

KANTIPUR ENGINEERING COLLEGE

AFFILIATED TO TRIBHUVAN UNIVERSITY

DHAPAKHEL, LALITPUR



SUBJECT CODE: CT654

A MINOR PROJECT PROPOSAL ON

SNAPTAG: CTRL+F FOR YOUR HANDWRITTEN NOTES

SUBMITTED BY:

AJAYA CHAUDAHRY [KAN077BCT007]

GAURAV GIRI [KAN077BCT034]

IZA K.C. [KAN077BCT039]

LAKESH SHRESTHA [KAN077BCT046]

**In Partial Fulfillment of the Requirement for the Degree of
Bachelor in Computer Engineering**

SUBMITTED TO:

**DEPARTMENT OF COMPUTER AND ELECTRONICS
ENGINEERING**

August, 2023

SNAPTAG: CTRL+F FOR YOUR HANDWRITTEN NOTES

SUBMITTED BY:

AJAYA CAHUDHARY [KAN077BCT007]

GAURAV GIRI [KAN077BCT034]

IZA K.C. [KAN077BCT039]

LAKESH SHRESTHA [KAN077BCT046]

**In Partial Fulfillment of the Requirement for the Degree of
Bachelor in Computer Engineering**

SUBMITTED TO:

**DEPARTMENT OF COMPUTER AND ELECTRONICS
ENGINEERING**

KANTIPUR ENGINEERING COLLEGE

DHAPAKHEL, LALITPUR

August, 2023

TABLE OF CONTENTS

TABLE OF CONTENTS.....	i
LIST OF FIGURE.....	iii
LIST OF ABBREVIATION.....	iv
CHAPTER 1 INTRODUCTION.....	1
1.1 Background.....	1
1.2 Problem Statement.....	1
1.3 Objective.....	2
1.4 Application and Scope.....	2
1.5 Features.....	3
1.6 Feasibility Study.....	3
1.6.1 Economic Feasibility.....	4
1.6.2 Technical Feasibility.....	4
1.6.3 Operational Feasibility.....	4
1.6.4 Schedule Feasibility.....	4
1.7 System Requirements.....	5
1.7.1 Development Requirements.....	5
1.7.2 Deployment Requirements.....	6
CHAPTER 2 LITERATURE REVIEW.....	7
2.1 Related Research.....	7
2.2 Proposed System.....	9
2.2.1 Block diagram for the proposed system.....	9
CHAPTER 3 METHODOLOGY.....	10
3.1 Working Mechanism.....	11
3.1.2 Pre-processing.....	11
3.1.3 Document Detection.....	11

3.1.4 Segmentation.....	12
3.1.5 Features Extraction.....	12
3.1.5.1 Training of Convolutional Neural Network.....	12
3.1.5.2 Document Recognition.....	14
3.1.6 Named Entity Extraction using spaCy.....	15
3.2 Algorithm Used.....	16
3.3 Software Development Model.....	17
CHAPTER 4 EPILOGUE.....	18
REFERENCES.....	20

LIST OF FIGURES

Fig 1.1 : Gantt Chart.....	5
Fig 2.1: Schematic block diagram of proposed system.....	9
Fig 3.1: Block diagram for the working mechanism proposed system.....	10
Fig 3.2: Dataset overview for the A good CHoiCe.....	13
Fig 3.3: Visual representation of E MNIST dataset.....	13
Fig 3.4: Configured layers of Convolution Neural Network.....	14
Fig 3.5: Incremental model for development of SNAPTAG.....	17
Fig 4.1: Expected Application Output.....	20

LIST OF ABBREVIATIONS

CNN: Convolutional Neural Network

EMNIST: Extended Modified National Institute of Standards and Technology

NER: Named Entity Recognition

OCR: Optical Character Recognition

PDF: Portable Document Format

ReLU: Rectified Linear Unit

CHAPTER 1

INTRODUCTION

1.1 Background

When it comes to self-study or informal research, taking quick notes has long been a favored practice. These notes serve as a valuable tool for reviewing and revisiting the gathered information. Traditionally, pen and paper have been the preferred medium for note-taking, as they offer a tangible and familiar experience. However, a common challenge arises when these notes need to be quickly referenced or accessed for a comprehensive overview.

Often, when we want to review the content of our notes, we may not have them readily available. They might be left behind, misplaced, or simply inaccessible at the time we need them. Additionally, the physical nature of pen and paper notes means that they are not always well-preserved over an extended period. They can get lost, damaged, or suffer from wear and tear, making them unreliable for long-term storage.

One approach to address the challenges of organizing and accessing handwritten notes is to digitize them by storing them in the gallery or creating PDF documents. While this offers convenience and preservation, it introduces its own set of issues. The unorganized nature of storing notes as image files or PDFs makes it difficult to quickly locate specific notes. Furthermore, the inability to easily search handwritten content within these digital formats hinders efficient information retrieval. Existing OCR tools that enable searching of handwritten text often come with premium access restrictions, creating a barrier for users who desire a straightforward OCR solution without additional costs. Therefore, there is a need for a free and unrestricted solution that provides efficient organization and searching of handwritten notes, allowing users to easily access and retrieve information.

1.2 Problem Statement

The problem at hand is the lack of a free and unrestricted solution for organizing and searching handwritten notes efficiently. The existing methods of storing notes in the gallery or creating PDF documents offer convenience but lack proper organization and search functionality. The handwritten content within these notes cannot be easily and accurately searched using available tools, often requiring premium access for OCR operations. This limitation hampers the ability of students and researchers to retrieve specific information quickly and effectively.

The objective of the project is to develop a solution that addresses these issues by providing a user-friendly platform for organizing handwritten notes and enabling efficient keyword-based searching. This solution should be accessible, free of charge, and not restricted by premium access limitations. By leveraging OCR technology, the project aims to convert handwritten keywords within the notes into searchable digital text, making it easier for users to retrieve relevant information when needed. Ultimately, this project seeks to empower students and researchers by providing them with a convenient and efficient tool for managing their handwritten notes and facilitating seamless information retrieval.

1.3 Objective

- I. To create a free and unrestricted platform for efficient organization and searching of handwritten notes, leveraging OCR technology to enable easy digitization and retrieval of information.

1.4 Application and Scope

1. This software can be utilized by researchers and academics to digitize and organize their handwritten research notes, allowing for efficient retrieval of information during the writing process. The OCR functionality enables quick keyword searching within the notes, helping researchers locate relevant information, references, and supporting evidence for their papers.

2. Students can leverage the software to digitize and organize their handwritten study notes, lecture summaries, and practice questions. The ability to search for keywords within the notes makes it easier for students to locate specific topics or concepts while preparing for exams.
3. Language learners can utilize the software to digitize their handwritten vocabulary lists, grammar rules, and language exercises. This enables them to access them quickly, helping in retention by facilitating vocabulary acquisition and language practice.

1.5 Features

1. Handwritten Note Digitization:

The platform allows users to easily convert their handwritten notes into digital format through scanning or capturing images, ensuring easy access and preservation.

2. Keyword-Based Searching:

Leveraging OCR technology, the platform enables users to search for specific keywords within their handwritten notes, providing quick and accurate retrieval of relevant information.

3. Cross Platform:

The platform can be accessed from various devices, such as computers, tablets, and smartphones, allowing users to retrieve their notes at any time.

4. Cost-Free and User-Friendly:

The platform is free to use and offers a user-friendly interface, ensuring that students and researchers can efficiently organize and search their handwritten notes without any financial barriers.

5. Efficient Content Indexing:

The platform automatically indexes the handwritten text within the notes, generating a searchable database of keywords. The indexing feature significantly improves the speed and accuracy of searching for specific information.

1.6 Feasibility Study

Before implementation of project design, the feasibility analysis of the project must be done to move any further. The feasibility analysis of the project gives an idea on how the project will perform and its impact in the real world scenario. So, it is of utmost importance.

1.6.1 Economic Feasibility

Our system is economically achievable as a result of the development of several tools, libraries, and frameworks. Since all the software required to construct it is free and readily available online, this project is incredibly cost-effective. Only time and effort are needed to create a worthwhile, genuinely passive system. The project doesn't come at a substantial cost. From an economic standpoint, the project appears successful in this sense.

1.6.2 Technical Feasibility

The software needed to implement a project can be downloaded from a wide variety of online resources. Technically speaking, the project is feasible as the necessary software is easily available. We were able to learn the information we required for the project through a variety of online sources, including classes. All the libraries and data are accessible online for free because this project does not require any licensing costs. It is technically possible if one has the necessary information and resources.

1.6.3 Operational Feasibility

Our project intends to improve hand-written notes taking experiences with a primary focus on searching keywords in notes/documents to make it more accessible. The project is intended for a sizable group of learners and students who desire to maintain their handwritten notes. Consequently, our project is operationally feasible.

1.6.4 Schedule Feasibility

The workload of the project is divided amongst the project members. The scheduling is done according to an incremental model where different modules are planned to be assigned to the group members. So, the project fulfills the schedule feasibility requirements.

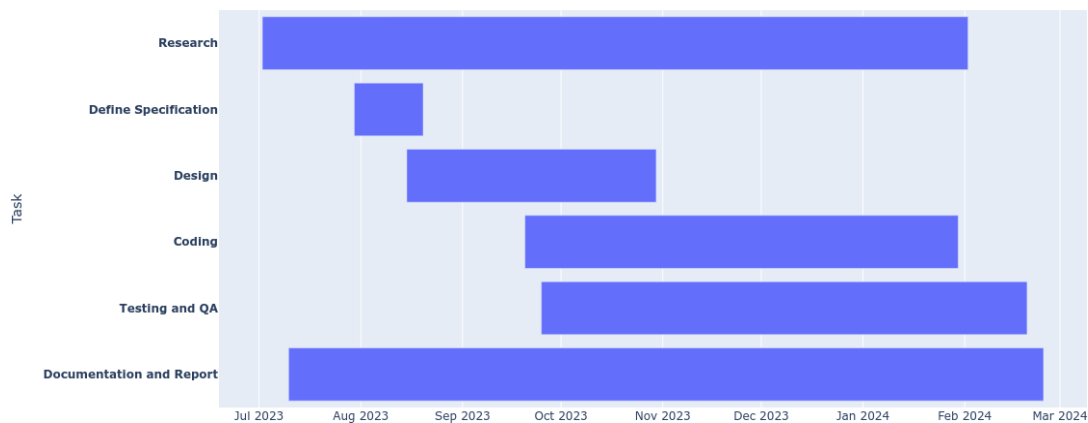


Fig 1.1 : Gantt Chart

1.7 System Requirements

1.7.1 Development Requirements

Table 1: Development Requirements

Software Requirements	Hardware Requirements
Programming Language: Python, Dart, Java	Camera: ≥ 12 Megapixels
Design Tools: Figma	RAM: ≥ 8 GB
OpenCV Library, spaCy	CPU: i5 10th (Recommended)
Frameworks: Flutter, Tensorflow	GPU: P100

Testing and Debugging Tools	Storage: ≥ 10 GB
-----------------------------	-----------------------

1.7.2 Deployment Requirements

Table 1: Deployment Requirements

Software Requirements	Hardware Requirements
Android: ≥ 10	Camera: ≥ 12 Megapixels
Read/Write FileSystem	RAM: > 4 GB
Internet Accessibility	Storage: ≥ 5 GB

CHAPTER 2

LITERATURE REVIEW

Regardless of whether they are documented or not, every project has helped to shape the world as it is today. Other researchers can benefit from documented projects by learning specifics about problems and how to solve them. Additionally, they boost project efficiency by removing the need to start the project from the beginning and specifying the starting point.

2.1 Related Research

This paper introduces an intelligent and efficient handwritten script recognition system using deep convolutional neural networks (CNNs). The research addresses several challenges in handwritten character recognition, such as distorted shapes, varying writing styles, and curved characters. To overcome these issues, the proposed method incorporates extensive image processing and pattern recognition techniques. The research utilizes a large dataset of 65,000 handwritten characters' images for training and testing the CNN model. The deep convolutional neural network, consisting of multiple hidden layers with numerous neurons, is trained on this dataset to recognize handwritten scripts. The system also employs three segmentation algorithms: line-based, word-based, and character-based, to accurately separate characters in the script before recognition. The results show promising accuracy of 90% in recognizing handwritten scripts and effectively converting them into electronic text documents[1].

This paper presents a thorough investigation into the challenging problem of quadrilateral detection in computer vision tasks. The detection of quadrilaterals is a crucial step in various applications, including document processing, signboards, whiteboards, license plates, and more. The paper provides an overview of existing approaches to quadrilateral detection, including contour-based, region-based, and vertex-based methods. It highlights the limitations and challenges faced by each approach, such as difficulties in handling occlusions, low contrast, strong lines, blurred images, and lack of a priori

information about object structure. To address the limitations of existing methods, the paper introduces a novel combination of contour and region-based approaches. The main idea is to rank competing contour hypotheses based on the contrast between areas inside and outside the border. By leveraging both approaches in the quadrilateral scoring process, the authors report a 40% decrease in ranking errors[2].

The paper introduces a novel system for automatically detecting signboards and extracting text from them, specifically tailored for guiding the navigation of autonomous robots indoors and outdoors. Unlike previous approaches, the system is not restricted to traffic signs and car plates; instead, it aims to detect a wide range of signboards, accommodating variations in shape, rotation, and scaling. The authors employ the Hough transform and other advanced processing schemes to achieve their goal. The system primarily focuses on identifying quadrilateral signboards present in images, assuming the entire signboard is visible and avoiding detection of overly small signboards with limited recognizability. To enhance text extraction accuracy, a geometric transformation is proposed to convert the detected quadrilaterals into rectangular shapes. The paper also discusses the segmentation of text from the found signboards. While the approach shows great promise, providing more extensive experimental evaluations and detailed discussions of segmentation techniques would further strengthen the paper's impact[3].

The paper highlights the significance of good benchmarks and standardized problems in machine learning and computer vision, particularly in the context of rapid advancements in these fields. It emphasizes the need for diverse benchmark tasks to comprehensively evaluate learning approaches and techniques. The authors introduce a suite of datasets called Extended Modified NIST (EMNIST) derived from the NIST Special Database 19, with the intention of providing a more challenging classification task for neural networks and learning systems. The datasets are designed to be drop-in replacements for existing systems, aligning with the image specifications and file formats of the well-known MNIST dataset. The paper thoroughly documents the dataset conversion process and presents benchmark results to validate the datasets. However, it acknowledges that MNIST, though widely used, may have lost its relevance as a challenging benchmark due

to exceptionally high accuracies achieved by deep learning models, questioning the labeling integrity of the dataset. In contrast, the EMNIST datasets offer a more complex task that ensures the longevity of meaningful benchmarks in the field[4].

2.2 Proposed System

Our proposed system finds the document along with preprocessing of the image. The dataset used for training the CNN is the combined dataset of EMNIST[5] and Complex Handwritten Character Dataset[6]. After OCR, the unique words are extracted using named entity recognition which are used as tags for searching the document later when stored.

2.2.1 Block diagram for the proposed system

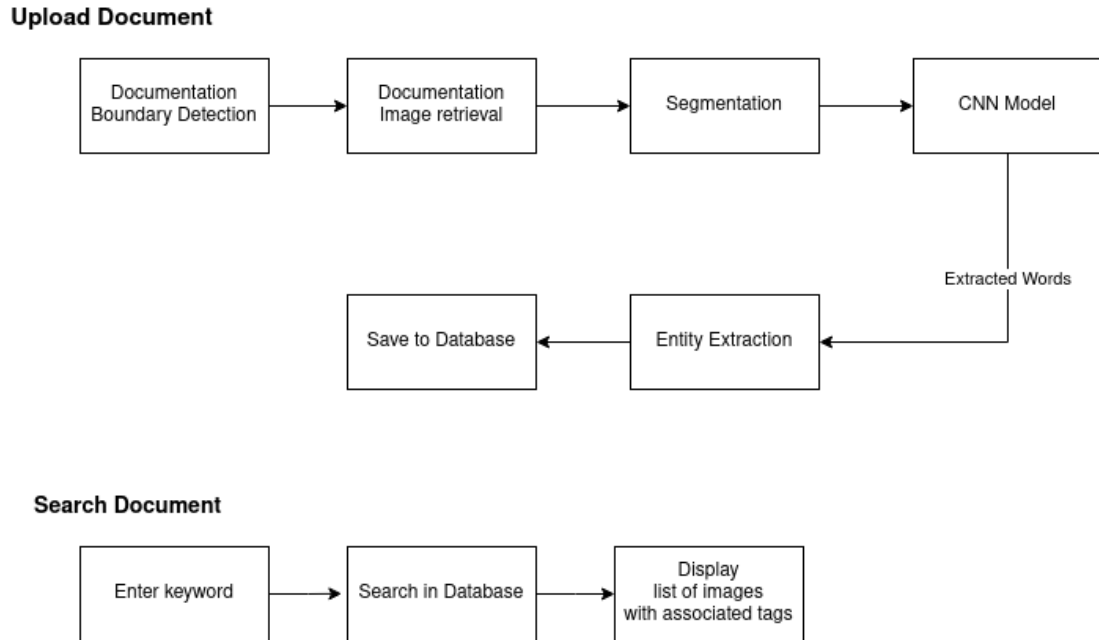


Fig 2.1: Schematic block diagram of proposed system

CHAPTER 3 METHODOLOGY

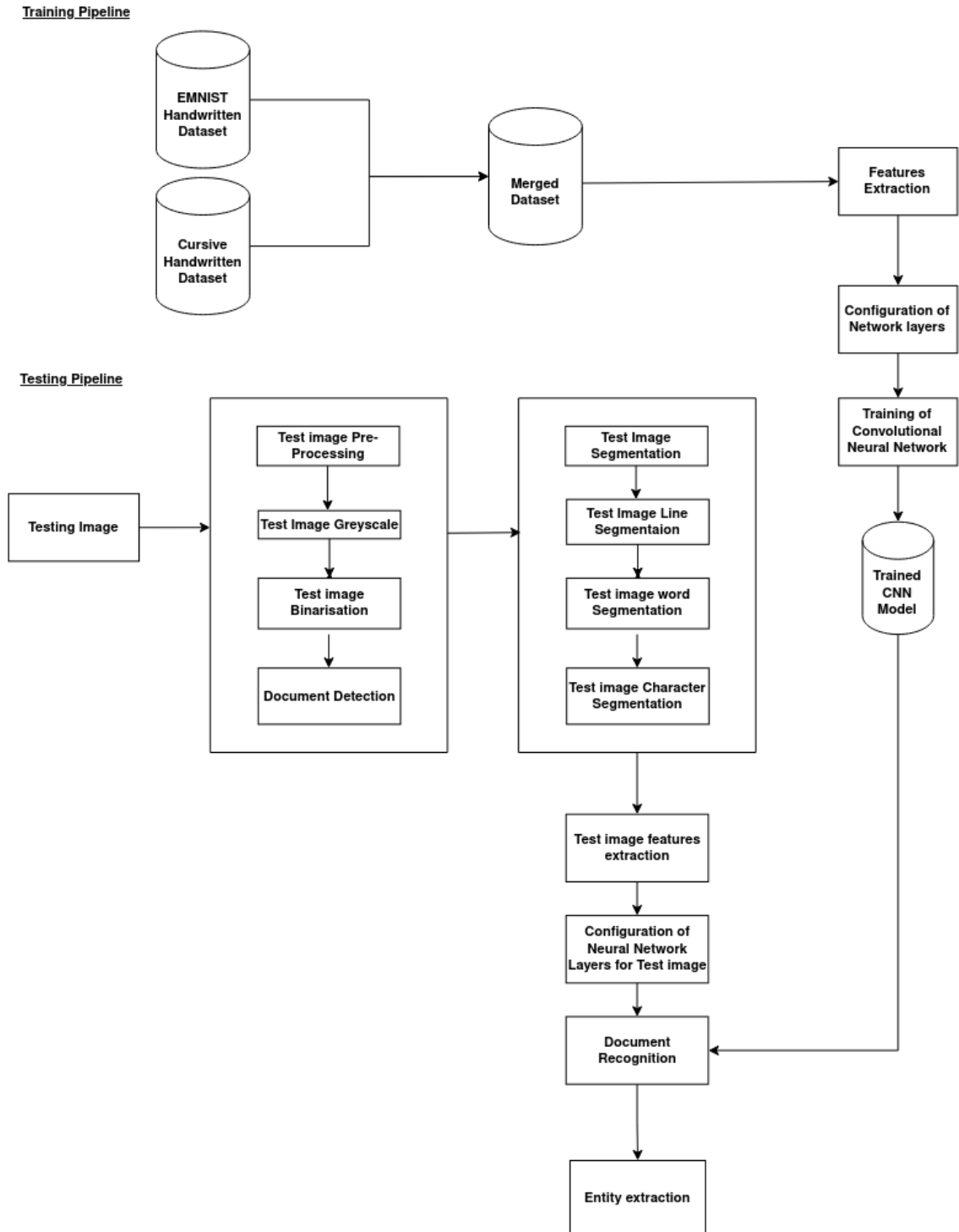


Fig 3.1: Block diagram for the working mechanism proposed system

3.1 Working Mechanism

The methodology diagram is shown above fig no. 3.1 which consists of several stages such as image acquisition i.e. testing image, pre-processing (grayscale conversion, binarization, document detection), segmentation (lines, words, and characters), feature extraction, and classification using CNN layers into 62 character classes.

3.1.1 Image Acquisition

The first step is to obtain a handwritten script through scanning physical documentation or images using a camera.

3.1.2 Pre-processing

It is essential to enhance the quality and prepare the image for further analysis. This step includes Grayscale Conversion, Binarization, Noise Removal and skew correction. Grayscale conversion is used to reduce complexity of handling multiple color channels (24-bit to 8-bit). Binarization is used to convert grayscale images into black and white to make it easier for segmentation.

Noise removal is used to remove unwanted variations in pixel intensity that degrade image quality. The proposed method uses a median filter to efficiently remove salt and pepper noise. This filter sorts adjacent pixel values and replaces the current pixel value with the median, reducing the noise and improving image.

3.1.3 Document Detection

The document detection is used for auto-cropping the image and correcting the perspective angle of the page of interest. For that quadrilateral points are obtained through the merged process of hough transformation and KMeans Clustering that clusters the intersection points into 4 groups which are the quadrilaterals. Then through the quadrilaterals, perspective transformation is performed to get a good perspective of the page[5].

3.1.4 Segmentation

In the segmentation process, we first tackle Line Segmentation, where the input image is analyzed to identify lines of text. By finding dark centroids in the image, we can mark the positions of each line. We employ a projection profile-based algorithm for this task and address the issue of skewed text by applying skew correction techniques.

Moving on to Word Segmentation, our approach involves morphological dilation, which allows us to connect individual characters in a line, ultimately helping us to extract separate words. Using labeling techniques, we efficiently isolate and extract words from the connected components.

Character Segmentation is a crucial step, particularly for cursive handwriting. To address the challenges of cursive characters, we implement skeletonization and vertical projection techniques to detect potential segmentation points. A distance-based approach is utilized to avoid over-segmentation, ensuring that characters like 'm', 'n', 'u', 'v', 'w', etc., are correctly identified. For untouched characters, a vertical projection is employed to identify spaces between characters, facilitating their separation.

Through these segmentation processes, we break down the handwriting image into lines, words, and individual characters, which is essential for accurate and efficient handwriting recognition.

3.1.5 Features Extraction

Feature extraction involves identifying important image features and saving them for further processing. It bridges the gap between pictorial and data representation in image processing. The proposed method employs a Convolutional Neural Network (CNN) for effective feature extraction, making the process more efficient and accurate.

3.1.5.1 Training of Convolutional Neural Network

The configured layers of Convolution Neural Network are shown below. To train the neural network, the dataset is divided into two sets: the training data and the testing data. The complex handwritten character dataset has data format of 28x28 ".png" format

picture. The data set has a total of 62 categories of 0-9, a-z and A-Z, corresponding to the files "0" to "61" in the order of "label.txt". The data set is divided into two parts, the unprocessed original data image is stored in the "0" to "61" in the "V0.3/data" folder, and the binarized data image Stored in "0" to "61" in the "V0.3/data-bin" folder [6].

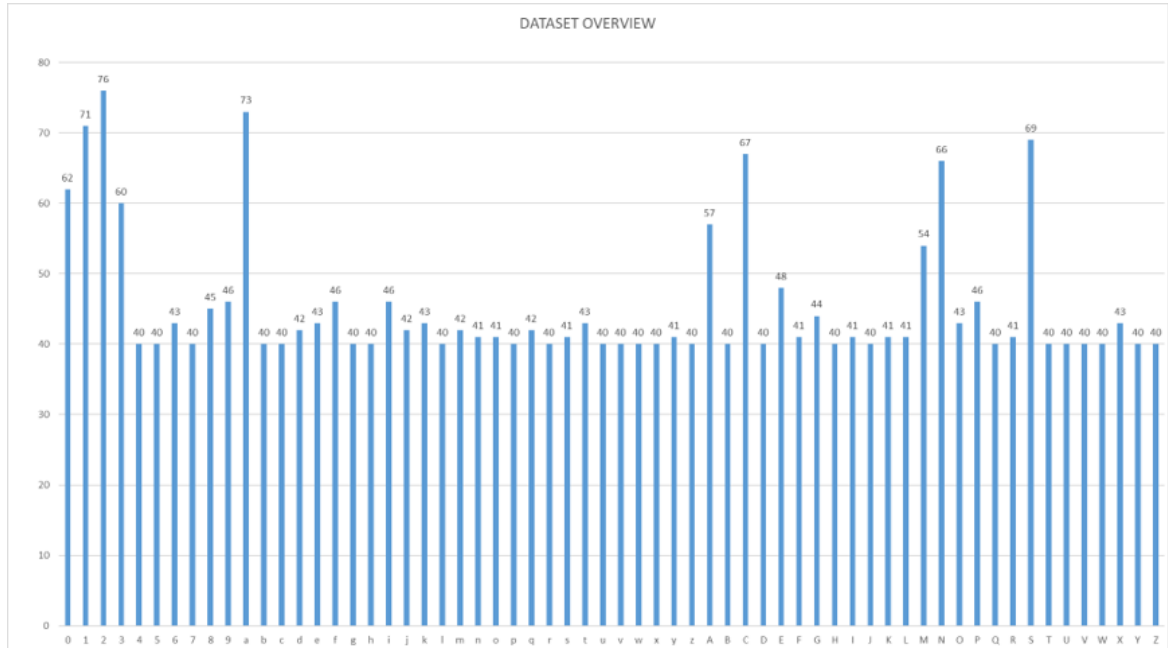


Fig 3.2: Dataset overview for the A good CHoiCe

The EMNIST also has 62 character classes divided into digit classes and letter classes containing a total 814,255 samples[5].

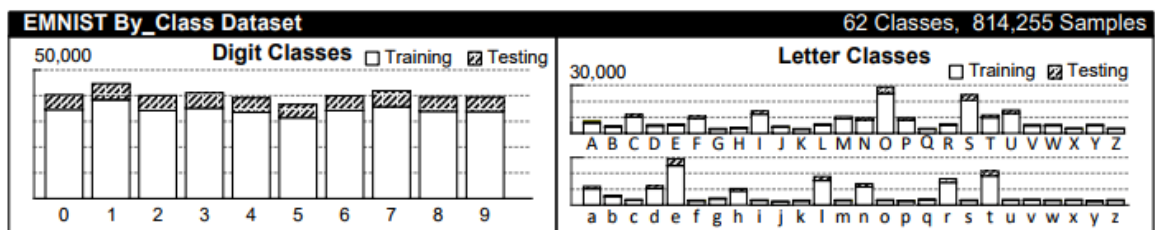


Fig 3.3: Visual representation of E MNIST dataset

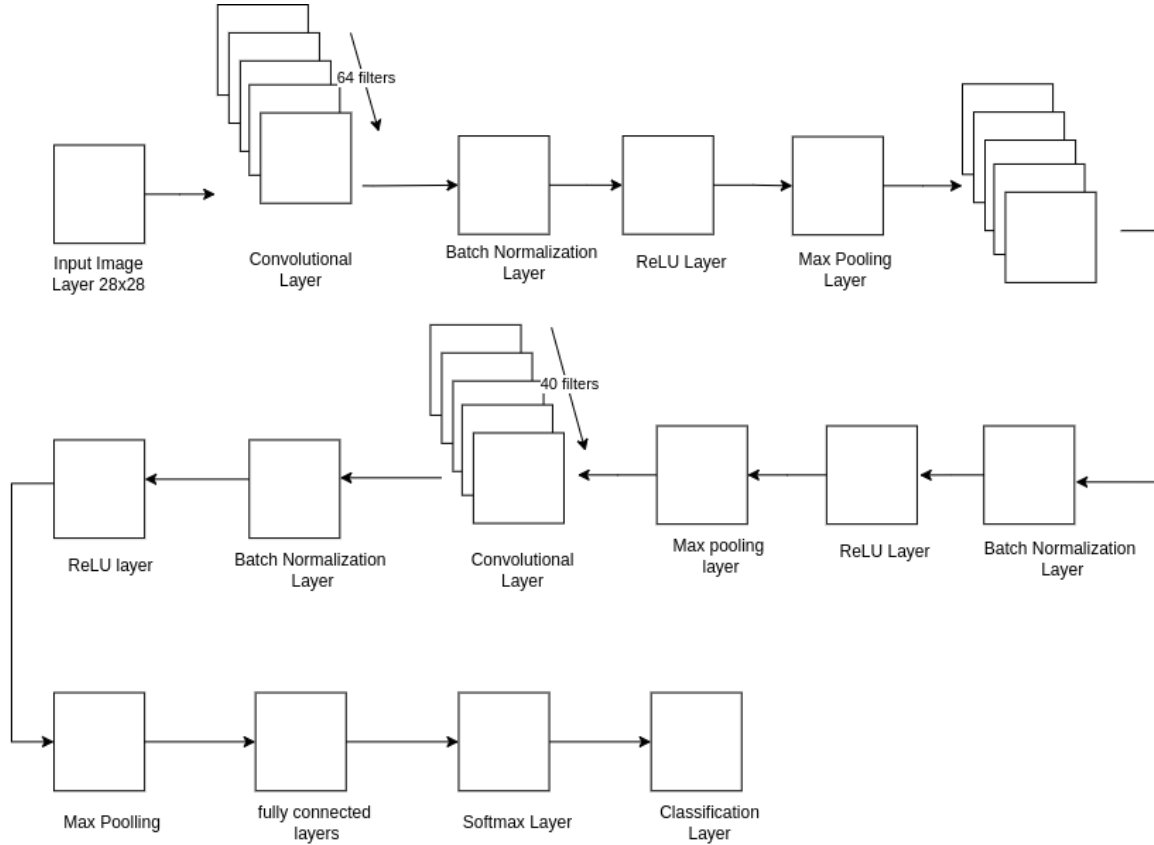


Fig 3.4: Configured layers of Convolution Neural Network

3.1.5.2 Document Recognition

The neural network used for image recognition involves four primary operations: Convolution, Non-Linearity (ReLU), Pooling, and Fully Connected Layer.

1. Convolution: The convolution operation extracts features from the input image M using a filter matrix N . The resulting feature map $f_{\text{con}}(k,l)$ can be calculated as follows:

$$f_{\text{con}} = \text{convolution}(m,n)$$

$$(m \otimes n)(k,l) = \sum_m \sum_n M(m,n)N(k-m,l-n) \dots \dots \dots (1)$$

2. Non-Linearity (ReLU): The ReLU function is applied element-wise to the feature map to introduce non-linearity and obtain the rectified feature map $\text{frec}(xk)$:

$$f_{\text{Rec}} = \text{ReLU}(x, l) = \max(0, x_k) \dots\dots\dots (2)$$

3. Pooling: Pooling reduces feature map dimensionality while preserving important information. The Max pooling technique selects the maximum value from a defined spatial matrix (e.g. 2x2):

$$f_{\text{pool}}(k, l) = \max(x_k, l) \dots\dots\dots (3)$$

4. Fully Connected Layer: In this layer, all neurons are connected to each other, and the output is used for classification.

Finally, the Softmax Layer outputs probabilities for each class of the handwritten characters. The Softmax function calculates the probability $p(O_i)$ for an input element x_k belonging to label i as follows:

$$p(O_i) = \text{softmax}(x_k) = \frac{\exp(x_k)}{\sum_{g=1}^y \exp(x_k)} \dots\dots\dots (4)$$

The Classification Layer then identifies the recognized character based on the highest probability assigned by the Softmax Layer.

3.1.6 Named Entity Extraction using spaCy

NER is a process in which anything that is denoted by a proper name or tag, for example, a location, an organization, or a person is identified as an entity. Named entities include things like geographical location, date, time, or money, and customization of the NER model for user-defined named entities is also possible[7].

spaCy is a free, open source library that allows advanced Natural Language Processing in Python. This library will be used to extract the unique word which will be used as tags for the image while searching it. Unique words are generated through named entity recognition for which “en_core_web_sm” is used, which is a linguistic model.

3.2 Algorithm Used

- **Line Segmentation:** The Line Segmentation algorithm starts with image padding to avoid segmentation errors. It computes the horizontal projection to identify text regions and detects dark lanes using a threshold in the binary image. The algorithm labels and counts dark regions to determine total lines of text. Centroids are calculated for precise positioning and stored for further processing. x and y coordinates of centroids are extracted. Text regions are cropped based on consecutive y-coordinates for accurate segmentation, enabling proper division of each handwritten script line for improved recognition and analysis.
- **Word Segmentation:** The Word Segmentation algorithm aims to extract individual words from previously segmented lines of handwritten text. It involves three steps: first, dilating the line image to expand the pixels and enhance visual distinction between connected components; second, labeling the connected components to identify distinct words; and finally, separating the labeled components to obtain individual images of each word. These separated images represent the segmented words of the handwritten text.
- **Cursive Character Segmentation:** The Cursive Character Segmentation algorithm separates connected characters in a handwritten text image by skeletonizing the image and computing its vertical projection. It marks segmentation points using the "Segmentation Points" (SP) array, starting from the first character. For each column and row, it checks if certain conditions are met to identify connected characters. If these conditions are fulfilled, the algorithm stores the column in the SP array based on a threshold distance. After marking the segmentation points, it draws red lines on the word image for visualization and stores 0 in the rows of SP to disconnect the characters. This process paves the way for further character segmentation and recognition.
- **Untouched Character Segmentation:** The Untouched Character Segmentation algorithm effectively segments individual characters in a handwritten text image that have not been separated previously. It begins by padding the image for better segmentation and computes the Vertical Projection (VP) to identify character

zones. Applying a threshold to the binary image detects dark lanes between letters. Each dark region is labeled, and the total character count is determined. Centroids are calculated for precise character positioning, and their x and y coordinates are extracted. The algorithm then crops text regions assuming a line between consecutive x-coordinates, ensuring accurate character segmentation. Cropped regions are stored separately, facilitating proper segmentation for subsequent recognition and analysis.

3.3 Software Development Model

This project will be developed using an incremental methodology since it offers a functioning prototype at an early stage of development. The requirements and scope of the project can be altered as necessary by studying the prototype. The rationale for the preference for this software development strategy is the flexibility offered by adopting the incremental technique. In this paradigm, the project goes through several releases or iterations prior to its official release

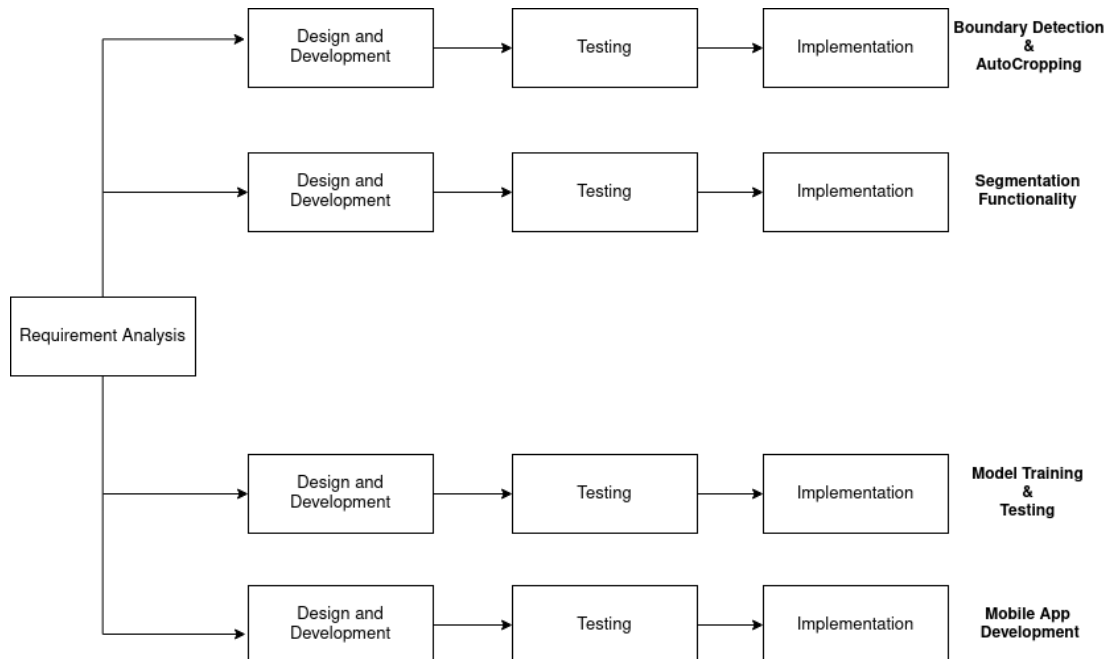
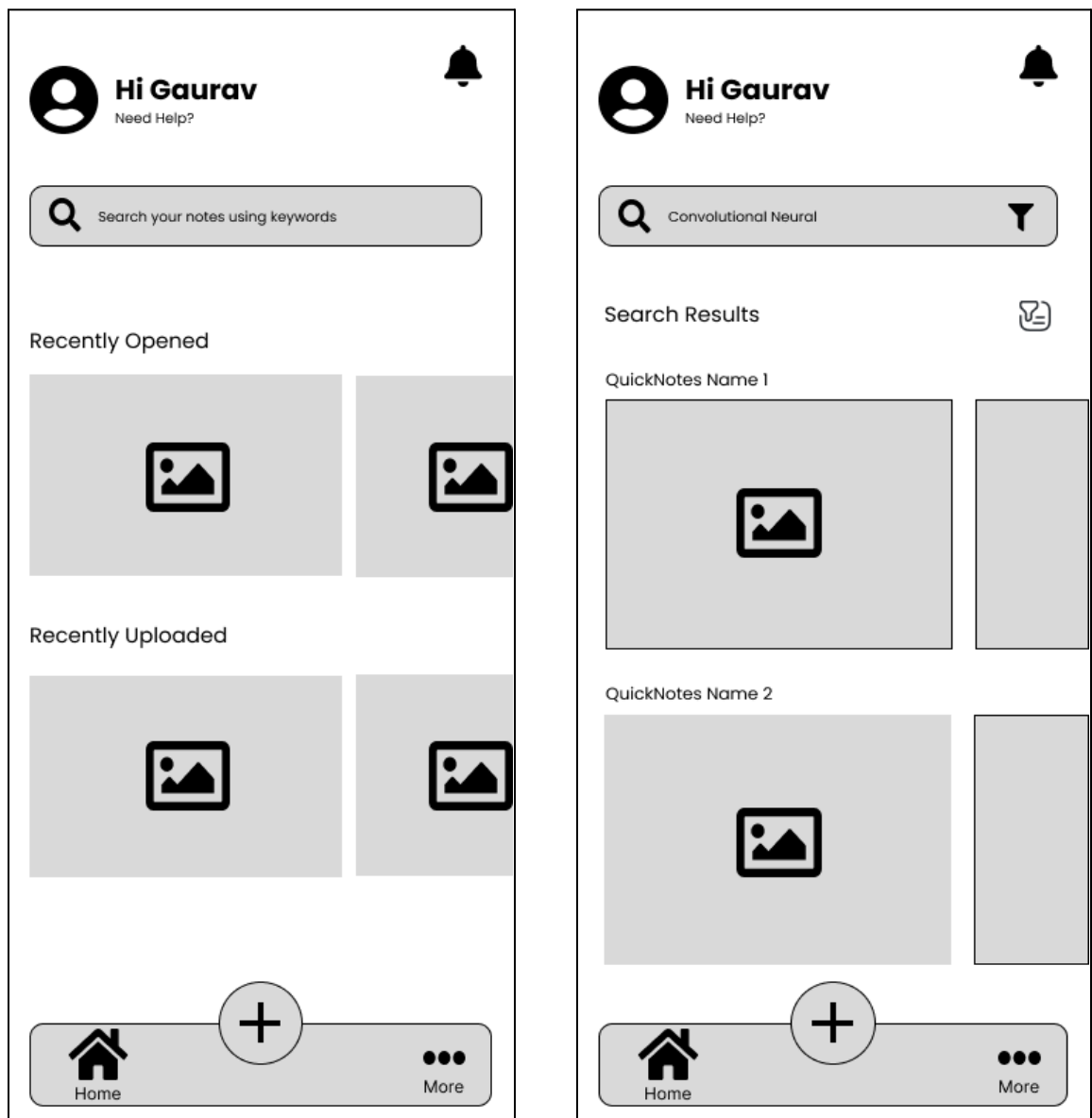


Fig 3.5: Incremental model for development of SNAPTAG

CHAPTER 4

EPILOGUE

The system is expected to produce a handwritten document scanner with search functionality. The system is anticipated to store the documents and correctly retrieve them upon searching the tags associated with them. The project also intends to develop a mobile application that includes the above mentioned features and is easy to use and convenient. The expected appearance of the system is shown above.



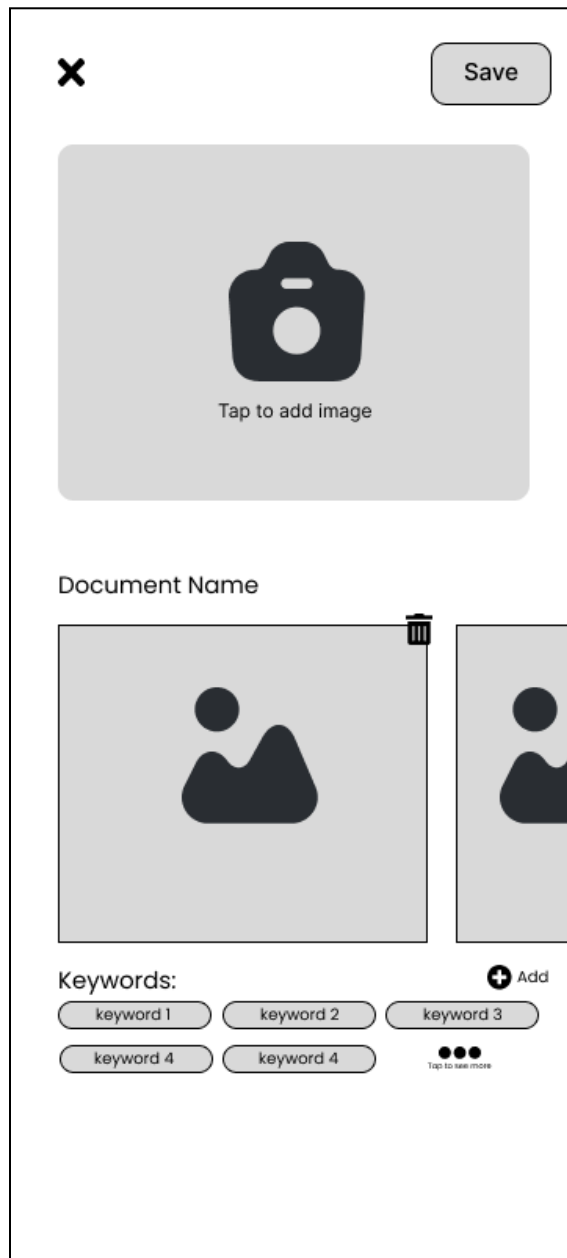


Fig 4: Expected Application Output

REFERENCES

- [1] S. Abbas , Y. Alhwaiti , A. Fatima , M.A. Khan , M. A. Khan ,T.M. Ghazal , A.Kanwal ,M.Ahmad and N.S. Elmitwally.“Convolutional Neural Network Based Intelligent Handwritten Document Recognition”,*Computers, Materials & Continua*, Tech Science Press, Vol.70, June 2021, 4564 - 4580
- [2] D.V. Tropin, S.A. Ilyuhin , D.P. Nikolaev, and V.V. Arlazarov, “Approach for Document Detection by Contours and Contrasts ”,International Conference on Pattern Recognition (ICPR),2020
- [3] A. Tam, H. Shen, J. Liu, and X. Tang, “Quadrilateral Signboard Detection and Text Extraction”, ResearchGate. 2003, 708-718
- [4] G. Cohen, S. Afshar, J. Tapson, and A.V. Schaik, “EMNIST: an extension of MNIST to handwritten letters”, arXiv. 2017
- [5] S. Ishfar. “Documentation Detection in Python”. *Medium.com*,2020.
<https://medium.com/intelligentmachines/document-detection-in-python-2f9ffd26bf65>
- [6] H. Zhang, T. Gedeon, S. Caldwell, and J. Plested, “A good CHoiCe: A Complex Handwritten Character Dataset”, *Github.com*, 2021.
<https://github.com/skyatmoon/CHoiCe-Dataset>
- [7] H. Shelar, G. Kaur, N. Heda and P. Agrawal “Named Entity Recognition Approaches and Their Comparison for Custom NER Model”. *Science & Technology Libraries*. 2020, 1-14