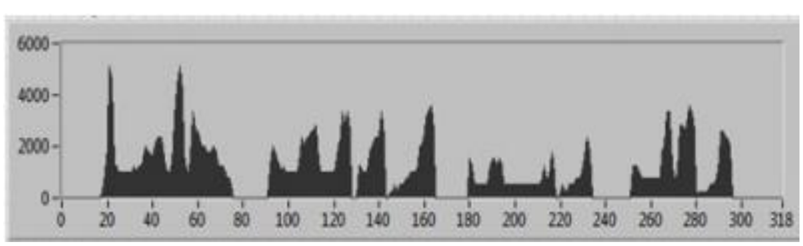


Figure 4.18: Clustered text line

Character segmentation process is the most important one in the OCR system, because character will be then entered to the recognition stage so it should be correctly separated with minimum error to be recognized correctly.

First we detect both base line (making horizontal projections and sum each row and pick the most dominant one) and we detect the Maximum transition line (we count in each row the transition from black to white and pick the row index with maximum transition), We then use the Maximum transition index and iterate on each column and if the transition is detected by a threshold, We calculate the start and end index as from black to white and from white to black is end index and then we choose a cut between them.

By constructing the vertical profile for the word image. The separation between two characters is considered as constant amplitude in the profile. A constant amplitude (low variation) passing filter is designed so that only low variations in the profile will be passed. The filter's output pulses are a locus of the characters connections. This locus takes the shape of separated sequential train of pulses. An image cutter tool is used on the original text image to extract sub-images which correspond to each pulse in the filter response. Each sub-image is extracted vertically starting from the first index to the last index of each pulse.

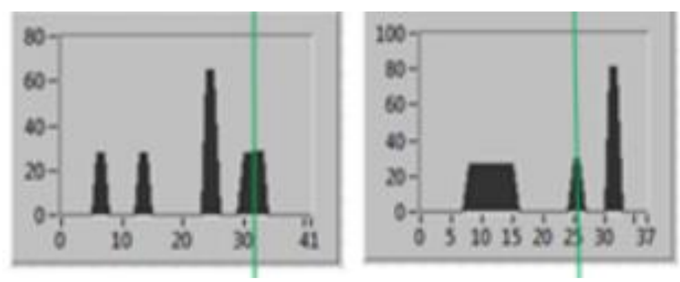


Figure 4.14: Illustration of Segmentation Process

4.6. Module 4: Character Recognition

This modules takes each letter or number that was separated in the previous stage, and then applies some techniques to compare and match the characters. In this module each character will go through the processing phase where feature extraction and classification for training the collected dataset will be applied using the Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) [24].

4.6.1. Functional Description

After the preprocessing and the segmentation stages each character will be classified, and the feature extracted by the Convolutional Neural Network (CNN) using backward and forward propagation technique, CNN performs two tasks which are feature extraction and classification to correctly classify images, and our proposed block diagram for our Character Recognition Module is shown below in figure 4.20.

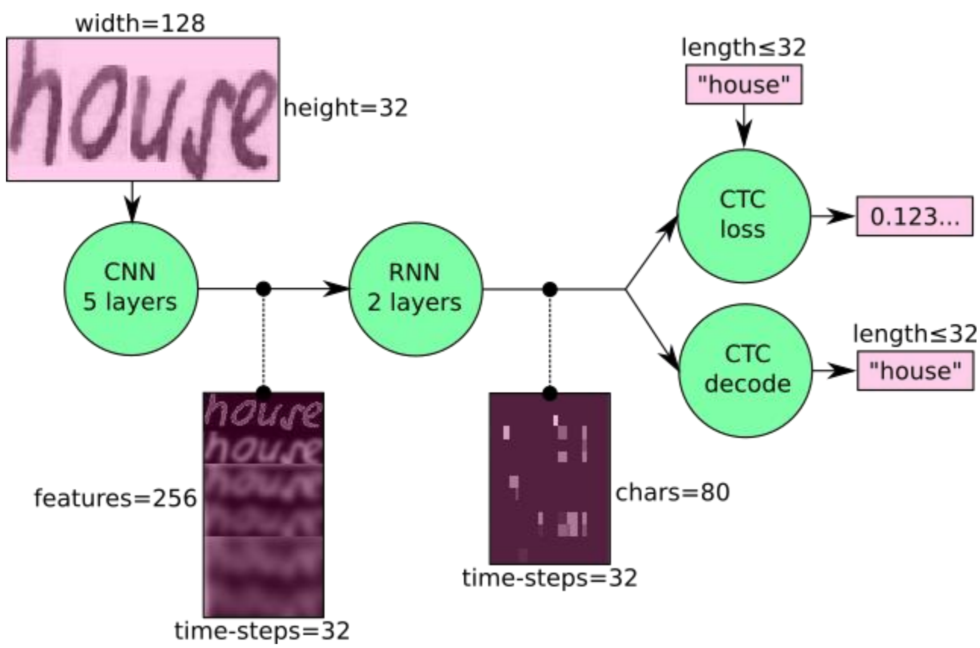


Figure 4.20: Block Diagram for Character Recognition Module

4.6.1.1. Feature Extraction Step

Firstly, we start the convolution step which includes the input image, a feature detector, and a feature map. Then the filter is taken and applied pixel block by pixel block through the multiplication matrix to the preprocessed middle image so the feature map is filled or completed. Many feature maps are created to get our first Convolutional layer.

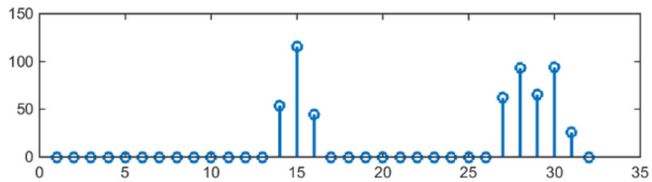


Figure 4.21: Snapshot of features extracted by CNN

4.6.1.2. The Rectified Linear Unit (ReLU) Layer

The Rectified Linear Unit (ReLU layer) is another step to the Convolution layer as an activation function is applied to the feature map to increase the nonlinearity in the network.

4.6.1.3. Max-Pooling and Flattening Layers

To achieve spatial variance, we use the max-pooling technique to gradually reduce the input representation size as it makes it easier to detect and identify objects wherever they are located inside the image. Not only does pooling helps in reducing the amount of processing and the number of required parameters required but also, it controls the issue of overfitting.

Finally, the pooled feature map is flattened into a sequential long vector to allow the information to enter the input layer in the ANN to be furtherly processed. In figure 4.22 we can see a clear description of how the process will be implemented, and the process is in the figure is just a simple way to look at the process as a whole, but that doesn't mean that we can't add more layers for our CNN. Adding more layers will definitely increase the accuracy, which is the goal.

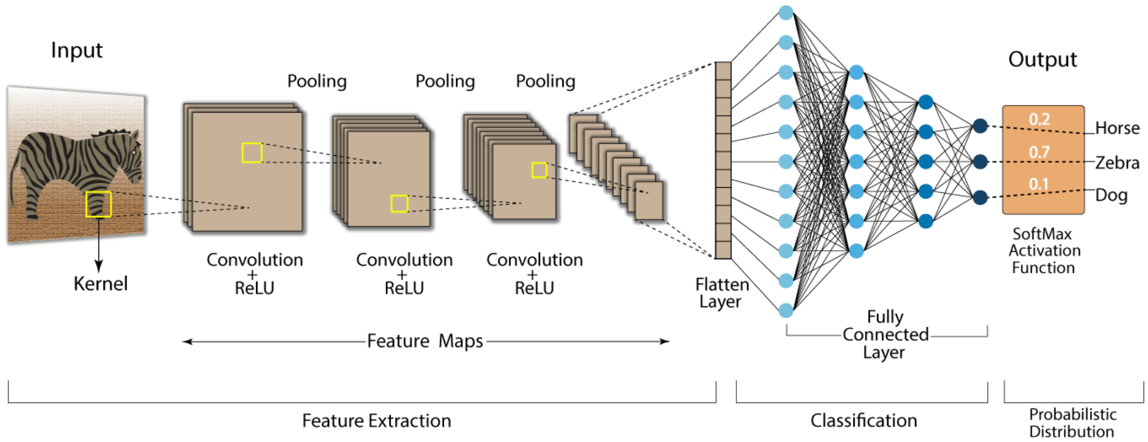


Figure 4.22: Clear description of the Processing module CNN

4.6.1.4. Classification and Training Layer

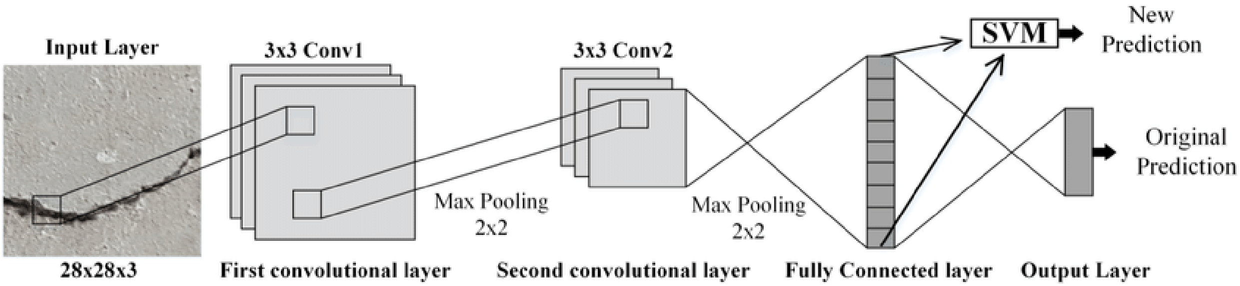


Figure 4.23: Classification Layer

After we train our model and making a dataset we choose a Gaussian SVM classifier then Neural Networks achieved much better accuracy, As we try to use different SVM classifiers like linear and Non-Linear Polynomial and Gaussian and Amongst the Gaussian kernel and polynomial kernel, we can see that Gaussian kernel achieved a better prediction rate while polynomial kernel misclassified sometimes. Therefore the Gaussian kernel performed slightly better. However, there is no hard and fast rule as to which kernel performs best in every scenario, but due to some runtime issues.

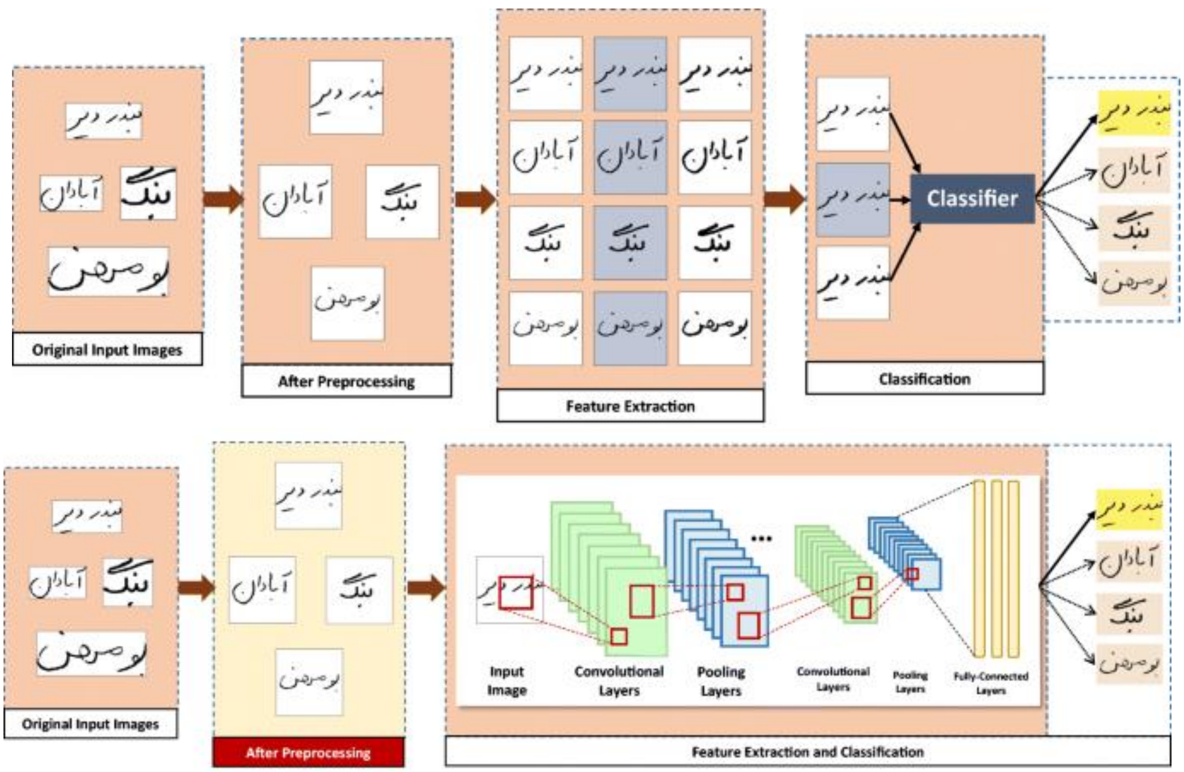


Figure 4.24: Character Recognition stages in depth

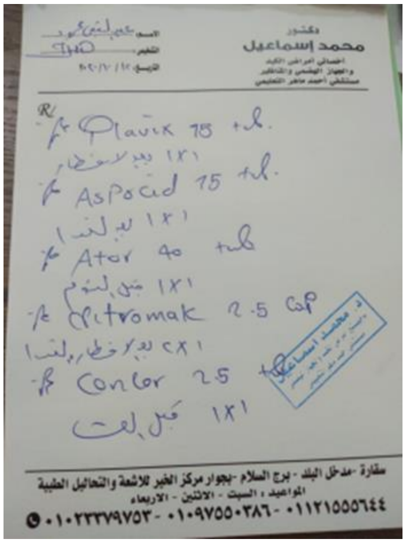
We trained, validated and tested the model on Google Colab22 platform using a single 12GB NVIDIA Tesla K80 GPU. Table 4.1 below shows the parameters we chose to train our model and all the settings the model has gone through. During the training phase, we evaluated the model on the validation set to ensure that the model doesn't over fit on the training data.

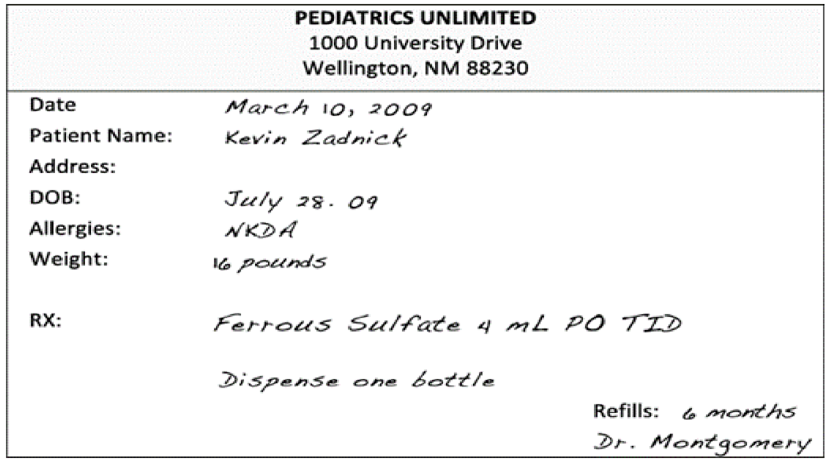
Training Parameter	Value
Optimizer	initial accumulator value of 0.1
Loss Function	Negative log likelihood at each time step 't' accompanied with coverage loss
Loss Function	Starting at 0.15 then decaying
Number of Epochs	15 epochs (260,000 training iterations)
Training Time	Approximately 4 days
Batch Size	16
Check Points	Every 1000 iterations
Coverage Loss Weight	ƛ = 1
Early Stopping	Loss is calculated on the validation set
Gradient Clipping	Clip gradients with norm greater than 2 to avoid gradient explosion problem in LSTM
Debugging and Monitoring	Tensorboard for debugging in python: Tensorflow library

Table 4.1: Character Recognition stages in depth

4.6.1.5. Dataset

The dataset is collected from multiple doctors and hospitals with varying specializations, our main aim is to collect numerous different prescriptions of each medicine with different hand-writings. The dataset has been divided into 70% training and 30% testing to train proposed model. The medical prescription is divided into 3 main parts, the first section until the R/ includes the name and the specialization of the doctor which will help in classifying the medicine according to the doctor's specialization, the second part which starts after the R/ which includes the handwritten prescribed medicines which will be classified and lastly the third part which is the footer includes the addresses and contact numbers of the hospital or clinic will be eliminated. There are various types of data sets as shown in the figure 4.25.




PEDIATRICS UNLIMITED
1000 University Drive
Wellington, NM 88230

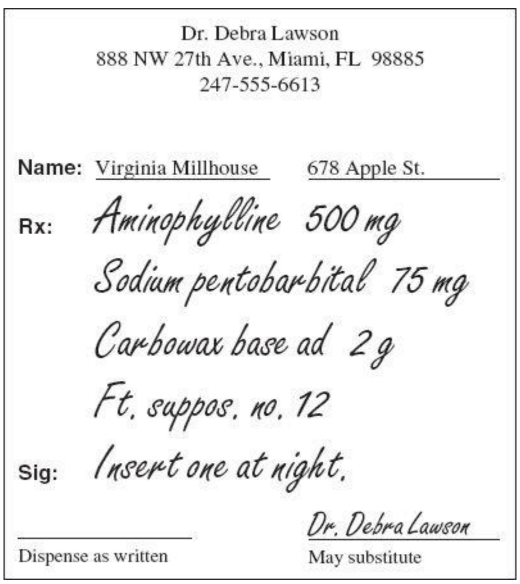
Date: March 10, 2009
Patient Name: Kevin Zadnick
Address:
DOB: July 28. 09
Allergies: NKDA
Weight: 16 pounds

RX: Ferrous Sulfate 4 mL PO TID

Dispense one bottle

Refills: 6 months

Dr. Montgomery


Dr. Debra Lawson
888 NW 27th Ave., Miami, FL 98885
247-555-6613

Name: Virginia Millhouse 678 Apple St.

Rx: Aminophylline 500 mg
Sodium pentobarbital 75 mg
Carbowax base ad 2 g
Ft. suppos. no. 12

Sig: Insert one at night.

Dispense as written

Dr. Debra Lawson
May substitute

Figure 4.25: various types of datasets

4.6.1.6. Recurrent Neural Network (RNN) Layer

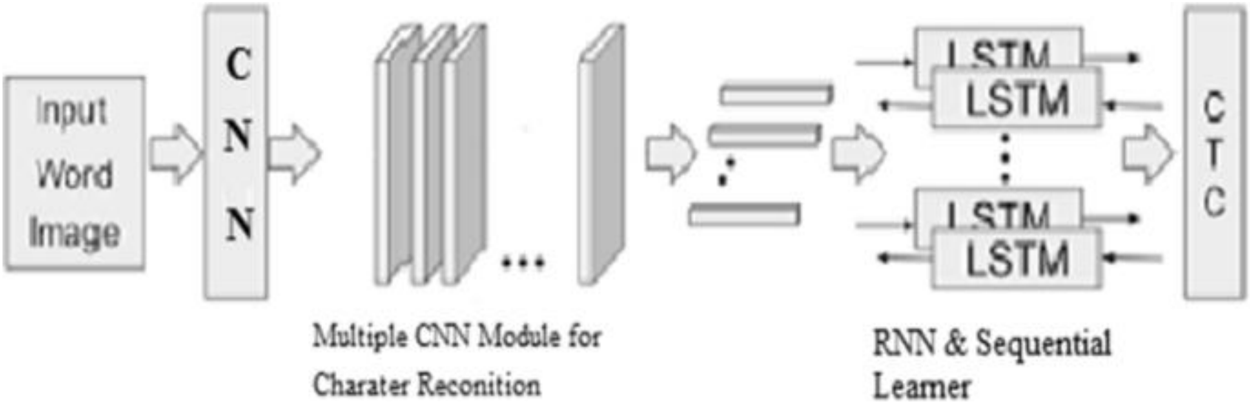


Figure 4.26: RNN Layer in the character recognition module

In this step we use Long Short Term Memory (LSTM) recurrent neural networks (RNN), LSTM RNN presents high raw performance and interesting training properties that allow us to break with the standard method at the state of the art. We present a simple and efficient way to extract from a single training a large number of complementary LSTM RNN, called cohort, combined in a cascade architecture with a lexical verification.

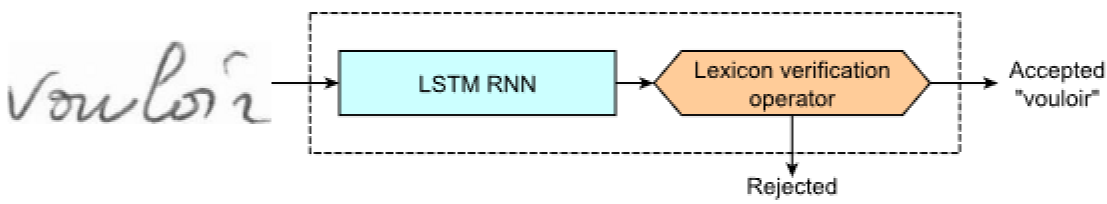


Figure 4.27: LSTM RNN Diagram

The Connectionist Temporal Classification CTC is a type of neural network output and associated scoring function, for training recurrent neural networks (RNNs) by allowing embedded training of character models from the word or sentence label without the need for knowing the location of each character.

The CTC is similar to the LSTM networks but more sufficient that's why we compared the accuracy between LSTM and the CTC and found out that the adding CTC after LSTM will make the accuracy better as we see in figure 4.20 below.

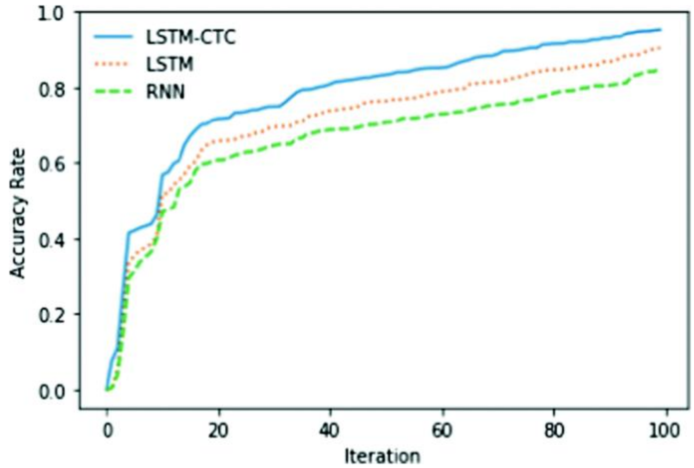


Figure 4.28: Accuracy score LSTM-CTC vs LSTM

We could have chosen to use Vanilla Beam Search (VBS) but it works on character-level and does not constrain its beams (text candidates) to dictionary words, that's why we decided to implement the CTC Word Beam Search decoder. The dictionary words are added to a prefix tree which is used to query possible next characters and words given a word prefix. Each beam of WBS is in one of two states. If a beam gets extended by a word-character (typically "a", "b" ...), then the beam is in the word-state, otherwise it is in the non-word-state.

4.7. Module 7: Database

In this module we are willing to have our organized output in a database.

4.7.1. Functional Description

We weren't sure whether to use MySQL database or Mongo database, so we recording the time required to insert the elements in both of the databases. We recorded the time from the beginning of the script runtime and until its completion, and as we see in figure 4.29 that 10,000 users were inserted into MySQL in 440 seconds, while in MongoDB, time was just 0.29 seconds.

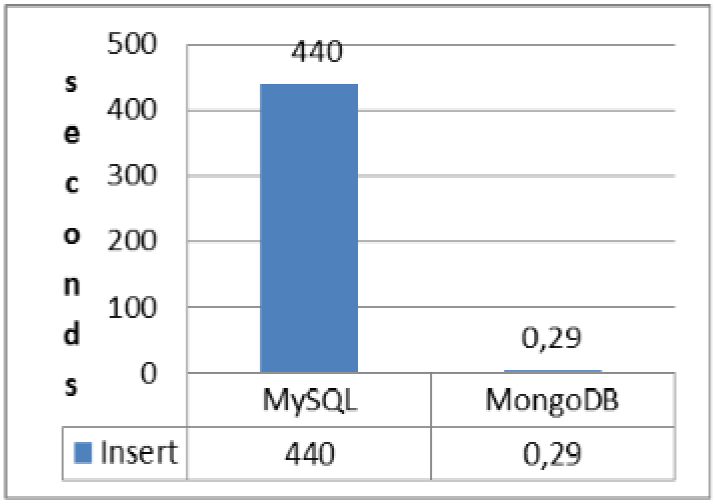


Figure 4.29: MySQL vs. MongoDB insert

That's why we decided to use Mongo Database, with the local server, and entered the data manually using JSON Language.

We created a new database named ScanMed, and added a new table named Products to add the details of our medicine, as it's shown in figure 4.30 below. We added details about the medicine, which is: Concentration, description, dosage, name, price, and type.

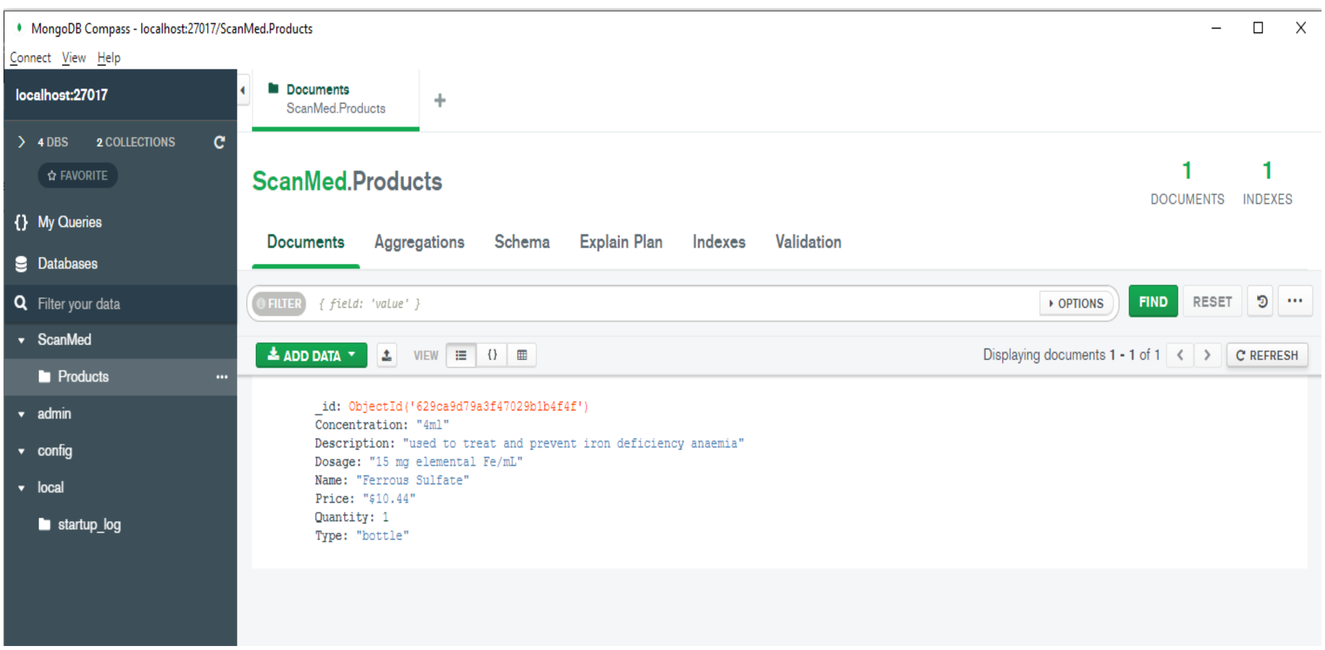
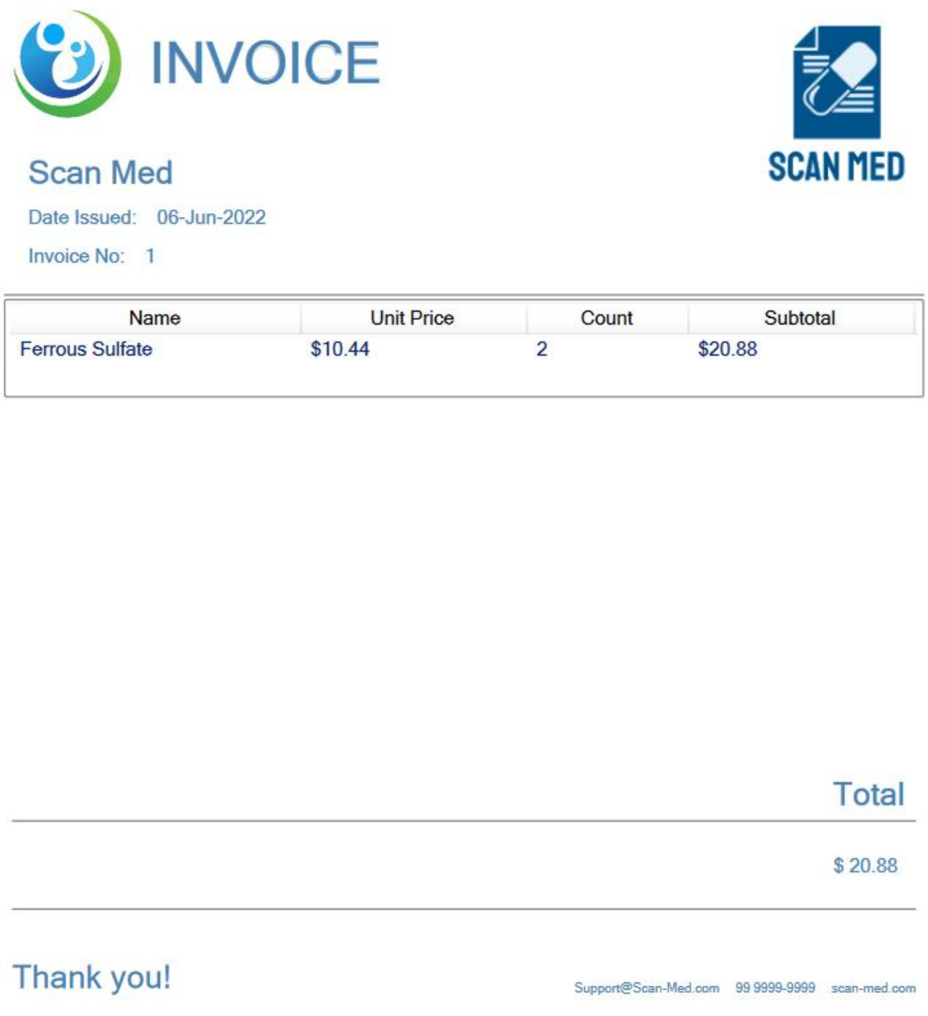


Figure 4.3015: Snapshot of Mongo database filled

And then we use that data stored in the database to make it easier for the user to checkout after he/she finds the correct recognized medicine by our system's GUI, and figure 4.31 is the invoice of this transaction.



INVOICE

SCAN MED

Scan Med

Date Issued: 06-Jun-2022

Invoice No: 1

Name	Unit Price	Count	Subtotal
Ferrous Sulfate	$10.44	2	$20.88

Total

$ 20.88

Thank you!

Support@Scan-Med.com 99 9999-9999 scan-med.com

Figure 4.3116: Snapshot of the invoice after looking at the medicine in the database

Chapter 5: System Testing & Verification

5. System Testing & Verification

References

[1] Ajay James, Bhagyasree P V, and Chandran Saravanan, "A Proposed Framework for Recognition of Handwritten Cursive English Characters using DAG-CNN April, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8741412/authors#authors [Accessed Jan. 21, 2022].
[2] Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, and Mueen Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive SystematicLiterature Review (SLR)," July, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9151144 [Accessed Jan. 21, 2022].
[3] Kevin George Joe, Meghna Savit, and K. Chandrasekaran, "Offline Character recognition on Segmented Handwritten Kannada Characters," Oct. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8978320/authors#authors [Accessed Jan. 21, 2022].
[4] Md. Anwar Hossain, and Sadia Afrin, "Optical Character Recognition based on Template Matching," May, 2019. [Online]. Available: https://www.researchgate.net/publication/333246578_Optical_Character_Recognition_based_on_Template_Matching [Accessed Jan. 21, 2022].
[5] Ocrsolutions.com, "OCR Medical Claims Systems Capture" Oct, 2019 [Online]. Available: https://ocrsolutions.com/software/document-capture/medical-claim/ [Accessed Jan. 02, 2022]
[6] Vezeeta.com, "Order all your medications from the pharmacy - Vezeeta.com" Feb, 2021 [Online]. Available: https://www.vezeeta.com/en-eg/pharmacy [Accessed Jan. 26, 2022]
[7] Yuming He, "Research on Text Detection and Recognition Based on OCR Recognition Technology," Sept. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9236870/authors#authors [Accessed Jan. 21, 2022].

WEE Water Engineering and Environment

STE Structural Engineering

PPC Petro Chemical Engineering

MDE Mechanical Design Engineering

CEM Construction Engineering and Management

CCE Communication and Computer Engineering

AET Architectural Engineering and Technology