



MONASH University

FIT3163 Data Science Project Part 1

Project Initial Concept and Design

Data Analysis Report

Automated Health Information System

Team MDS2

Foo Kai Yan | 33085625 | kfoo0012@student.monash.edu

Alicia Quek Chik Wen | 33045240 | aque0004@student.monash.edu

Eunice Lee Wen Jing | 33250979 | elee0075@student.monash.edu

Jesse Yow San Gene | 32794649 | jyow0001@student.monash.edu

Supervised by:

Dr Muhammad Fermi Pasha

Table of Contents

Project Introduction.....	3
Project Goals.....	3
Problem / Opportunity Statement.....	4
Project Overview.....	4
Data Analysis.....	5
Introduction.....	5
Purpose of Data Analysis.....	5
Background Information.....	6
Problem Statement.....	6
Significance of Data Analysis.....	6
Data Collection.....	6
Data Pre-processing.....	7
File Format.....	7
Image Pre-processing.....	7
CSV files Pre-processing.....	9
Data Post-processing.....	9
Data Analysis.....	9
Methodology.....	9
Characteristics of Dataset.....	9
Missing Values.....	10
Data Inconsistency and Bias.....	11
Data Distribution.....	12
Results and Findings.....	13
Conclusion.....	14
Software Specification.....	14
Software Framework.....	14
Software libraries.....	15
Database system.....	16
Operating software.....	17
Programming Language environment.....	17
Visual Studio Code.....	17
Jupyter Notebook.....	18
Project management tool.....	18
Hardware Specification.....	19
CPU.....	20
RAM.....	20
GPU.....	20
Storage.....	20
Camera.....	20
Network Connection.....	21
References.....	21
Acknowledgement.....	24

Automated Health Information System

MDS02: Foo Kai Yan, Alicia Quek, Eunice Lee, Jesse Yow
School of Information Technology
Monash University Malaysia
Subang Jaya, Selangor 47500, Malaysia

Project Introduction

The focus of our project is on "Automated Health Information System", aimed at transforming the way healthcare data is managed and accessed. It incorporates a handwritten text recognition module to enhance efficiency and accuracy in data entry.

The rapid advancement of technology, along with the global impact of the pandemic, has accelerated the adoption of health information systems by healthcare professionals around the world. Healthcare institutions are filled with vast amounts of data from diverse sources, including clinical records, patient information, insurance records, and pharmaceutical data (Renju et al., 2015). Managing and analyzing this data can be a daunting task, leading to potential errors, delays, and inefficiencies in healthcare delivery (Shine, 1996). Nevertheless, the ongoing issue of sluggish data entry makes it difficult to accurately capture patient data, which impacts patients and healthcare providers alike during the registration process. Sometimes, doctors are really busy and find it hard to use digital systems for prescriptions as they are more comfortable to write down the prescriptions in a paper. They might not be used to key in the prescriptions into the computer and find them confusing. Our project's output is to create a mobile friendly webpage automated health information system that uses artificial intelligence (AI) technologies, like handwriting recognition, to address this problem. By using AI technologies like handwriting text recognition, healthcare professionals will no longer need to spend valuable time manually inputting data, reducing the risk of errors and increasing efficiency. At the end of our project, we aim to help both healthcare workers and patients by making data entry easier. Our system's goal is to make healthcare environments more productive, efficient, and organized.

Project Goals

Our project goals include:

- Implement and develop an advanced automated health information system incorporating handwritten recognition modules capable of accurately transcribing both cursive and non-cursive handwriting.
- Extend the web-based health information system to seamlessly integrate with a mobile application platform.
- Enable healthcare professionals to input patients' data by scanning the handwritten notes into the system, thereby expediting the data entry process.
- Reduce reliance on manual input methods to improve efficiency and productivity in healthcare workflows.
- Ensure accurate transcription of handwritten notes and prescriptions into patient records, enhancing data integrity and quality.
- Improve overall user experience and satisfaction by streamlining data entry tasks and facilitating access to patient information in real-time.
- Ensure that only authorized personnel can access the system database, preventing the leaking of patients' data or unauthorized editing of patients' information.
- Implement a scalable database design to handle the growing patients' data.

Problem / Opportunity Statement

In the realm of healthcare, the inefficiencies surrounding data entry processes pose significant challenges, both digitally and within physical registration counters. Traditional methods of patient data collection, such as manual filling of information forms and subsequent data entry by nurses, contribute to prolonged waiting times, increased likelihood of errors, and reduced overall efficiency in healthcare delivery.

The current system for collecting patient data at healthcare facilities involves manual data entry at registration counters, resulting in lengthy lines due to patients having to complete information forms. This not just results in longer waiting periods but also leads to annoyance and unease, especially for individuals who are already experiencing ill health. Additionally, nurses responsible for entering this information are under extra pressure, leading to possible mistakes in patient records and impacting the quality of care. These inefficiencies may also intrude on the valuable time patients have for appointments, as they ultimately decrease the clinic's overall effectiveness. Moreover, the fact that forms are filled out in a public setting can lead to worries about the privacy of confidential patient data.

Furthermore, during doctor consultations, handwritten prescriptions and diagnoses further exacerbate the issue, as nurses must decipher and manually input this information into the system, often encountering difficulties due to illegible handwriting. Additionally, extending system accessibility to mobile devices ensures faster input of patient information, particularly in clinics with limited computer resources.

The relevance of this problem lies in its direct impact on the efficiency and effectiveness of healthcare delivery. Prolonged waiting times and errors in data entry not only inconvenience patients but also compromise the quality of care provided. By addressing these challenges, through technology-driven solutions that optimize data entry processes, we have the opportunity to improve the overall patient experience, enhance healthcare efficiency, and ultimately contribute to better health outcomes for society as a whole.

Project Overview

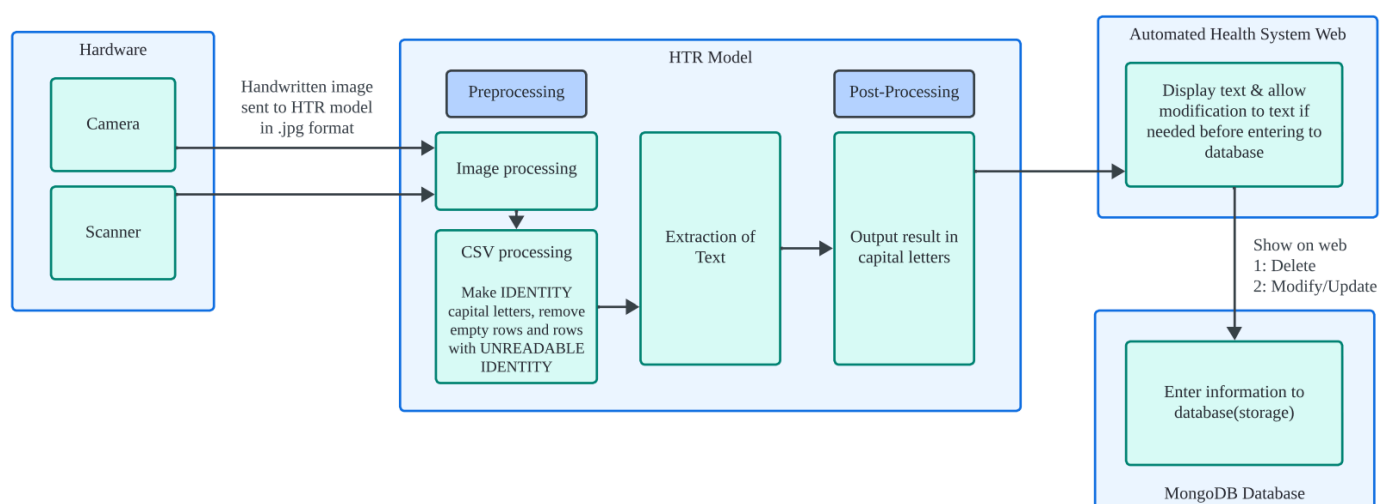


Figure 1. Project Diagram Representation

The diagram shown above (Figure 1) demonstrates a methodical and effective process created for entering handwritten patient information into a digital database. The step-by-step process illustrated in

Figure 1 simplifies the transfer of important patient data from handwritten records to a digital format. Healthcare professionals can improve the precision and efficiency of entering data by using this efficient technique, which also guarantees smooth integration of patient information into the electronic database system.

At first, the written text is scanned with a phone camera or scanner which is captured in JPG format, which is a commonly used image format for its broad compatibility in the image processing and deciphering stages. This structure guarantees both compatibility and simplifies manipulation and sharing on different platforms. After converting the handwritten images into JPG file format, the images would be sent to the Handwritten Text Recognition (HTR) model, which is a sophisticated technology that would be improved by the team to accurately analyze and interpret written text. The HTR model employs algorithms to analyze the recorded handwriting and convert it into editable digital text. This process of transformation is essential to improve the efficiency and accuracy of extracting data from physical documents. Using advanced machine learning techniques, the HTR model is able to differentiate between individual characters and words in handwritten content, making it easy to integrate into digital workflows. In the end, this efficient process allows businesses and individuals to convert old handwritten information into digital forms, improving productivity and making data management easier.

The HTR model undergoes several processing stages, including image preprocessing (refer to Figure 2 for more detail) and CSV preprocessing. CSV preprocessing involves standardizing the character identities within the text to ensure consistency. Lastly, there is post-processing, a crucial step undertaken to refine the recognized text further by ensuring uniformity. Through this meticulous process, the final output text is transformed into capital letters, enhancing its visual consistency and readability.

The output from the HTR model is then displayed on a web-based automated health information system, allowing users to review and modify the recognized text as needed before entering the information into the database. The web interface also provides functionalities such as delete, add, and update to facilitate basic data manipulation.

Lastly, the modified text is entered into a MongoDB database for storage, ensuring accessibility and organisation of patients' information. This integrated system speeds up the data entry process, hence enhances accuracy, and facilitates efficient management of healthcare data.

Data Analysis

Introduction

Purpose of Data Analysis

The goal of the data analysis report for this project is to examine and analyse existing handwritten text training dataset that would potentially be used in the project to train the HTR model. The HTR model would be used to identify handwritten information like patient registration information or diagnosis and input it into the health information system database.

The aim of this section is to give information on the training, validation and testing dataset's features, such as how data is gathered and prepared before being utilized to train and test the HTR model. This analysis serves as a detailed manual for making well-informed choices, offering detailed information on model effectiveness, integration with systems, ethical concerns, and the possible effects on patient data management in automated healthcare systems.

Background Information

The dataset obtained from Kaggle contains more than four hundred thousand handwritten names collected through charity projects and is utilized as the training dataset for the HTR model in this project. There are a total of 206,799 first names and 207,024 surnames contained in the dataset and the data would be broken up into test, training, and validation datasets (Handwriting Recognition, n.d.).

As part of the research plan, the team intends to collect additional data personally in the future. This data will extend the existing testing dataset by including filled-in patient registration forms. The primary objective of this extra step to collect data personally is to enhance the performance of the HTR model by ensuring accurate association of data with corresponding sections within the database.

Even though these handwritten documents are important tools needed for comprehending and identifying written text, a comprehensive analysis of the dataset is still required to fully grasp its dataset structure and address any potential challenges that might arise from it.

Problem Statement

Images containing handwritten text could have been captured in low light conditions, be blurry, or have a poor quality, which might potentially impact feature extraction accuracy and model performance (5 Common Issues Issues and Challenges in Digital Image Processing, 2019). Ensuring robustness against such variations becomes crucial for reliable text recognition hence, data analysis is done to extract vital features from the dataset that could aid in the analysis of text from images, improving data gathering techniques and the data collection process moving forward.

Significance of Data Analysis

The data analysis of this report plays a crucial role in improving the performance of the HTR model. By thoroughly analyzing this dataset, the team can gain insights into handwriting variations, patterns, and its challenges. This information can guide the team on preprocessing strategies which ensures accurate data association, and aids in creating a strong HTR model. Ultimately, the analysis contributes to model training, efficient patient information management and system reliability.

Data Collection

The dataset used for this analysis is obtained from the Handwriting Recognition dataset which was downloaded from Kaggle (Handwriting Recognition, n.d.). This dataset consisted of handwritten names collected through charity projects. The dataset is all contained within a Compressed (zipped) Folder (.zip) with a size of 1.25 GB (1,353,076,736 bytes) on disk. Inside the primary compressed folder, there are 3 CSV files along with 3 separate folders for testing, training, and validation purposes. The author has included the testing dataset in the testing, training, and validation folders, consisting of handwritten images in .jpg format. In contrast, the 3 csv files include the correspondence between the handwritten image and the transcribed handwritten names. The 3 csv files consist of 2 columns each: the image filename and the transcribed handwritten names of the image (Handwriting Recognition, n.d.).

The training dataset is utilized for the purpose of training the HTR model, allowing the model to adapt its weights in order to make correct text predictions based on the target variable. The purpose of the validation dataset will be utilized to assess the model's efficacy. It serves as a safeguard against overfitting, where the model is excessively customized to the training data and struggles to apply to unfamiliar data. The validation dataset aids in creating a model that can effectively be applied to unseen

data. After training and validating the HTR model, the testing dataset is utilized to give an impartial, conclusive assessment of its performance. It is utilized only after the model has been tuned successfully.

As discussed previously, it was mentioned that handwritten text images may have been taken in different uncontrollable conditions, like different lighting conditions, leading to suboptimal image quality for the HTR model. This brings about many unwanted complications like the brightness of the lighting, the legibility of the text, the existence of background noise, and artifacts. Background sounds and artifacts such as ink smudging, spots, marks, or folds may disrupt the precision of the HTR model since nearby characters in a text could combine, thereby complicating the segmentation of individual letters or the identification of words for the HTR model.

Data Pre-processing

File Format

The dataset is all contained within a zip file on disk but because the zip file is too large, code will be employed to extract the data from it.

Image Pre-processing

Each individual image will go through pre-processing tasks like enhancing contrast, removing noise, greyscaling and binarization to enhance the quality of input images before inputting them into the HTR model (Patel, 2023). After completing that stage, the pictures will be resized to accommodate the handwritten text contained in the image, cutting out any extra or unnecessary parts so the HTR model would not need to have any additional and unnecessary computational needs for.

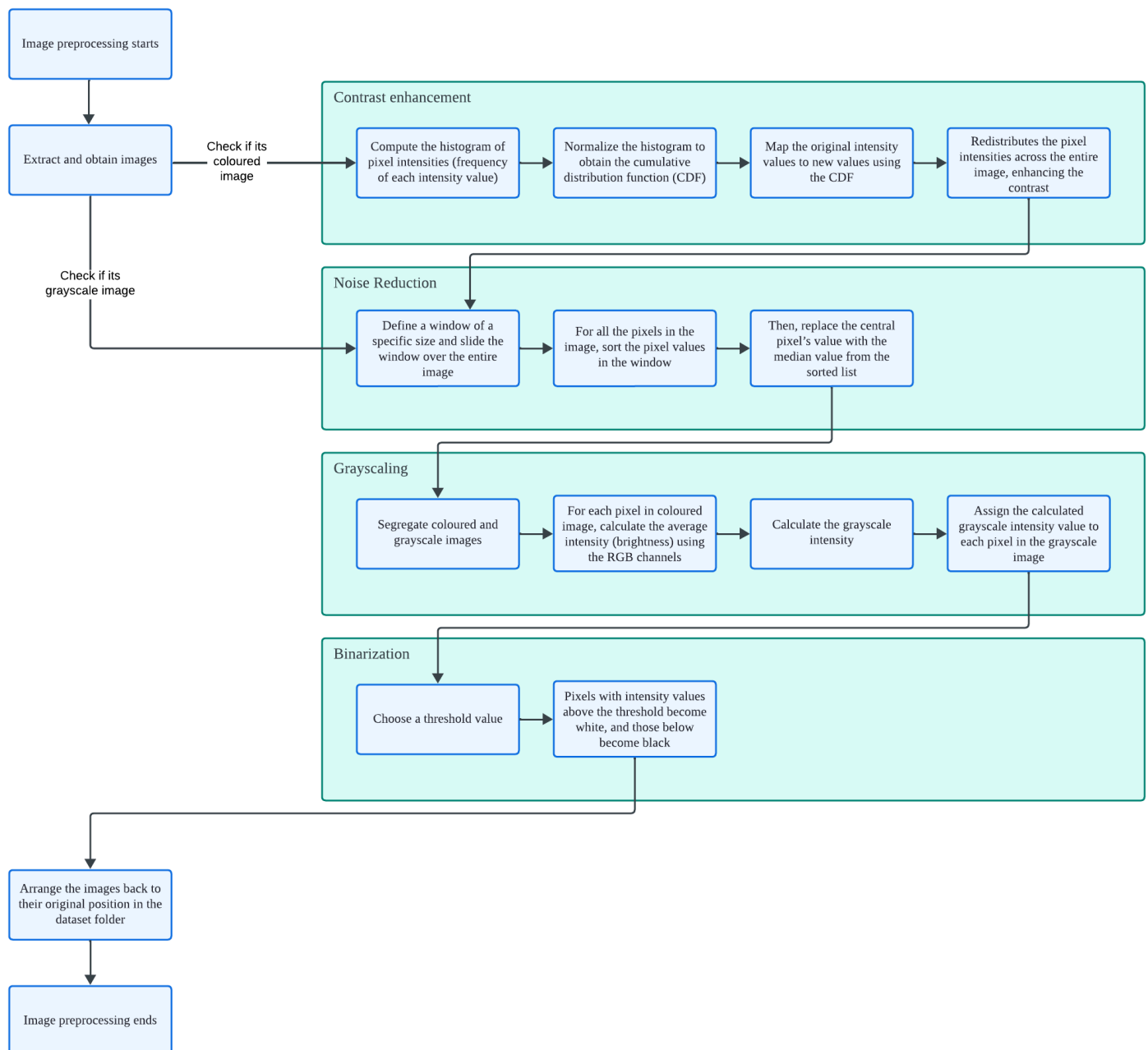


Figure 2. Image Pre-processing Process

The method shown in Figure 2 aims to enhance the image quality and streamline the computational process for better outcomes from the HTR model (KUMAR, 2021). Contrast enhancement techniques improve image quality and highlight specific features, such as handwritten text, to enable more effective analysis (Mescioglu, 2022). Noise reduction allows the removal of unwanted noise from images (Patel, 2023). Grayscaleing and binarization convert colored images to grayscale and then further grayscale images to black and white. Grayscale simplifies algorithms and reduces computational requirements while binarization creates a clear contrast between text and background which helps the HTR model to identify regions of interest (Why We Should Use Gray Scale for Image Processing, n.d.).

Once the images have been processed, the three csv files will be processed as well to clean the transcribed handwritten names, which are the ground truth of the dataset, before using them to verify the results produced by the HTR model. The CSV files have 2 columns which are the image filename and the transcribed handwritten names of the image. These 4 crucial image pre-processing steps ensure

consistency, reduce computational complexity, and improve the efficiency of image processing (Python | Grayscale of Images Using OpenCV, 2019).

CSV files Pre-processing

Any missing values, special characters, punctuation marks, or unwanted symbols in the second column of the CSV files will result in the whole row of the CSV files to be removed and the transcribed text will be verified and converted to uppercase before being inputted into the database to ensure all information is in capital block letters.

Data Post-processing

Once the images have gone through the HTR model, before the patient information data is entered into the database, the text would be converted to uppercase to ensure consistency and ease of searching. Queries for patient names or other relevant details can be case-insensitive, allowing the database to uphold uniformity in the stored data.

Data Analysis

Methodology

The analysis performed here does not rely on the preprocessed dataset, but rather on the original raw data of the dataset. The main purpose for this section is to acquire insight into the features and qualities of the dataset. Our approach to collecting, analyzing, and interpreting data is devised to uphold the reliability and precision of our results. First, the data is collected from kaggle and then thoroughly cleaned and prepared using Python, utilizing libraries like pandas for manipulating data and NumPy for numerical tasks. The graph shown below was created with Python code in Jupyter using the Seaborn and Matplotlib library for visualization. Information on the dataset was extracted with Python using various libraries like cv2, pandas, and zipfile (FIT3163_DataAnalysisReport_Code.pdf, n.d.).

Characteristics of Dataset

The directory within the dataset Zipped folder has 3 csv files and 3 folders within the main directory. The test_v2 folder consists of a folder called test with image files for testing the HTR model, the train_v2 folder includes a folder named train with image files for training the HTR model, and the validation_v2 folder has a validation folder with image files for validating the HTR model. The 3 csv files in the main directory are written_name_test_v2.csv, written_name_train_v2.csv and written_name_validation_v2.csv.

Testing Dataset	Training Dataset	Validation Dataset
File: written_name_test_v2.csv FILENAME IDENTITY count 41370 41300 unique 41370 20279 top TEST_0001.jpg THOMAS freq 1 227	File: written_name_train_v2.csv FILENAME IDENTITY count 330961 330396 unique 330961 100539 top TRAIN_00001.jpg THOMAS freq 1 1825	File: written_name_validation_v2.csv FILENAME IDENTITY count 41370 41292 unique 41370 20227 top VALIDATION_0001.jpg THOMAS freq 1 219

.describe() was used to obtain the output as shown in the table above (FIT3163_DataAnalysisReport_Code.pdf, n.d.). The table above shows that the first testing dataset to be used is TEST_0001.jpg featuring the handwritten word THOMAS. From the results, it is apparent that there are 20279 distinct IDENTITY in the testing dataset. The word 'THOMAS' has occurred 227 times in the testing dataset.

The table above also shows that the first training dataset to be used is TRAIN_0001.jpg which also contains the handwritten word THOMAS. The findings indicate that the training dataset contains 100539 unique IDENTITY. The term 'THOMAS' has been seen 1825 times in the training data.

Last but not least, the information presented in the table shows that the first validation dataset chosen is VALIDATION_0001.jpg, which also has the handwritten word THOMAS. It can be observed from the results that there are 20227 distinct IDENTITY in the validation dataset. The word 'THOMAS' has a frequency of occurrence of 219 in the validation dataset.

Missing Values

The number of missing values can be known using the 3 csv files provided with the help of Python (FIT3163_DataAnalysisReport_Code.pdf, n.d.). It was discovered that there are indeed missing values present in the dataset as shown in the table below:

Filename	Total number of image files	Total number of missing values
written_name_test_v2.csv	41370	70
written_name_train_v2.csv	330961	565
written_name_validation_v2.csv	41370	78

The testing, training, and validation datasets contain many image files, with the training dataset having the most image files. This is because the training dataset is essential for the model to learn patterns, features, and relationships from the input data images. The total number of missing values displayed on the table is not based on the number of missing image files but on the missing annotation on the image files. Other than that, images with the annotation of "UNREADABLE" will be removed from the training, testing and validation dataset. Images annotated with "UNREADABLE" is enumerated as below in table format (FIT3163_DataAnalysisReport_Code.pdf, n.d.):

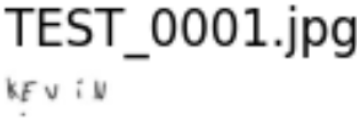
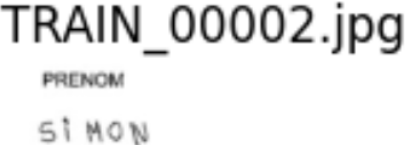

Filename	Original number of image files	"UNREADABLE" image files
written_name_test_v2.csv	41370	11
written_name_train_v2.csv	330961	102
written_name_validation_v2.csv	41370	12

Hence, the table below shows the number of image files with annotations and not labelled as "UNREADABLE" that will be used for the actual model training, testing and validation (FIT3163_DataAnalysisReport_Code.pdf, n.d.):

Filename	Original number of image files	Image files that will be used
written_name_test_v2.csv	41370	41289
written_name_train_v2.csv	330961	330294
written_name_validation_v2.csv	41370	41280

Data Inconsistency and Bias

There are no inconsistencies in the dataset regarding image file formats, as all images are saved in JPG format, so no additional effort is needed to convert the images to a different format. However, the image files have a different naming format for testing and validation dataset images with training dataset images due to training dataset having been allocated a higher amount of image files to train the model (FIT3163_DataAnalysisReport_Code.pdf, n.d.).

Testing Dataset	Training Dataset	Validation Dataset
		

Hence, all validation and testing image files would have to be renamed to have an additional 0 within the file name like from TEST_0001.jpg to TEST_00001.jpg instead. On the other hand, the team may have saved the images in different sizes and resolutions because of the varying sizes of the handwritten text and the variety of equipment used to capture it, which leads to the team being required to address these data inconsistencies.

The dataset being analyzed shows little bias, mainly because it was collected in a public setting related to charity work. This setting guarantees a varied display of handwritten examples, as it includes input from a diverse group of people. The diversity of handwriting styles from each individual adds variation to the dataset, decreasing the potential for bias compared to a more uniform or tightly controlled sample. Diversity plays a vital role in the creation of strong HTR models by offering a wide range of handwriting styles, ultimately improving the model's capacity to perform well with new data. The all-encompassing data collection process, which records the handwriting of various individuals, enhances the dataset's usefulness and dependability for HTR purposes.

Data Distribution

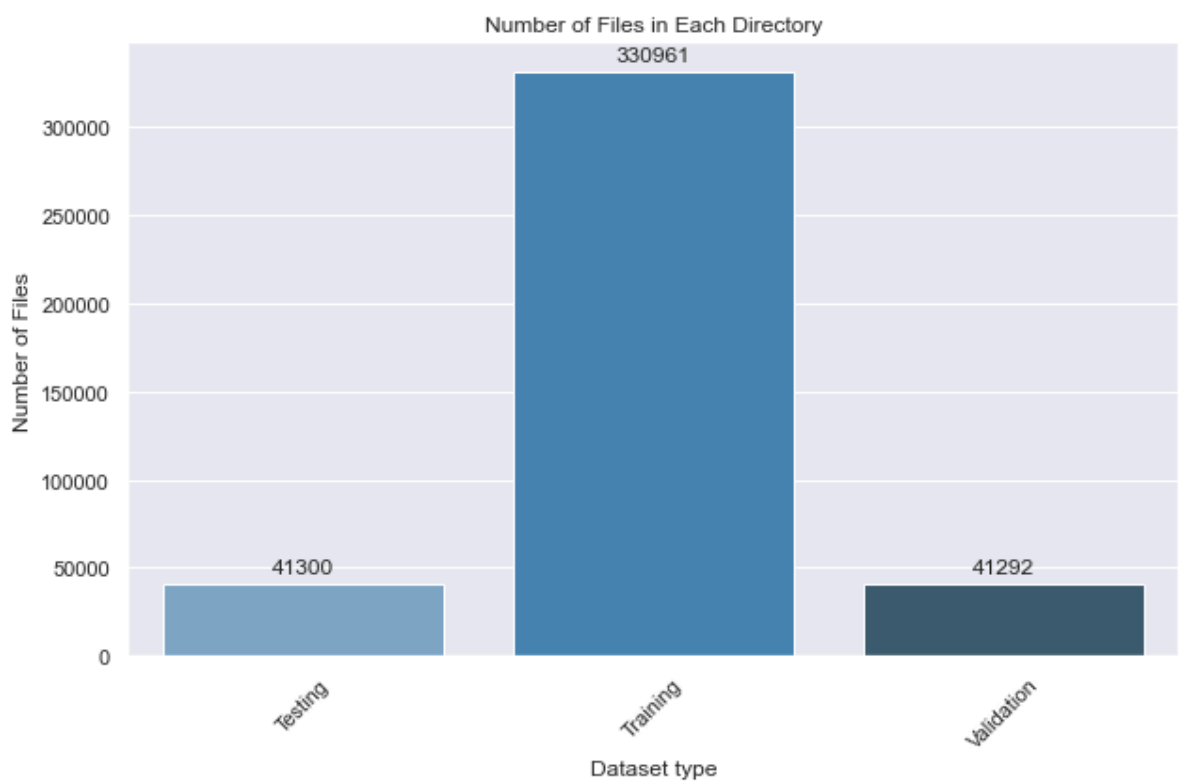


Figure 3. Distribution of Dataset Size

The bar plot as depicted in Figure 3 above shows the breakdown of the dataset on the total number of image files in each training, validation and testing dataset. Based on the bar plot observed above in Figure 3, it can be concluded that the training dataset has a higher amount of images that serves as training data to build the base of the HTR model after removal of missing values.

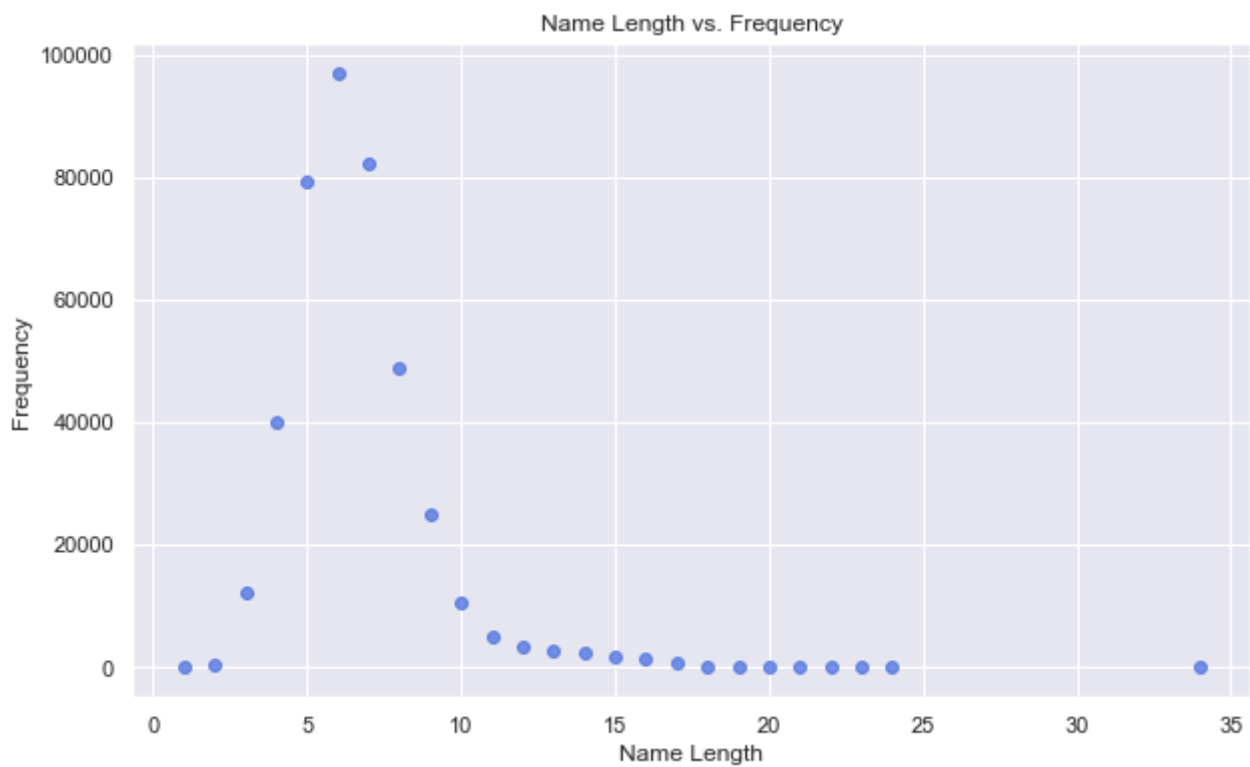


Figure 4. Distribution of Variation in Name Length

The frequency of name lengths is depicted in Figure 4 using a Scatter plot, which includes names ranging from the lowest of 1 character to the longest of 34 characters in length. The scatter plot in Figure 4 indicates that the majority of names have a length of approximately 6 characters.

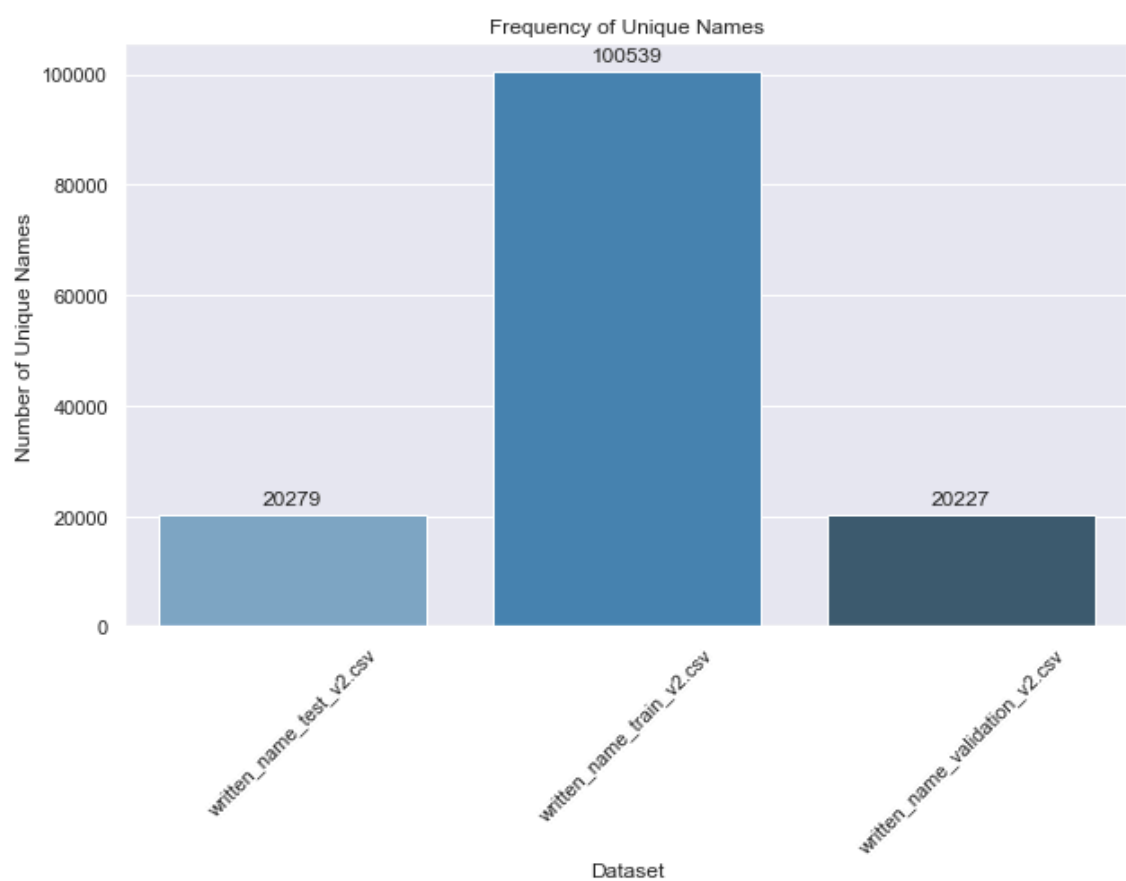


Figure 5. Distribution of Unique Names

Figure 5 above illustrates the distribution of unique names that is present within each testing, training and validation dataset. According to the bar graph in Figure 5, it can be inferred that the distribution of unique names frequency is comparable to the dataset size distribution in Figure 3, where the training dataset contains more images than the testing and validation datasets.

Results and Findings

Figure 3 displays a bar graph showing the overall count of image files within each dataset section, including training, validation, and testing. The largest number of images in the training dataset forms the basis for constructing the HTR model. Having a bigger training dataset enables the model to learn various patterns and features better, improving its capability to generalize to new data, therefore, there are more image files in the training dataset for HTR model training.

Figure 4's scatter plot displays the frequency of name lengths between 1 and 34 characters, with the majority of names in the dataset being around 6 characters in length. Knowing the average length of names aids in creating suitable model structures and managing sequences of different lengths when recognizing text.

Figure 5 depicts how unique names are spread out among the datasets, and the frequency of unique names, as shown in the figure, matches the distribution of dataset sizes in Figure 3. This suggests that a higher number of unique names in the training dataset leads to greater exposure to different name variations, enhancing the model's resilience.

Conclusion

It is important to mention that through image pre-processing and training the HTR model with a diverse range of images dataset, the HTR can handle more robust data. The wide range of images enables the model to gain knowledge from different writing styles, backgrounds, and levels of noise levels. As a result, the HTR model is able to improve its ability to identify handwritten text in various situations, even when dealing with difficult conditions like different lighting.

Furthermore, incorporating a variety of images can help reduce potential biases stemming from a small dataset. Exposing the model to different writing styles helps to minimize the risk of bias towards certain groups of writers and generating inaccurate results for marginalized samples.

Overall, the HTR model's performance and adaptability benefit greatly from thorough pre-processing and a varied training dataset.

Software Specification

Software Framework

For the web-based health information system, MEAN and MERN stack frameworks are taken into consideration as a full-stack development framework in creating the project web application. These stacks boost development efficiency as it compounds compatible tools that work well together, pre-equipped with libraries, frameworks, and modules that simplify common tasks, allowing developers to focus on the application's logic and content (Stewart, 2023). In addition, as all components of these stacks support applications written in JavaScript, the application can be written in one language for both server-side and client-side execution environments. This accelerates the development process and enhances cohesiveness in building the application.

The MEAN stack is a free and open-source JavaScript software stack for building dynamic websites and web applications. It comprises MongoDB, Express.js, Angular and Node.js with each component playing a crucial role in the development process, enabling seamless development of both the frontend and backend of a website using Javascript. While MERN stack is a variation of MEAN stack with the main difference being the use of React.js instead of Angular framework.

MongoDB will be the database system, serving as a NoSQL open-source document-oriented database that efficiently stores data in JSON-like documents (MongoDB, n.d.). Express.js is a back-end web application framework for Node.js that provides a robust set of features that simplifies the creation of server-side functionalities such as the interaction between the frontend and the database (Erickson , 2024). Node.js is the back-end runtime environment for event-driven server-side and networking applications in coordinating communication between front-end and back-end components (Erickson , 2024). While for Angular and React.js, both are front-end components used to build user interfaces. Both technologies have many similarities with them having component-based architecture, being open source and having large community support offering extensive resources for learning and troubleshooting (Powell, 2023).

The table below compares the difference between the two popular front-end technologies (Powell, 2023).

Parameter	Angular	React.js
Type	TypeScript-based JavaScript	JavaScript library

	framework.	
Data binding	Two-way data binding	One-way data binding
DOM Type	Real DOM	Virtual DOM
Testing and debugging	Complete solution within a single tool	Needs an additional set of tools

The table below compares MEAN and MERN stack (Bhagat, 2023).

Parameter	MEAN Stack	MERN Stack
Popularity	More established	Growing in popularity
Learning Curve	Easier to learn	Steep for beginners
Performance	Better server-side performance	Fast UI rendering
Scalability	Better for larger and more complex apps	Good for small to medium-sized apps
Community Support	Large and established community	An active, growing community
Third-Party Support	Offers different ready to use features	Requires additional libraries for supporting similar requests

After much discussion and in depth research and comparison between the two stacks, my team concluded that the MEAN stack would be the optimal choice for our project. This is because half of our team members have prior knowledge and hands-on experience with the MEAN stack for web development, acquired from the FIT2095 unit. With that, leveraging existing expertise is expected to streamline the development process as it minimises the time spent on learning new technologies and maximises productivity.

Moreover, MEAN stack approachability is advantageous for members who are unfamiliar with it, as it's easier to grasp compared to the MERN stack with its steep learning curve for beginners. This ensures that all team members can quickly become proficient in MEAN stack, thereby enhancing the team capability without significant time investment in training. Additionally, MEAN stack has a better server-side performance and is well-suited for handling the complexities of a large scale application. This is particularly relevant for our web-based healthcare system that calls for intricate data management and processing capabilities.

Software libraries

Software Libraries	Justification
--------------------	---------------

TensorFlow.js	TensorFlow.js is a machine learning library in Javascript that will be used to develop our handwritten text recognition model. It is chosen as it is used for training and deploying machine learning models in the web browser and in Node.js (TensorFlow.js, n.d.), which fit our project development design, where the model will be integrated in our web-based health information system.
Keras	Keras is a high-level neural network API written in Python that's integrated into TensorFlow (Simplilearn, 2024). It is chosen as it is an open-source library that simplifies the creation and training of deep learning models.
OpenCV	OpenCV is an open-source computer vision library for image processing and machine learning (GeeksforGeeks, 2024). It is mainly used to process image datasets to identify the handwritten text.
Pandas	Pandas is a Python library used for data manipulation and basic data analysis, where its functions allow data cleaning, exploration, and visualization to be done easily (Lee, 2022). Additionally, allowing display of data in a way that is easily understood.
Matplotlib	Matplotlib is a plotting library that works well with Pandas to plot a wide variety of graphs, which help further understand the underlying data trends and patterns.
Zipfile	Zipfile library is used for reading and writing ZIP files (Python, 2023). With that, it is useful when dealing with external datasets that are distributed in compressed formats, where data needs to be extracted from the zip files.
Seaborn	Seaborn is a data visualization library built on top of Matplotlib to draw attractive and informative statistical graphs (Seaborn: Statistical Data Visualization — Seaborn 0.13.2 Documentation, n.d.).
NumPy	NumPy is a Python library that serves as an important key tool for machine learning with many useful functions for performing mathematical calculations (Mahto, 2021).

Justification summary

After much research, the team concluded that the listed software libraries will be the main libraries used in the project in developing the handwritten text recognition model. Using these combinations of libraries that are specialised for machine learning, image processing and data manipulation, such as TensorFlow.js, OpenCV and Pandas, will help provide a strong foundation for the model development as we currently have limited knowledge and experience in developing such a model. Some libraries listed are said to be beginner friendly such as Keras where it's easy to learn and use, in hopes that the team will be able to grasp the usage quickly.

Database system

MongoDB, which will be our database system, serves as a NoSQL open-source document-oriented database that efficiently stores data in JSON-like documents with flexible schemas (MongoDB, n.d.). This

database system is suitable for our project usage in storing and managing healthcare data as MongoDB efficiently handles large volumes of data and high velocity of insertion, which aligns with the needs of the health information system database being scalable. On top of that, with data privacy and security being important in healthcare, MongoDB provides many security features that ensure compliance with the privacy and security of data by implementing user authentication in the web application and using MongoDB's Role-Based Access Control (RBAC) for authorization to control access to data and many other security features (MongoDB Developer Data Platform With Strong Security Capabilities, n.d.).

Moreover, MongoDB is open source and free to use with an abundance of available documentation, making it easy to set up and use for developers (Adalyn, 2023). Additionally, MongoDB has a free user-oriented tool known as MongoDB Compass that enables database interaction using a graphical user interface that allows users to analyze their data in a visual environment (Adalyn, 2023). Furthermore, the team is acquainted with MongoDB in terms of its usage, with two members having experience in adopting MongoDB for backend development.

Operating software

Our web-based system will be a cross-platform system that is compatible with Windows, MacOS, and Linux for desktop operating systems as well as Android and IOS for mobile operating systems. This broad compatibility guarantees that healthcare facilities utilizing any of these operating systems can access our system effortlessly, fostering a consistent user experience across various platforms. These desktop operating systems are well suited for healthcare environments as they are reliable and secure with the capability to handle busy workloads (Weakley, 2022). According to Statcounter Global Stats report, Windows has the highest user base with 72.52% market share for desktop operating systems, followed by MacOS with 14.68% and Linux with 4.05% making Windows the most popular OS globally (StatCounter Global Stats, 2024). As for mobile operating systems, Android has a whopping 70.79% market share for mobile operating systems followed by IOS with 28.46% (StatCounter Global Stats, 2024). These popular operating systems present a large potential user base for our health information system to be utilised within the healthcare industry.

Programming Language environment

Visual Studio Code

Visual Studio Code is a flexible source code editor with powerful developer tools like built-in support for debugging, IntelliSense code completion, formatting, code navigation, refactoring, and version control. Our team chose Visual Studio Code as our primary code editor in implementing the project as the team has extensive experience in using this code editor for various developments, including web applications. Moreover, the flexibility and customizability of the code editor through tons of extensions of features that we need are highly favored as the web-based health information system and the handwritten text recognition model will be built in the same code editor. With that, specific software tools are needed for our project development such as Tensorflow and Postman.

Furthermore, Visual Studio Code is tailored for building cross-platform web and cloud applications with Microsoft Azure easing the process of deploying and hosting web applications built with Angular, Node, and many others, all from within VS Code (Visual Studio Code, 2021). Besides that, Visual Studio Code supports various programming languages, including JavaScript, HTML/CSS, Python, and other additional coding languages through available extensions on the VS Code Marketplace, covering the language needed for our project. VS Code has Git commands allowing developers to manage version control easily,

from examining differences, preparing files for commit, finalizing commits, and push and pull from any hosted source control management service directly from within the code editor (Visual Studio Code, 2021).

Jupyter Notebook

Jupyter Notebook is a web-based notebook environment tailored for interactive computing (IBM Watsonx as a Service, n.d.-b) that we use heavily for data exploration and preprocessing purposes, which is an essential part of our data workflow. Jupyter Notebook can also be used for machine learning experimentations and modeling (What Is Jupyter Notebook? | Domino Data Lab, n.d.), which we might use during the exploration of implementing the handwritten text recognition model. The team chose Jupyter Notebook as it provides a flexible and efficient way to manage the iterative exploration process frequently encountered in data analytics and machine learning (Gunn, 2023). Moreover, with its user-friendly interface and the team's familiarity with this notebook environment, it will reduce the need for the team to learn new software, thus helping streamline our project’s progress.

Project management tool

The table below lists the tools used for project management and collaboration.

Project Management Category	Tools	Justification
Communication	WhatsApp	WhatsApp, which is an instant messaging app, will be used as the team's primary communication channel, for quick and immediate discussions, including scheduling meetings, individual day-to-day work progress and any other relevant information.
	Discord	Discord, which is also an instant messaging platform, will be used as a secondary communication channel with the team and the supervisor, for prompt responses to general project inquiries and to schedule meetings.
	Zoom	Zoom, which is a video audio communication platform, will be mainly used to conduct weekly online meetings with the team as well as with the supervisor for project discussions and consultations. Weekly or biweekly meetings are conducted to ensure the team and supervisor are up to date with the project's current progress to ensure the project is on the right track and not astray.
	Email	Email will be the team formal communication method in communicating with our supervisor.
Project Management	GitLab	GitLab, which is a free web-based Git repository, will be used as our version control system to manage and track changes made in our project source code files. GitLab allows us to collaborate and coordinate our work without affecting each

		other's progress. It also helps monitor the team's individual progress.
	GoogleDrive	GoogleDrive, which is a cloud-based storage service, will be used to store any project-related work files and relevant documentation that the team can easily access online through the browser.
	Figma	Figma, which is a collaborative UI design web tool, will be used to help design, prototype and visualise the UI for our web-based health information system. Through Figma, it will help us understand and pinpoint the components needed for the system.
	LucidChart	LucidChart, which is a web diagramming application, will be used to create our system and model workflow, to help visualise, improve and understand our system process.
	ProjectLibre	ProjectLibre, which is a free project management software, will be used to help manage projects, in terms of task management, resource allocation, work breakdown structure and visualise project schedule via Gantt Chart.
	Trello	Trello, which is a web-based collaboration tool, will be used to help track our project tasks as well as manage and monitor our team progress at a glance.

Justification summary

The listed communication tools were chosen as the team is familiar with its usage and due to their user friendliness. Moreover, they are easily accessible through various devices, ensuring constant efficient communication throughout the project lifecycle. These communication tools play a vital part in communicating with our supervisor and the team as it ensures a smooth progress throughout the project development with minimum miscommunication.

While for the project management part, the tools listed are chosen as they are useful and important to keep the project on track with the project schedule and tasks workload. These tools ensure the team is not lagging behind, reducing the risk of project failure. All the tools listed are free and easily accessible by the team, allowing ease in collaboration and ensuring mutual understanding of the project end product and goals amongst the team members.

Hardware Specification

Hardware Component	Recommended Hardware Specification
CPU	Intel i7/i5 Gen 9 and above
RAM	DDR4 16 GB

GPU	Integrated Graphics (Windows)/Nvidia GeForce RTX 3050
Storage	512 GB SSD
Camera	Phone Camera

CPU

CPU is important for handling demands of machine learning tasks in handwritten recognition projects. Multi cores of these processors can execute numerous tasks simultaneously that increases computational efficiency. This capability is essential for tasks involving large datasets and numerous image processing tasks. Therefore, prioritizing CPUs with multiple cores and higher clock speeds is essential for achieving faster training times and smoother inference in handwritten recognition projects. We decided an Intel i7/i5 processor should suffice.

RAM

RAM is essential for storing and processing data in machine learning tasks. It temporarily holds datasets and computational tasks during model training and inference. Sufficient RAM ensures efficient handling of large datasets, smoother model training and faster inference performance. Based on the dataset size and algorithms used for processing images and handwritten recognition, we decided that a DDR4 16GB RAM is a safe amount to have.

GPU

Integrated graphics provide basic graphical capabilities suitable for most image processing associated with handwritten recognition. These integrated solutions can efficiently handle tasks such as image preprocessing but may have limitations when it comes to more complex graphical representations. Most of our laptops already have integrated graphics which serve us well during the initial development and testing phases of our handwritten recognition project.

On the other hand, having a dedicated GPU like the Nvidia GeForce RTX 3050 enhances graphical performance and accelerates computing tasks required in machine learning. It can significantly boost the speed of model training, especially deep learning algorithms and complex image processing tasks. This will come in handy when we encounter computational intensive tasks or larger datasets enabling faster processing and improved performance overall.

Storage

Storage is essential to our project to keep our code and project related files. SSD is selected as it offers faster reading and writing speeds compared to HDD. This speed will lead to faster loading times, quicker model training and smoother overall performance. 512 GB storage space is a good amount that we will not have to worry about lack of data space. On top of this, our project files are uploaded to Google Drive and Github to serve as a reliable backup for our project. Furthermore, all group members are able to access it at any time. This will be further explained in the Software specifications section.

Camera

Any phone camera is fine as long as it has high resolution and is able to scan handwritten text suitable for our project. The key requirement is that the camera is able to capture clear and detailed images of handwritten text enabling accurate recognition and optimal image processing for our handwritten

recognition model. Cameras with features like auto-focus and low-light performance enhance the quality and versatility of document scanning in various conditions.

Network Connection

Stable and fast network connectivity such as Monash Wifi Eduroam is essential for researching, referencing, data retrieval, and collaboration. Reliable network connectivity ensures team members are able to access online resources, databases and research articles without interruptions. Furthermore, allowing us to communicate smoothly between each other allows virtual meetings and discussions. This connectivity also allows efficient data access through platforms like Google Drive and Github that are shared among group members. It also synchronizes each other's work in shared environments within the project management tool section which increases effective collaboration and version control.

References

- Handwriting Recognition. (n.d.). [www.kaggle.com.
https://www.kaggle.com/datasets/landlord/handwriting-recognition](https://www.kaggle.com/datasets/landlord/handwriting-recognition)
- Patel, M. (2023, October 23). The Complete Guide to Image Preprocessing Techniques in Python. Medium. <https://medium.com/@maahip1304/the-complete-guide-to-image-preprocessing-techniques-in-python-dca30804550c>
- Mescioglul, B. (2022, November 10). Image Processing Tutorial Using scikit-image — Contrast Enhancement. Medium. <https://medium.com/@betulmesci/image-processing-tutorial-using-scikit-image-contrast-enhancement-519232716c87>
- KUMAR, B. (2021, February 10). Image Preprocessing - Why is it necessary? Spider. <https://medium.com/spidernitt/image-preprocessing-why-is-it-necessary-8895b8b08c1d>
- why we should use gray scale for image processing. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/12752168/why-we-should-use-gray-scale-for-image-processing>
- Why to use Grayscale Conversion during Image Processing? (n.d.). Wwww.isahit.com. <https://www.isahit.com/blog/why-to-use-grayscale-conversion-during-image-processing>
- Python | Grayscale of Images using OpenCV. (2019, April 15). GeeksforGeeks. <https://www.geeksforgeeks.org/python-grayscale-of-images-using-opencv/>
- Brook, C. (2023, May 8). What is a Health Information System? Digital Guardian. <https://www.digitalguardian.com/blog/what-health-information-system>
- Image Data Collection in 2023: What it is and Best Practices. (n.d.). Research.aimultiple.com. <https://research.aimultiple.com/image-data-collection/>
- 5 Common Issues and Challenges in Digital Image Processing. (2019, June 7). Eminenture Blog - Research, Data, Technology & Marketing. <https://www.eminenture.com/blog/5-common-issues-with-image-processing/>
- FIT3163_DataAnalysisReport_Code.pdf. (n.d.). Google Docs. Retrieved April 30, 2024, from <https://drive.google.com/file/d/1JOHvq9g5Je6uB8gekz9fntdfIAEaQ-yJ/view?usp=sharing>

Rosebrock, A. (2018, July 19). OpenCV Tutorial: A Guide to Learn OpenCV. PyImageSearch. <https://pyimagesearch.com/2018/07/19/opencv-tutorial-a-guide-to-learn-opencv/>

kulhary, R. (2019, September 23). OpenCV - Overview. GeeksforGeeks. <https://www.geeksforgeeks.org/opencv-overview/>

python - Import OpenCV on jupyter notebook. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/52832991/import-opencv-on-jupyter-notebook>

How to import cv2 in python3? (n.d.). Stack Overflow. Retrieved April 30, 2024, from <https://stackoverflow.com/questions/46610689/how-to-import-cv2-in-python3>

MongoDB Developer data platform with strong security capabilities. (n.d.). MongoDB. <https://www.mongodb.com/products/capabilities/security#authentication>

MongoDB. (n.d.-b). Why use MongoDB and when to use it? <https://www.mongodb.com/why-use-mongodb>

Adalyn, N. (2023, August 2). The Benefits of using a NoSQL database like MongoDB for your Backend Development. <https://www.linkedin.com/pulse/benefits-using-nosql-database-like-mongodb-your-backend-nav-adalyn>

Desktop Operating System Market Share Worldwide. StatCounter Global Stats. (2024, March). <https://gs.statcounter.com/os-market-share/desktop/worldwide>

Mobile Operating System Market Share Worldwide. StatCounter Global Stats. (2024, March). <https://gs.statcounter.com/os-market-share/mobile/worldwide#monthly-202211-202311>

Weakley, L. (2022, June 28). Best Operating Systems for Healthcare Facilities - Healthcare Salary World. <https://healthcaresalaryworld.com/best-operating-systems-for-healthcare-facilities/>

Maitray-Gadhavi. (2023, December 18). Full-Stack vs MEAN Stack vs MERN Stack: The Right Technology Stack for You in 2024. Radixweb. <https://radixweb.com/blog/full-stack-vs-mean-stack-vs-mern-stack-development#difference>

Bhagat, V. (2023, April 15). MEAN vs MERN: The Ultimate guide to selecting the right stack. PixelCrayons. <https://www.pixelcrayons.com/blog/dedicated-teams/mean-vs-mern/#:~:text=MEAN%20Stack%20%E2%80%93%20An%20Overview,-The%20MERN%20stack&text=js,the%20client%20and%20server%20sides.>

Powell, Z. (2023, December 29). Angular vs react: A detailed side-by-side comparison. Kinsta®. <https://kinsta.com/blog/angular-vs-react/>

Stewart, E. (2023, August 25). What is a tech stack? definition, types, benefits. What is a Tech Stack? Definition, Types, Benefits | Enterprise Tech News EM360. <https://em360tech.com/tech-article/tech-stack-definition>

Erickson , J. (2024, January 30). What is The mern stack?. What is the MERN Stack? Guide & Examples | Oracle Malaysia. <https://www.oracle.com/my/database/mern-stack/>

Visual Studio Code - Code editing. Redefined. (2021, November 3). <https://code.visualstudio.com/>

- Quoy, L. (2024, March 19). Visual Studio vs Visual Studio Code: What's the Key Difference? DistantJob - Remote Recruitment Agency.
<https://distantjob.com/blog/visual-studio-vs-visual-studio-code/#:~:text=The%20main%20difference%20between%20Visual,an%20Extension%2Dbased%20Code%20Editor>.
- Kelley, K. (2024, February 15). What is GitLab and How to Use It? Simplilearn.com.
https://www.simplilearn.com/tutorials/git-tutorial/what-is-gitlab#what_is_git
- GeeksforGeeks. (2024, April 15). What is OpenCV Library? GeeksforGeeks.
<https://www.geeksforgeeks.org/opencv-overview/>
- TensorFlow.js | Machine learning for JavaScript developers. (n.d.). TensorFlow.
<https://www.tensorflow.org/js>
- Simplilearn. (2024, February 15). What is Keras: The best Introductory Guide to Keras. Simplilearn.com.
https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-keras#what_is_keras
- Lee, A. (2022, August 27). Why and How to Use Pandas with Large Data - Towards Data Science. Medium.
<https://towardsdatascience.com/why-and-how-to-use-pandas-with-large-data-9594dda2ea4c>
- seaborn: statistical data visualization — seaborn 0.13.2 documentation. (n.d.).
<https://seaborn.pydata.org/#:~:text=Seaborn%20is%20a%20Python%20data,attractive%20and%20informative%20statistical%20graphics>.
- Python, R. (2023, March 1). Python's zipfile: Manipulate Your ZIP Files Efficiently.
<https://realpython.com/python-zipfile/>
- Mahto, P. (2021, December 15). NUMPY for Machine Learning - MLPoint - Medium. Medium.
<https://medium.com/mlpoint/numpy-for-machine-learning-211a3e58b574#:~:text=NumPy%20library%20is%20an%20important%20foundational%20tool%20for%20studying%20Machine,c%20an%20be%20performed%20using%20NumPy>.
- Babitz, K. (2023, May 30). Introduction to Plotting with Matplotlib in Python.
<https://www.datacamp.com/tutorial/matplotlib-tutorial-python>
- Wikipedia contributors. (2024, April 28). Figma. Wikipedia. <https://en.wikipedia.org/wiki/Figma>
- What is Trello? | Trello | Atlassian Support. (n.d.). Atlassian Support.
<https://support.atlassian.com/trello/docs/what-is-trello/>
- Gunn, E. (2023, February 3). Jupyter Notebook Tutorial [Data Analytics for Beginners]. CareerFoundry.
<https://careerfoundry.com/en/blog/data-analytics/jupyter-notebook-tutorial/#:~:text=Because%20Jupyter%20notebooks%20support%20code,data%20analytics%20and%20machine%20learning>.
- What is Jupyter Notebook? | Domino Data Lab. (n.d.).
<https://domino.ai/data-science-dictionary/jupyter-notebook>
- IBM watsonx as a Service. (n.d.-b).
<https://www.ibm.com/docs/en/watsonx/saas?topic=models-build-model-in-jupyter-notebook>
- Ruyu Bai, Xiaoli Wang and Qiang Su, "The impact of healthcare information technology on quality and safety of healthcare: A literature review," 2015 12th International Conference on Service Systems

and Service Management (ICSSSM), Guangzhou, China, 2015, pp. 1-4, doi: 10.1109/ICSSSM.2015.7170274

Shine, K. I. (1996). Impact of information technology on medicine. *Technology in Society*, 18(2), 117–126. [https://doi.org/10.1016/0160-791x\(96\)00004-8](https://doi.org/10.1016/0160-791x(96)00004-8)

Acknowledgement

I acknowledge the use of Microsoft Copilot (<https://copilot.microsoft.com/>) to generate materials for background research and self-study in the process of completion of this assessment. I entered the following prompts on 23 April 2024:

- Explain and describe the basic operations to deal with huge zip file in Python
- Common pre-processing steps for images
- Explain what is the meaning of 'Methodology'
- Explain on cv2
- How to import cv2 in python3
- Explain what is problem statement and opportunity statement

The generated output from the artificial intelligence was adapted, modified and used for the final response.