Handwritten Text Recognition

A PROJECT REPORT

Submitted by

Mehul Kurkute (19CP205)

In partial fulfillment for the award of the degree of

B. TECH. in COMPUTER ENGINEERING

4CP33: Full Semester External Project (FSEP)

BIRLA VISHVAKARMA MAHAVIDYALAYA
(ENGINEERING COLLEGE)

(An Autonomous Institution)

VALLABH VIDYANAGAR

Affiliated to

GUJARAT TECHNOLOGICAL UNIVERSITY, AHMEDABAD

Project Carried out at

ISO 9001:2008
ISO 27001:2013
CMMI LEVEL-5

Bhaskaracharya National Institute for Space Applications & Geo-informatics
Ministry of Electronics and Information Technology, Govt. of India.

Gandhinagar

Academic Year: 2022 – 2023

BVM ENGINEERING COLLEGE, VALLABH VIDYANAGAR-388120

APPROVAL SHEET

The project work entitled **“Handwritten Text Recognition”** carried out by **Mehul Kurkute (19CP205)** is approved for the submission in the course **4CP33, Full Semester External Project** for the partial fulfillment for the award of the degree of B. Tech. in Computer Engineering.

Date:

Place: Vallabh Vidhyanagar

Signatures of Examiners:

(Names and Affiliations)

CERTIFICATE

This is to certify that Project Work embodied in this project report titled "**Handwritten Text Recognition**" was carried out by **Mehul Mohji Kurkute (19CP205)** under the course **4CP33, Full Semester External Project** for the partial fulfillment for the award of the degree of B. Tech. in Computer Engineering. Followings are the supervisors at the student:

Date:

Place: Vallabh Vidhyanagar

Mr. Punit Lalwani, Mr. Harsh Kirasata	Prof. Kirtikumar J Sharma	Mr. Parimal Solanki
Software developer, BISAG-N, Gandhinagar, Gujarat	Assistant Professor, Computer Engineering Department, BVM Engineering College	Assistant Professor, Computer Engineering Department, BVM Engineering College

Dr. Darshak G Thakore
Prof. & Head,
Computer Engineering Department,
BVM Engineering College

COMPUTER ENGINEERING DEPARTMENT, BVM ENGINEERING COLLEGE, VALLABH VIDYANAGAR-388120

Bhaskaracharya National Institute for Space Applications and Geo-informatics

MeitY, Government of India

Phone: 079 - 23213081   Fax: 079 - 23213091

E-mail: info@bisag.gujarat.gov.in, website: https://bisag-n.gov.in/

ISO 9001:2008
ISO 27001:2013
CMMI LEVEL-5

CERTIFICATE

This is to certify that the project report compiled by ***Mr. Harshad Parghi and Mr. Mehul Kurkute*** *students of 8th Semester* ***B.Tech – Computer Engineering from Birla Vishvakarma Mahavidyalaya, Vallabh Vidyanagar, Gujarat Technological University, Ahmedabad*** *have completed their final Semester internship project satisfactorily. To the best of our knowledge this is an original and bonafide work done by them. They have worked on Machine Learning-based application for "****Handwritten Text Recognition****", starting from January 04th, 2023 to April 30th, 2023.*

During their tenure at this Institute, they were found to be sincere and meticulous in their work. We appreciate their enthusiasm & dedication towards the work assigned to them.

We wish them every success.

Harsh Kirasata

External Co-Guide

BISAG- N, Gandhinagar

Punit Lalwani

CISO,

BISAG- N, Gandhinagar

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this report under the course 4CP33 (Full Semester External Project) and that neither any part thereof nor the whole of the report has been submitted for a degree to any other University or Institution.

I certify that, to the best of my knowledge, the current report does not infringe upon anyone's copyright nor does it violate any proprietary rights and that any ideas, techniques, quotations or any other material from the work of other people included in my report, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the boundary of fair dealing within the meaning of the Indian Copyright (Amendment) Act 2012, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in the current report and have included copies of such copyright clearances to the appendix of this report.

I declare that this is a true copy of report, including any final revisions, as approved by the report review committee.

I have checked write-up of the present report using anti-plagiarism database and it is in permissible limit. However, at any time in future, in case of any complaint pertaining of plagiarism, I am the sole responsible person for the same. I also understand that, in case of such complaints of plagiarism, as per the University Grants Commission (UGC) norms, the University can even revoke the degree conferred to the student submitting such a plagiarized report.

Date:

Institute Code: 007

Institute Name: Birla Vishvakarma Mahavidyalaya (BVM) Engineering College

Mehul Kurkute
19CP205

ACKNOWLEDGEMENT

I would like to express our sincere gratitude to Prof. Kirtikumar J. Sharma and Mr. Parimal Solanki as well as Dr. Narendra M. Patel for their guidance and constant supervision. I would also like to thank our institution for their support towards achieving our goal.

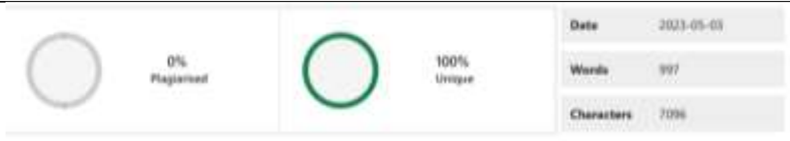
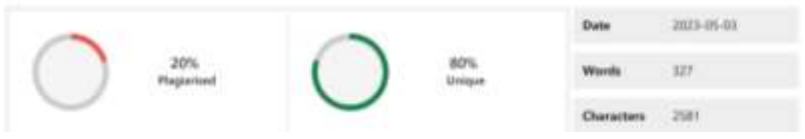
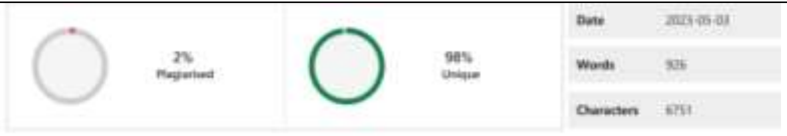
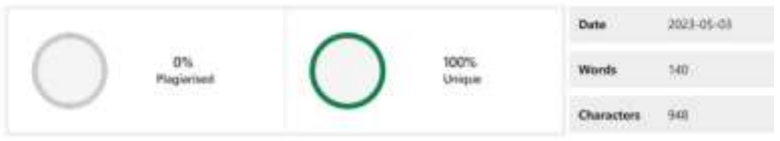
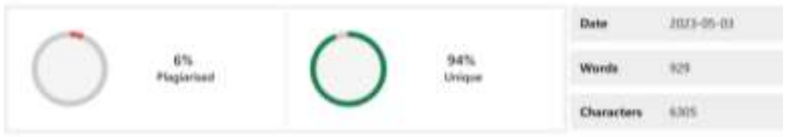
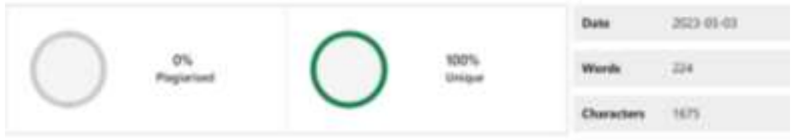
I would like to express our endless thanks to our external guide and Co-Guide Mr. Punit Lalwani & Harsh Kirasata and to Training Cell Mr. Punit Lalwani & Mr. Sidhdharth Patel at Bhaskaracharya National Institute of Space Application and Geo-informatics for their sincere and dedicated guidance throughout the project development.

I would also like to thanks my group partner Harshad Parghi for helping towards the implement project.

I am indebted to Birla Vishvakarma Mahavidyalaya Engineering College and our project guides for providing necessary information regarding the project and for their support in completing the project. With their feedback, I was able to improve our project in various aspects.

Mehul Kurkute
19CP205

Plagiarism Report

Abstract & Introduction	 0% Plagiarised 100% Unique Date 2023-05-03 Words 997 Characters 7096
Existing Solutions and how our approach differs from them	 20% Plagiarised 80% Unique Date 2023-05-03 Words 327 Characters 2581
Related Work and Background.	 2% Plagiarised 98% Unique Date 2023-05-03 Words 926 Characters 6751
Modelling & Design	 0% Plagiarised 100% Unique Date 2023-05-03 Words 140 Characters 948
Implementation	 6% Plagiarised 94% Unique Date 2023-05-03 Words 929 Characters 6305
Future Scope & Conclusions	 0% Plagiarised 100% Unique Date 2023-05-03 Words 224 Characters 1675
Dupli Checker	Tool Used : **Duplichecker** Total Plagiarism : 4.66% Unoqueness of the project:95.33%

Abstract

Handwritten Text Recognizer is designed to convert handwritten text into machine-readable text and translate it from one language to another. It also designed for digitizing handwritten notes, attendance sheets, and checking handwritten multiple-choice question (MCQ) tests.

The project name is "Handwritten Text Recognizer", which helps for language translation, digitizing handwritten notes into PDF format, converting handwritten attendance into an Excel file, and also we can Check Handwritten MCQ by setting the correct MCQ answer with help of answer system will provide marks of particular student.

It also allows for storing result data into a database and filtering and visualizing student results using the stored data.

For developed model used CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network). I was working on CNN for handwriting text/digits recognition. For develop model used MNIST and IAM dataset, For handwritten digits recognition used MNIST dataset and character recognition used IAM dataset.

The applications of project are language translation, digitizing handwritten notes into PDF format, converting handwritten attendance into an Excel file, and Handwritten MCQ test checker. The application of project are divided into two levels among the group partner, digitized attendance sheet and MCQL test checker which are developed by one person/me and rest by other.

TABLE OF CONTENTS

List of Figures

List of Symbols, Abbreviations and Nomenclature

- OpenCV – Open Source Computer Vision Library
- OCR – Optical Character Recognition
- CNN - . Convolutional Neural Network
- RNN- Recurrent Neural Network
- GUI -Graphical User Interface
- GPU-Graphical Processing Unit
- RAM - Random-access memory
- MB - Mega Bytes
- GB - Giga Bytes

1. Introduction

1.1 Introduction to the project

Handwritten text recognition (HTR) is a field of research that focuses on developing algorithms and systems to accurately recognize and convert handwritten text into digital text. It has applications in various areas, such as digitizing historical documents, document digitalization, healthcare and pharmaceutical, digitizing handwritten notes, attendance sheets.

The project aims to convert handwritten text into machine-readable text and translate it from one language to another. It also includes features such as digitizing handwritten notes, attendance sheets, and checking handwritten multiple-choice question (MCQ) tests.

The project name is "Handwritten Text Recognizer", which helps for language translation, digitizing handwritten notes into PDF format, converting handwritten attendance into an Excel file, and also we can Check Handwritten MCQ by setting the correct MCQ answer with help of answer system will provide marks of particular student.

It also allows for storing data into a database and filtering and visualizing student results/marks using the database.

Handwritten written text recognition is a challenging task, but this project efficiently accomplishes it.

Also aims of the project is to contribute to the field of HTR by developing an innovative approach that leverages deep learning techniques to accurately recognize and convert handwritten text into digital text. The project involves creating a large-scale annotated dataset of handwritten text samples, training and evaluating deep learning models using state-of-the-art techniques, and conducting extensive experiments to assess the performance and adaptability of the proposed approach. The ultimate goal is to develop an HWR system that performs with high accuracy, is adaptable to different writing styles, and can be applied to practical applications like digitizing historical documents, document digitalization, healthcare and pharmaceutical, digitizing handwritten notes, and attendance sheets.

1.2 Motivation

While discussing the definition for the project we found that the Handwritten Text Recognition is one area where we can work on wide variety of challenges then we discussed with Harsh Kirasata sir and he allowed us to work on this project .

I observed that Handwritten Text Recognition (HTR) is a challenging area because of the vast variability and complexity of human handwriting. Each individual's handwriting is unique.

Manually checking MCQ answer sheets can be time-consuming and prone to human error. An automated system that can quickly and accurately recognize handwritten text can significantly improve the efficiency of the grading process, saving time and reducing errors.

That is why we thought we should work on handwritten text recognizer project.

1.3 Literature Survey

1.3.1 Handwritten character recognition by ayush purohit , shardul singh chauhan in 2016

The study by Ayush Purohit and Shardul Singh Chauhan in 2016 focused on developing a system for handwritten character recognition. They used the MNIST and IAM Handwriting Database datasets to train and test their system, which utilized a combination of feature extraction techniques and machine learning algorithms. The study found that their system achieved high accuracy rates for recognizing handwritten characters, especially when using a combination of feature extraction techniques. The results suggest that their system has the potential to be applied to real-world applications, such as automated form processing and digitizing historical documents.

1.3.2 A comprehensive data analysis on handwritten by meer zohra, d.rajeswara rao

The study by Meer Zohra and D. Rajeswara Rao aimed to provide a comprehensive data analysis of handwritten datasets, including MNIST, IAM Handwriting Database, and CEDAR. They analyzed the characteristics of the datasets, including the distribution of samples across classes, the quality of the images, and the variations in handwriting styles across different writers. They also compared the performance of various machine learning algorithms for handwriting recognition tasks using these datasets. The study found that the quality of the dataset significantly impacts the performance of machine learning algorithms and recommended the use of larger and more diverse datasets for more accurate results.

1.3.3 Review on handwritten digit recognition 1priya, 2 rajendra singh, 3dr. soni changlani.

The review by Priya, Rajendra Singh, and Dr. Soni Changlani focused on the topic of handwritten digit recognition. They analyzed the various techniques and algorithms used for digit recognition, including feature extraction methods, neural networks, and support vector machines. The study found that deep learning techniques, such as convolutional neural networks (CNNs), have shown promising results for digit recognition tasks, outperforming traditional machine learning algorithms. The review also highlighted the importance of dataset quality and preprocessing techniques in achieving accurate results. Overall, the study provided a comprehensive overview of the current state-of-the-art techniques for handwritten digit recognition.

1.3.4 Optical character recognition for cursive handwriting by mehak naz mangoli1, prof. sujata desai.

The study by Mehak Naz Mangoli and Prof. Sujata Desai proposed a system for recognizing cursive handwriting using a combination of feature extraction techniques and machine learning algorithms. The system utilized a database of cursive handwriting samples, which were preprocessed and segmented into individual characters before being used to train and test the OCR system. The study found that their proposed system achieved high accuracy rates for recognizing cursive handwriting characters. The results suggest that their system has the potential to be applied to real-world applications, such as digitizing handwritten documents and forms, and can contribute to the field of OCR for cursive handwriting recognition.

1.4 Existing Solutions and how our approach differs from them. [01]

1.4.1 ABBYY FineReader :[001]

A commercial OCR software that supports handwritten text recognition. It offers advanced features for document recognition, including handwriting recognition, and is widely used in industries such as finance, healthcare, and administration.

ABBYY
FineReader

Fig 1.1 ABBYY FindReader[001]

1.4.2 MyScript : [002]

A handwriting recognition technology developed by MyScript, which offers a range of products for handwriting recognition in various applications, including note-taking apps, drawing apps, and forms processing. MyScript uses deep learning models and offers both online and offline recognition capabilities.

MyScript

Fig 1.2 MyScript[002]

1.4.3 Microsoft Office Lens : [003]

The Microsoft Lens app is a fantastic tool for gathering data from whiteboards, documents, handwritten memos, business cards, and other sources. You can obtain the information in digital form rather than manually inputting it. This tool can digitise drawings, equations, and sketches in addition to converting handwriting to text. The digitised data can be saved and exported to other platforms.

Fig 1.3 Microsoft Office Lens[003]

1.4.4 PDFelement : [004]

On Windows, the utility effectively tackles the greatest range of PDF-related issues. This programme allows you to easily edit, convert, organise, annotate, and protect PDF files. This is a PDF solution for multiple platforms that allows for quick and easy access. The OCR capability of the software quickly turns handwriting into digitised text.

Fig 1.4 PDFelement [004]

1.4.5 GoodNotes 5 : [005]

GoodNotes 5 **app to convert written notes to text**. This is a note-taking tool for storing handwritten notes in digital notebooks. It can also annotate and edit imported PDF files for a paperless workflow. The app maintains your digital notes in a neat organization for quick access.

The app comes with a trial version for feature analysis. It works well with Kindle, Android, and iOS devices.

Fig 1.5 GoodNotes 5[005]

These solutions have different levels of accuracy, performance, and features, and the choice of solution depends on the specific requirements of the application

2. Related Work and Background.

2.1 Functional requirements

2.1.1 Image pre-processing:

The OCR software should be able to pre-process the handwritten text image to improve its quality, such as noise removal, image thresholding, skew correction, and binarization.

2.1.2 Character segmentation:

The software should be able to segment individual characters from the handwritten text image to recognize them correctly.

2.1.3 Feature extraction:

The OCR software should be able to extract features from the segmented characters, such as shape, size, and texture, to identify them accurately.

2.1.4 Machine learning model:

The OCR software should be able to use a machine learning algorithm, such as a neural network, to train a model on a large dataset of handwritten text to recognize characters.

2.1.5Text recognition:

The OCR software should be able to recognise the characters in the segmented text and convert them into digital text that can be used for further processing.

2.1.6Accuracy and performance:

The OCR software should have a high accuracy rate and perform efficiently on different types of handwritten text images, regardless of variations in writing styles, sizes, and shapes.

2.1.7 User interface:

The OCR software should have a user-friendly interface that enables users to upload handwritten text images and view the recognized text output.

2.1.8 Translate one language into another:

The user should be able to convert one language into another.

2.1.9 Digitizing Handwritten Notes:

The user should be able to generate a PDF file from converted handwritten text that has been converted to digital text.

2.1.10 Digitizing Attendance:

The user should be able to digitize handwritten attendance in the form of Excel. It should take handwritten attendance images and digitize them.

2.1.11 Handwritten MCQ test:

The user should be able to evaluate the mark from a handwritten MCQ test by setting the correct option.

2.1.12 Store Result:

The user should be able to store data in the database.

2.1.13 Filter and visualize results:

It should allow for storing data in a database and filtering and visualizing student results using the database.

2.1.14 Generate Result:

The user should be able to generate results by filtering the data.

2.2 Nonfunctional requirements

2.2.1 Accuracy:

The system should accurately recognize hand written text with minimal errors.

2.2.2 Scalability:

The system should be able to handle different handwriting styles, languages, and document formats.

2.2.3 Robustness:

The system should be able to handle variations in handwriting styles, input image quality, and other factors.

2.2.4 Usability:

The system should have a user-friendly interface with feedback and error correction options.

2.2.5 Reliability:

The system should be highly reliable with minimal downtime or disruptions.

2.2.6 Integration:

The system should be easily integratable into existing workflows or systems.

2.2.7 Performance

An ideal application should possess a reliable performance and low occurrence of failures. Both the hardware and software components must be capable of efficiently sending and receiving data from databases, utilizing high baud rates that can range from Mbps to Gbps.

2.2.8 GUI should be interactive and user-friendly.

One of the basic needs for a better user experience is GUI. It should be not simple so the user gets bored but a decent presentation of data attracts the user the use it.

2.3 Technical requirements

- **System requirement**
  - Memory Space Required : Up to 4GB
  - A Webcam or External Camera
  - Min. RAM : 8 GB

- **Operational environment**
  - **Operating system:** Windows, Linux, IOS

2.4 Tools & Technology:

2.4.1 Python [02]

Python is a versatile and user-friendly programming language that can be used for a wide range of purposes. It is considered a high-level language with a focus on readability, achieved through the use of significant indentation. Python is dynamically-typed, which means that the type of a variable is determined at runtime, and garbage-collected, which means that the language automatically frees up memory that is no longer needed. It supports various programming paradigms, such as structured, object-oriented, and functional programming. Python also offers a rich standard library, often referred to as "batteries included," which provides extensive functionality for common tasks.

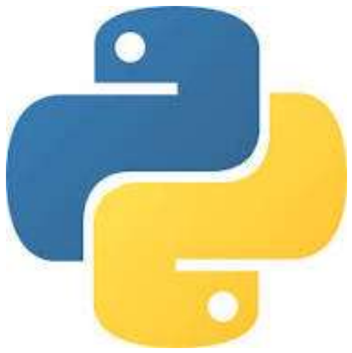


Fig. 2.1 Python [03]

2.4.2 Open cv [04]

OpenCV is a widely-used open-source library that provides extensive functionality for computer vision, machine learning, and image processing. Its capabilities are crucial for real-time operations, which are essential in many modern systems. OpenCV enables users to perform advanced image and video processing tasks, such as object recognition, face detection, and even handwriting recognition. The library's broad range of features makes it a valuable tool for a variety of applications.

Fig. 2.2 Open cv [05]

2.4.3 Object Detection [06]

Object Detection is a computer technology related to computer vision, image processing, and deep learning that deals with detecting instances of objects in images and videos.

Fig. 2.3 Object detection [07]

2.4.4 OCR [08]

OCR is the process of converting images of typed, printed, or handwritten text into machine-readable text. This can be done electronically or mechanically, and can involve scanned documents, photographs of documents, or text from subtitles overlaid on images.

Fig. 2.4 OCR [09]

2.4.5 Tesseract [08]

Tesseract: An open-source OCR engine developed by Google that supports recognition of various languages, including handwritten text. It uses a combination of traditional machine learning algorithms and neural networks for text recognition.

Fig 2.5 Tesseract[11]

2.4.6 Tensorflow[08]

TensorFlow is a Google's open-source machine learning framework, enables developers to create models for image recognition and natural language processing. It offers a high-level Keras API for easy model definition, supports CPU/GPU acceleration, and includes features like automatic differentiation and distributed computing. With a vibrant community of developers and extensive pre-trained models, TensorFlow is popular for its flexibility, efficiency, and tooling.

Fig 2.6 Tensorflow [08]

2.4.7 Keras [08]

Keras is a high-level neural networks API for Python that runs on TensorFlow. It provides an easy-to-use interface for defining, training, and evaluating deep learning models. With support for various network architectures like CNNs, RNNs, and MLPs, Keras simplifies the process of building complex models. It also includes tools for data preprocessing, model evaluation, and visualization. Keras has a large community of users and contributors, making it a popular choice for deep learning projects.

Fig 2.7 Keras[08]

2.4.8 Image Processing[08]

Image processing is a branch of computer vision that deals with modifying digital images in order to extract information, improve visual quality, or perform activities on them. Image filtering, segmentation, feature extraction, recognition, restoration, and compression are some of the techniques covered. Medical imaging, remote sensing, surveillance, and robotics are all disciplines where image processing is used. It is a rapidly expanding subject that is being fueled by advances in machine learning and deep learning approaches.

Fig 2.8 Image Processing[08]

Pre-processing is a series of operations that improves image quality and thereby increases image accuracy. The following pre-processing techniques are used for hand-written character recognition.

2.4.8.1 Noise-removal:

The process of reducing noise from an image is known as noise removal. This also refers to smoothening the image by removing undesirable signals. There are numerous algorithms for removing image noise. Some of these are the Gaussian filtering method, the Min-max filtering method, the Median filter, and so on.

2.4.8.2 Binarization:

Binarization is the process of transforming a grayscale or coloured image to a binary image. Binary images are made up of simply 0s and 1s. Images' pixels are divided into 0s and 1s depending on a fixed value. If the pixel value is less than the constant, it is replaced by 0; otherwise, it is replaced by 1

.**2.4.8.3 Morphological operation:**

This is the process of enlarging or contracting an image. This is done primarily because the algorithm anticipates a consistent image size. We can add pixels to the image's boundaries to enhance its size. We can reduce the size of an image by removing pixels from the image's boundaries.

2.5 Tools

2.5.1 Jupyter Notebook[08]

Jupyter Notebook is a free and open-source web-based tool that lets users create, run, and share live code, equations, visualizations, and narratives in a single document. It supports more than 40 programming languages, including popular ones like Python, R, Julia, and Scala. With its rich set of features for data analysis, machine learning, and scientific computing, such as code execution, markdown cells for documentation, interactive widgets, data visualization, and support for LaTeX equations, Jupyter Notebook is widely used by data scientists, researchers, and educators to create reproducible and interactive computational workflows. Its ability to combine code, text, and multimedia content enables the creation of engaging data-driven narratives.

Fig 2.9 Jupyter Notebook[08]

2.5.2 V.S Code[20]

Visual Studio Code is a Microsoft source code editor developed by Microsoft. It is a free, open-source, cross-platform code editor with powerful features for development and debugging. Visual Studio Code (VS Code) was first released in 2015 and has since become a popular code editor for developers. VS Code is written in Electron, a JavaScript library for creating cross-platform desktop applications with HTML and CSS. It is based on the open-source Monaco editor, which is used by Microsoft's Visual Studio Online.

VS Code is optimized for development in a wide range of languages, including JavaScript, TypeScript, HTML, CSS, and C#. It also provides support for other languages such as Python, Go, and PHP. VS Code provides developers with an integrated development environment (IDE) that supports debugging and intelligent code completion. It has an advanced code editor that understands the syntax of the language and provides code completion, error highlighting, and formatting. VS Code also supports version control systems such as Git and provides built-in support for popular source control systems.

Fig 2.10 VS-code[08]

2.6 Libraries

```
from tkinter import *
from PIL import Image, ImageTk
from tkcalendar import Calendar, DateEntry
from tkinter import filedialog
from tkinter import messagebox
import tkinter.filedialog as fd
import tkinter as tk
import pymysql.cursors
import datetime
import pickle
import math
import shutil
import cv2
import re
```

3. Modeling and Design

For better visualization of implemented methodology, software design diagrams were constructed which provided a glimpse of the flow of control as well as the data in our project. The following diagrams proved useful information for a better understanding of this project.

- Flow chart shows a clear picture of the project how it work.
- Phase Of Handwritten Text Recognition show the what are phases of Handwritten Text Recognition.
- Flow Chart For Proposed Model shows the how model recognizes the handwritten text.
- Model diagram show how we have the create the model.
- Image Pre-processing Diagram how we pre-process the image.
- The Component Of OCR System diagram give idea of how OCR work.
- The ER diagram illustrates the entities that are involved in the application and also the type of data information that they are having.

3.1 Flow Chart Diagram

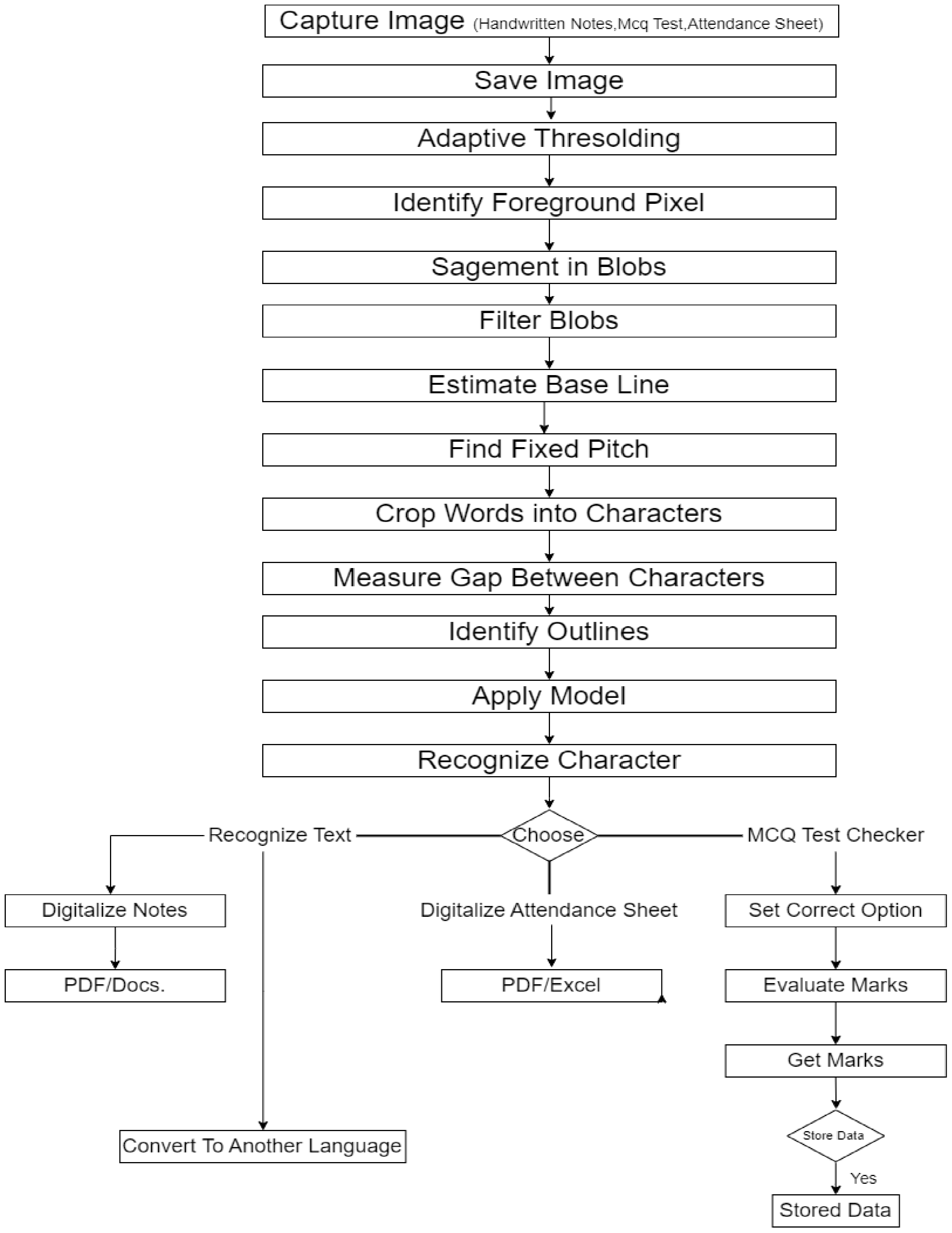


Fig. 3.1 Flow Chart Diagram

3.2 Phase Of Handwritten Text Recognition

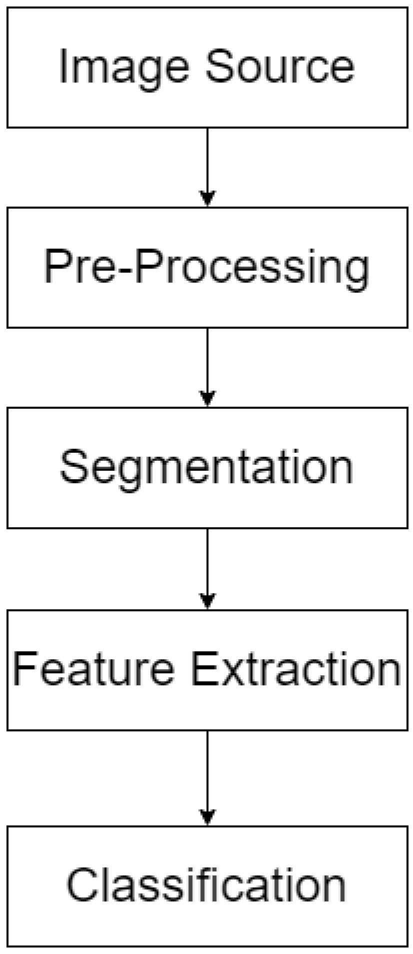


Fig. 3.2 Phase Of Handwritten Text Recognition

3.3 Flow Chart For Proposed Model

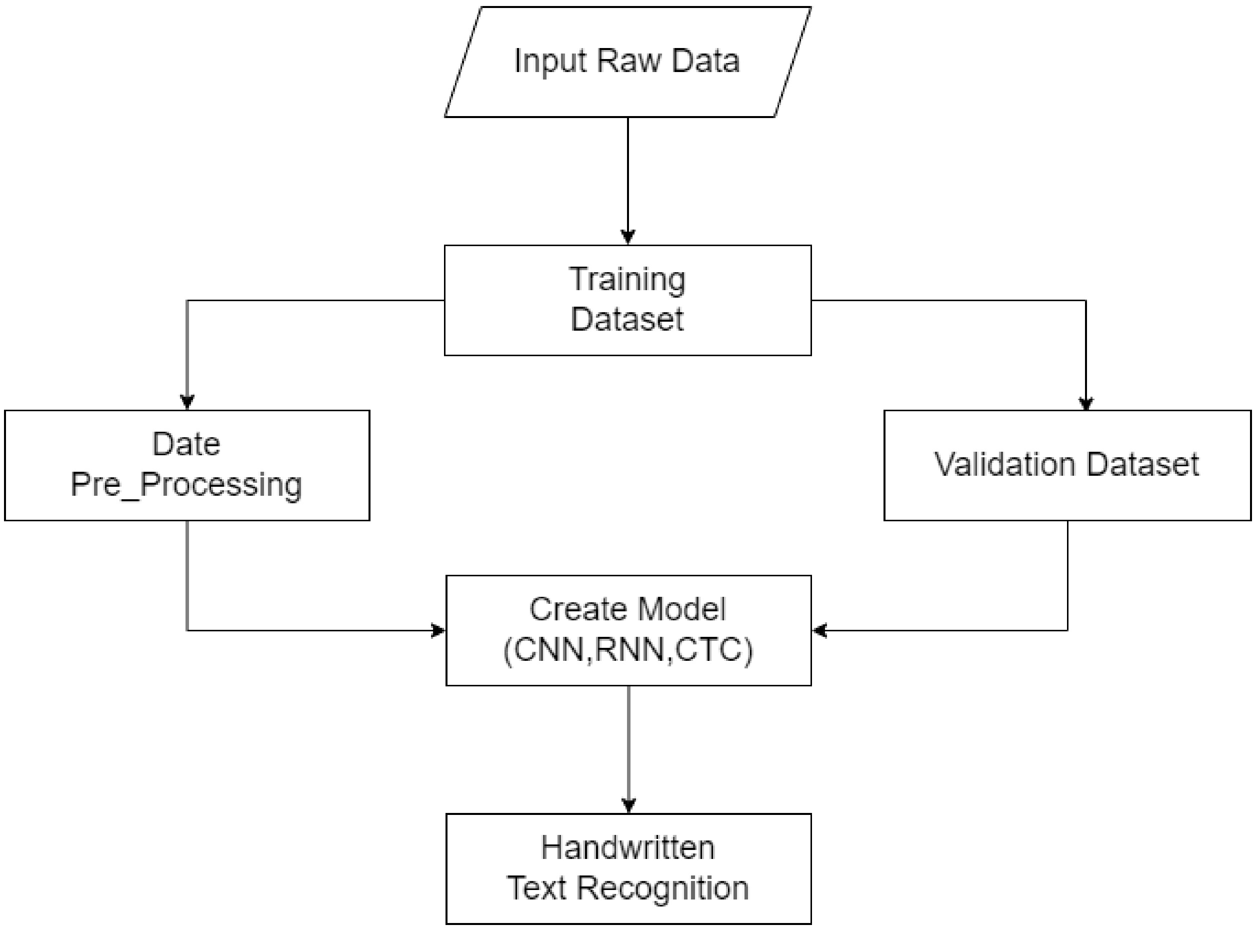


Fig. 3.3 Flow Chart For Proposed Model

3.4 Model Diagram

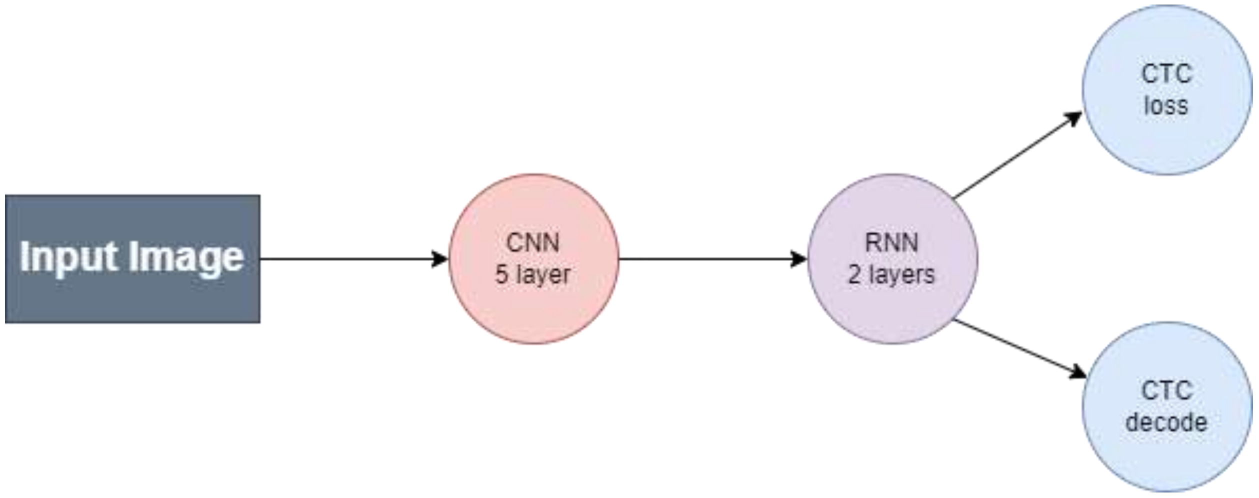


Fig 3.4 Model Diagram

3.4.1 CNN Diagram [08]

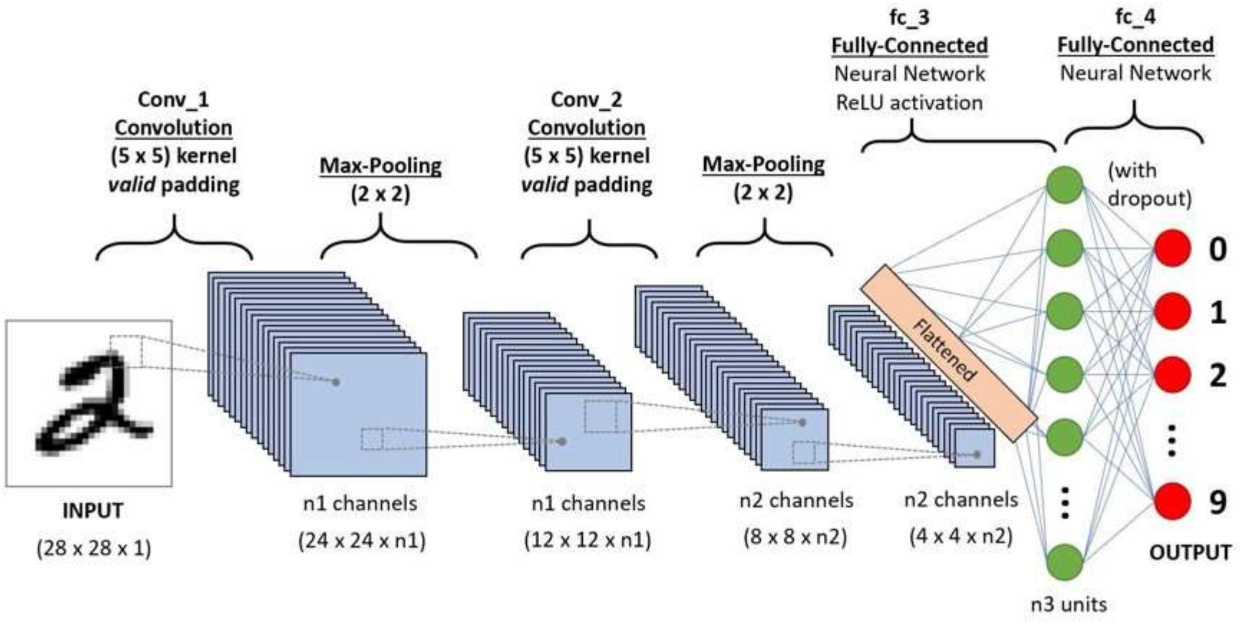


Fig 3.4.1 CNN Diagram

3.4.2 RNN Diagram [08]

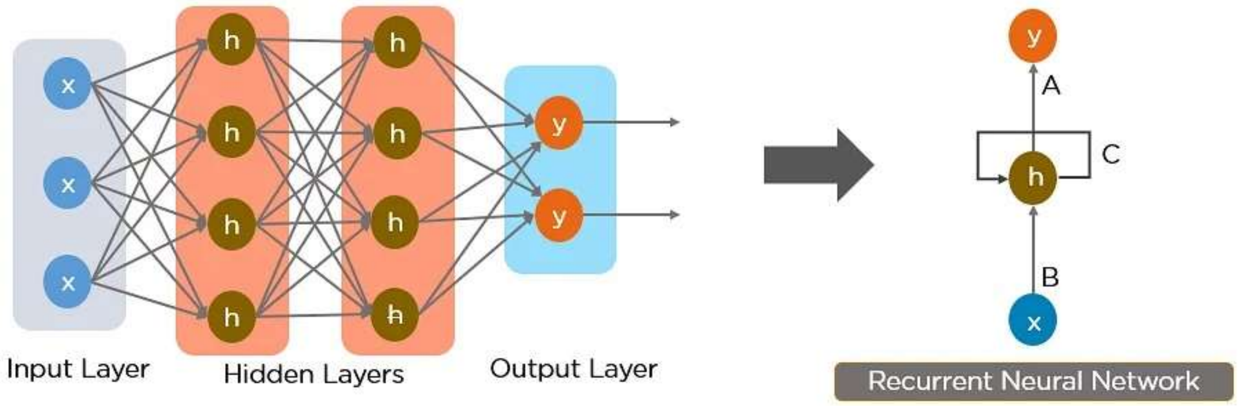


Fig 3.4.2 RNN Diagram

3.5 Image Pre-processing Diagram

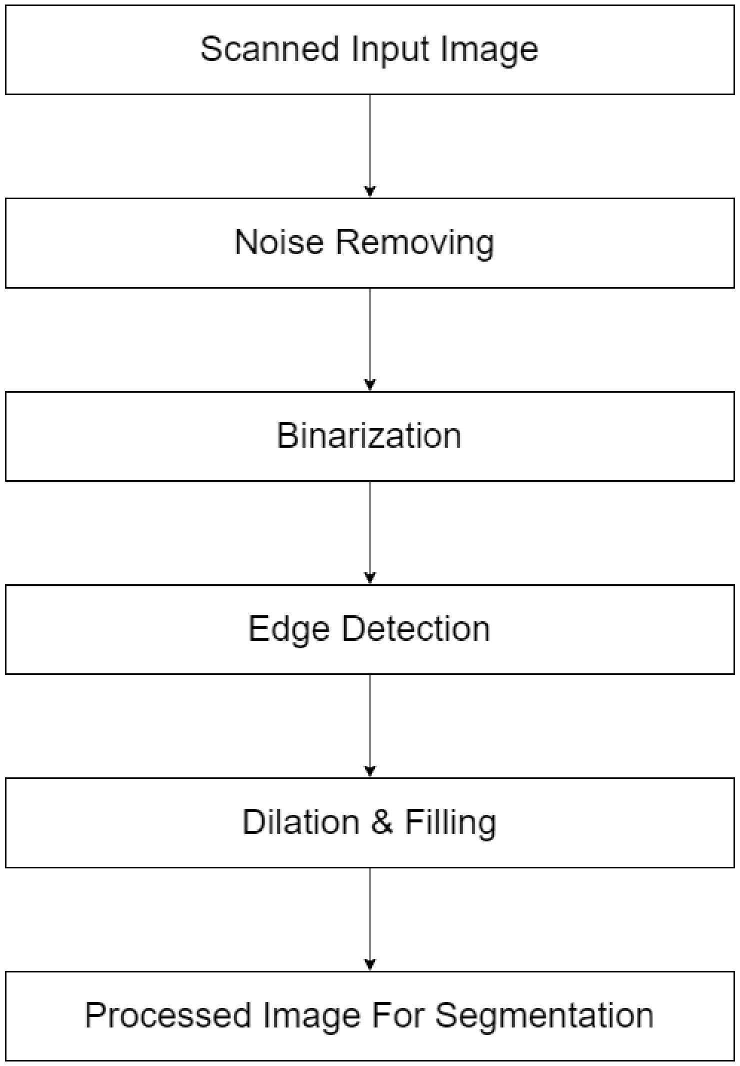


Fig 3.5 Image Pre-processing Diagram

3.6 The Component Of OCR System

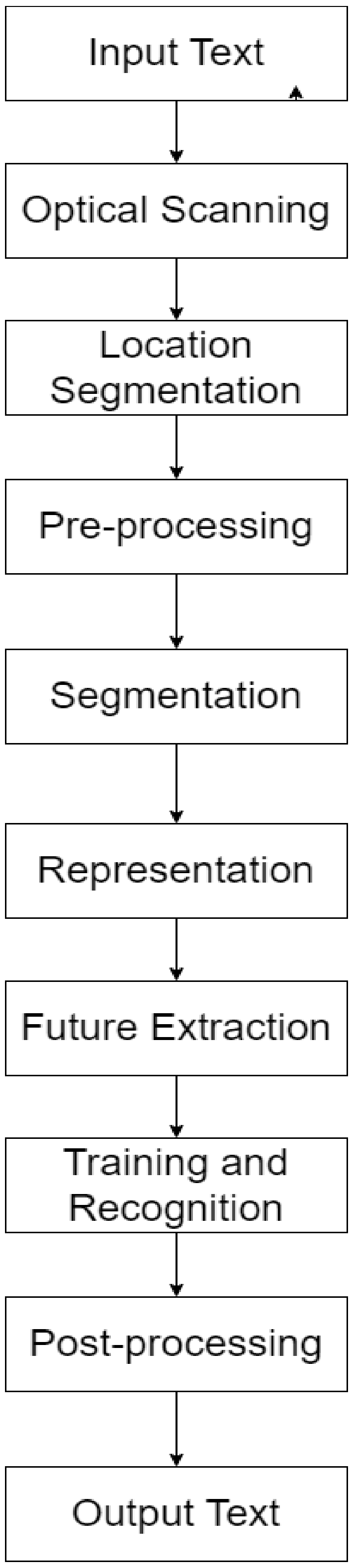


Fig. 3.6 The Component Of OCR System

3.7 E-R Diagram

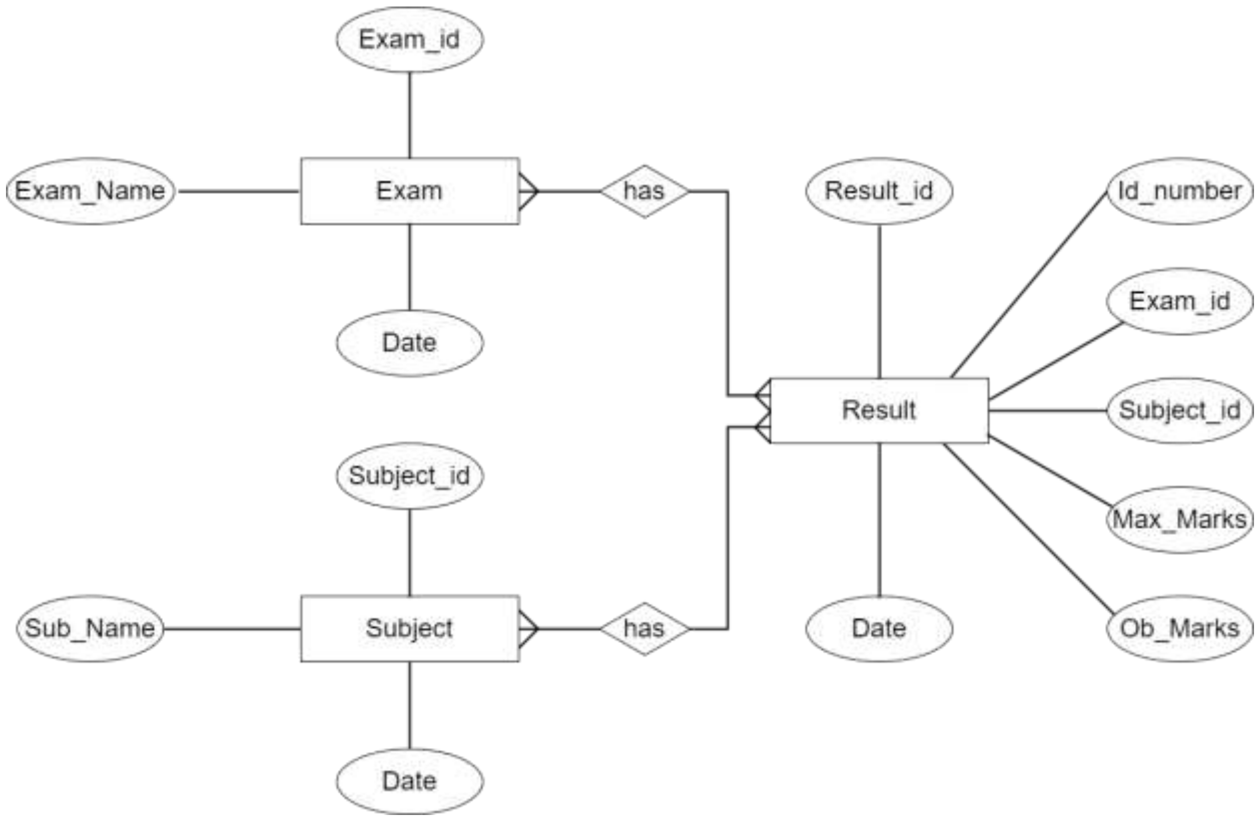


Fig. 3.7 ER diagram

4. Implementation

4.1 Model Implementation

Training a model for all the different handwritings in this world is impossible as handwriting is unique to every individual. So we can train a model using a large dataset of handwritten like the MNIST and IAM dataset.

For digit recognition i have used MNIST dataset and for handwritten text recognition i have used IAM dataset.

In this implementation, I have used the Convolutional Neural Network (CNN).

4.1.1 Dataset

- **MNIST Dataset**
  For create Handwritten digit recognition model we have use mnist dataset. Mnist dataset contains handwritten single digits from 0 to 9. It is a dataset of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9.
  The MNIST dataset is a widely-used benchmark dataset in machine learning, consisting of 70,000 handwritten digits images, with 60,000 images in the training set and 10,000 images in the test set. It is a popular dataset for training and testing machine learning algorithms in image classification.

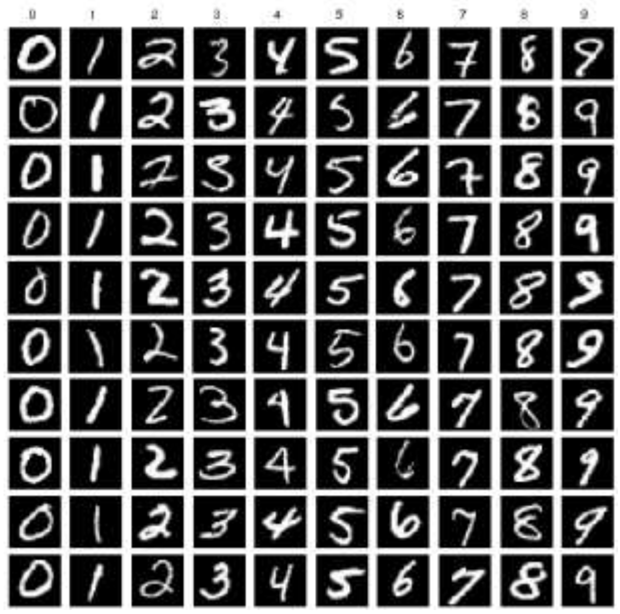


Fig 4.1 MNIST Dataset[25]

- **IAM Dataset**

  We have created model using CNN, RNN and CTC in this we used 5 layer of CNN and 3 layer of RNN.

  We used IAM dataset for creating Handwritten Text Recognition model. IAM dataset contains about 100k images of with words written by 657 different authors. This dataset contains handwritten text samples from over 657 writers, including both English and non-English languages.

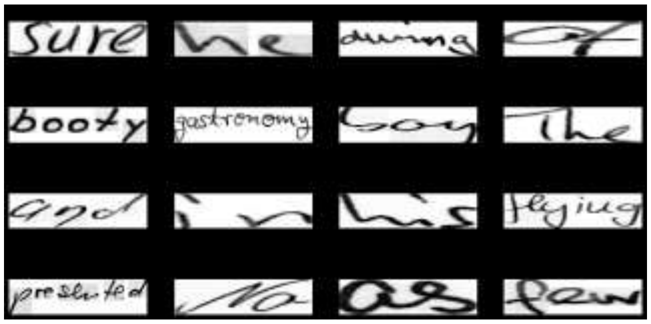
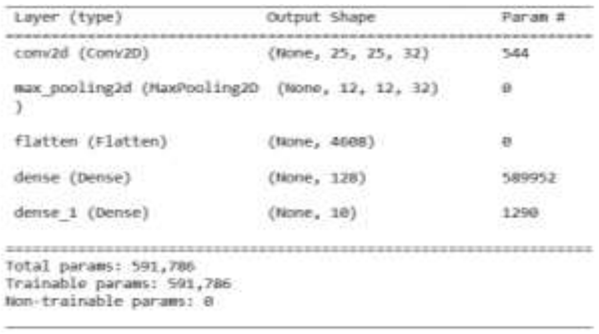


Fig 4.2 IMA Dataset[26]

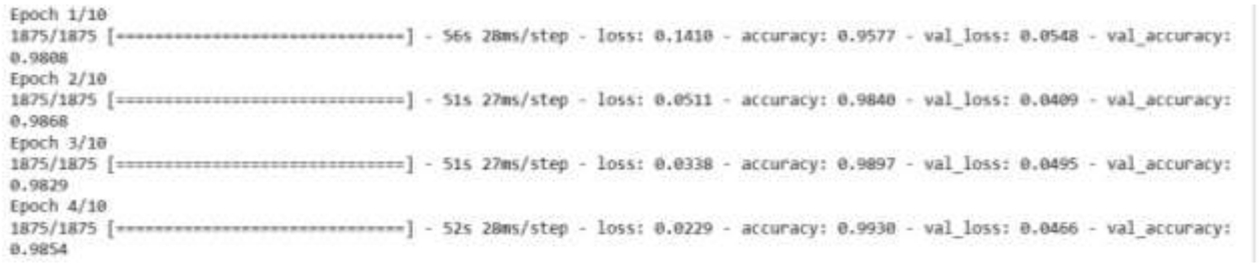
4.1.2 CNN-Model on MNIST dataset

I have used Conv2D layer, which takes a 28x28 input size image and gives output of 25x25 size. I have used the relu activation function, then I added the max-pooling layer, the kernel size is 2x2. For classifying images, I have used dense layers.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 25, 25, 32)        544

 max_pooling2d (MaxPooling2D  (None, 12, 12, 32)       0
 )

 flatten (Flatten)           (None, 4608)              0

 dense (Dense)               (None, 128)               589952

 dense_1 (Dense)             (None, 10)                1290

=================================================================
Total params: 591,786
Trainable params: 591,786
Non-trainable params: 0
_________________________________________________________________
```

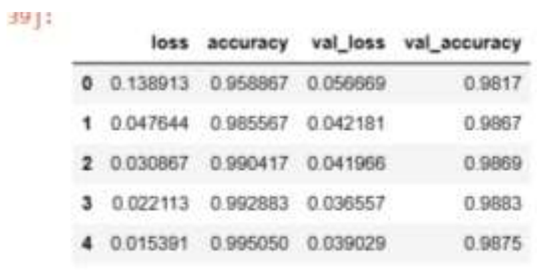
Fig 4.3 Model Summary



```
Epoch 1/10
1875/1875 [==============================] - 56s 28ms/step - loss: 0.1410 - accuracy: 0.9577 - val_loss: 0.0548 - val_accuracy:
0.9808
Epoch 2/10
1875/1875 [==============================] - 51s 27ms/step - loss: 0.0511 - accuracy: 0.9840 - val_loss: 0.0409 - val_accuracy:
0.9868
Epoch 3/10
1875/1875 [==============================] - 51s 27ms/step - loss: 0.0338 - accuracy: 0.9897 - val_loss: 0.0495 - val_accuracy:
0.9829
Epoch 4/10
1875/1875 [==============================] - 52s 28ms/step - loss: 0.0229 - accuracy: 0.9930 - val_loss: 0.0466 - val_accuracy:
0.9854
```

Fig 4.4 Model Training

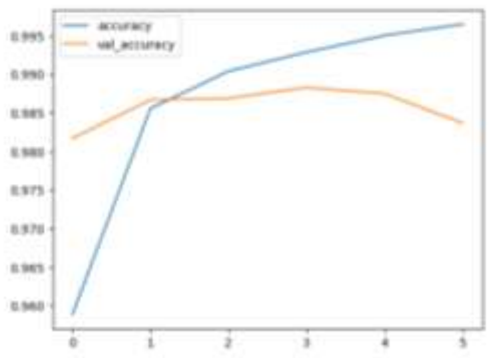
The minimum training accuracy is 95.88% and the minimum validation accuracy is 98.17%. The maximum training accuracy is found 99.50% at epoch 5, the maximum validation accuracy also found 98.75%. The total test loss is found approximately 0.039029.



39]:

	loss	accuracy	val_loss	val_accuracy
0	0.138913	0.958867	0.056669	0.9817
1	0.047644	0.985567	0.042181	0.9867
2	0.030867	0.990417	0.041966	0.9869
3	0.022113	0.992883	0.036557	0.9883
4	0.015391	0.995050	0.039029	0.9875

Fig 4.5 Model Training



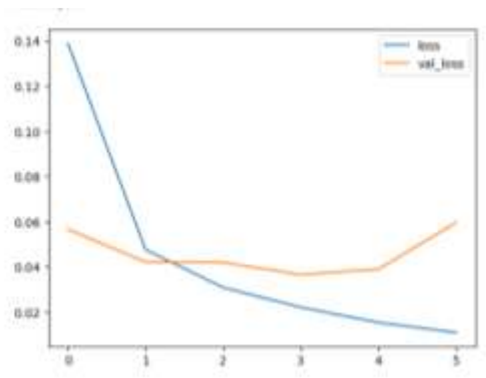


Fig 4.6 Training Accuracy Vs Validation Accuracy

Fig 4.7 Training Loss Vs Validation Loss

4.1.2 CNN-Model on IAM dataset

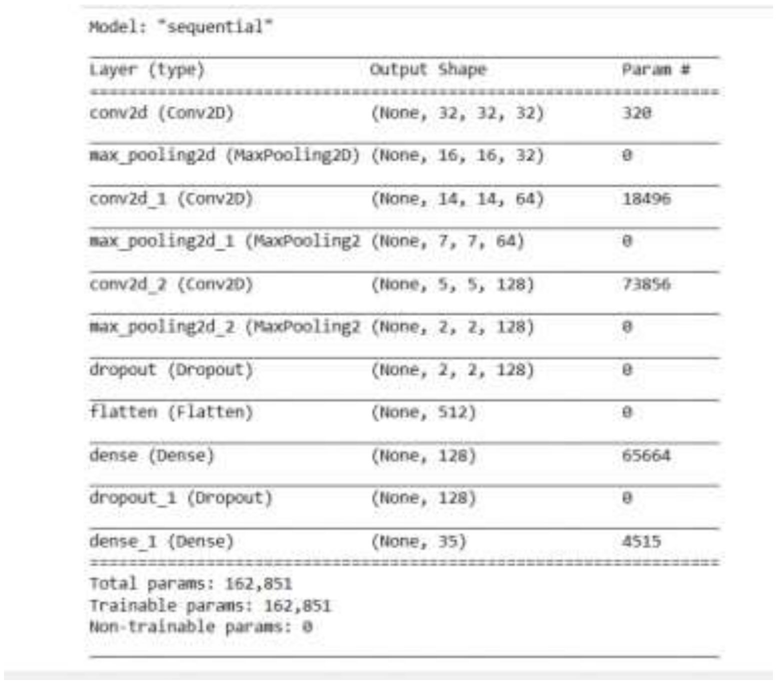
I have used Conv2D layer, which takes a 32x32 input size image. Conv2D() adds a convolutional layer with 32 filters, a kernel size of 3x3. The input_shape parameter specifies the input shape of the images to be fed to the model, which is 32x32 with a single channel (i.e., grayscale images). The `activation` parameter is set to 'relu'.

MaxPooling2D() adds a max pooling layer with a pool size of 2x2, which downsamples the output of the convolutional layer.

Two more Conv2D and MaxPooling2D layers are added to capture more complex features.

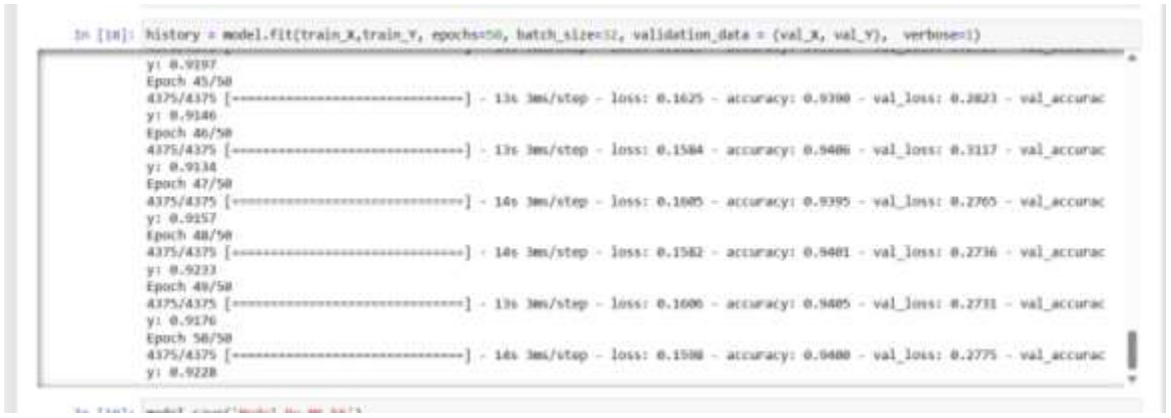
Dropout() adds a regularization layer that randomly drops 25% of the neurons to prevent overfitting. Dropout() adds another regularization layer that randomly drops 20% of the neurons to prevent overfitting.

Two fully connected neural network layers with 128 and 35 neurons are added, respectively. The 'relu' activation function is used in the first layer, and the 'softmax' activation function is used in the last layer to obtain the probability distribution of the predicted classes.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 32, 32, 32)        320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 14, 14, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 7, 7, 64)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 5, 5, 128)         73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 2, 2, 128)         0
_________________________________________________________________
dropout (Dropout)            (None, 2, 2, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 128)               65664
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 35)                4515
=================================================================
Total params: 162,851
Trainable params: 162,851
Non-trainable params: 0
_________________________________________________________________
```

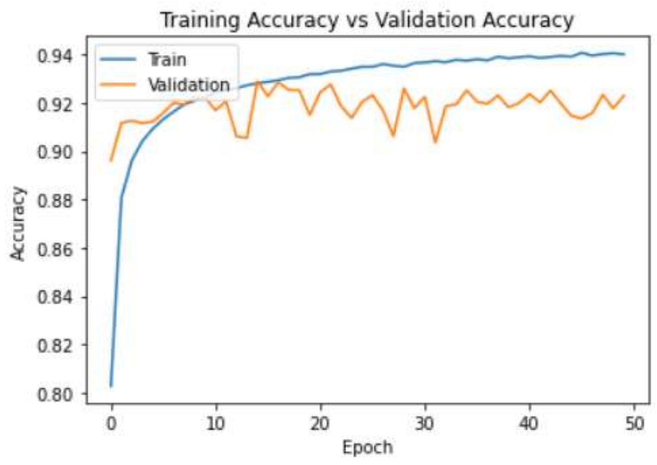
Fig 4.8 Model Summary



```
In [18]: history = model.fit(train_X,train_Y, epochs=50, batch_size=32, validation_data = (val_X, val_Y),  verbose=1)
```

```
y: 0.9197
Epoch 45/50
4375/4375 [==============================] - 13s 3ms/step - loss: 0.1625 - accuracy: 0.9390 - val_loss: 0.2823 - val_accurac
y: 0.9146
Epoch 46/50
4375/4375 [==============================] - 13s 3ms/step - loss: 0.1584 - accuracy: 0.9406 - val_loss: 0.3117 - val_accurac
y: 0.9134
Epoch 47/50
4375/4375 [==============================] - 14s 3ms/step - loss: 0.1605 - accuracy: 0.9395 - val_loss: 0.2765 - val_accurac
y: 0.9157
Epoch 48/50
4375/4375 [==============================] - 14s 3ms/step - loss: 0.1582 - accuracy: 0.9401 - val_loss: 0.2736 - val_accurac
y: 0.9233
Epoch 49/50
4375/4375 [==============================] - 13s 3ms/step - loss: 0.1606 - accuracy: 0.9405 - val_loss: 0.2731 - val_accurac
y: 0.9176
Epoch 50/50
4375/4375 [==============================] - 14s 3ms/step - loss: 0.1598 - accuracy: 0.9400 - val_loss: 0.2775 - val_accurac
y: 0.9228
```

Fig 4.9 Model Train



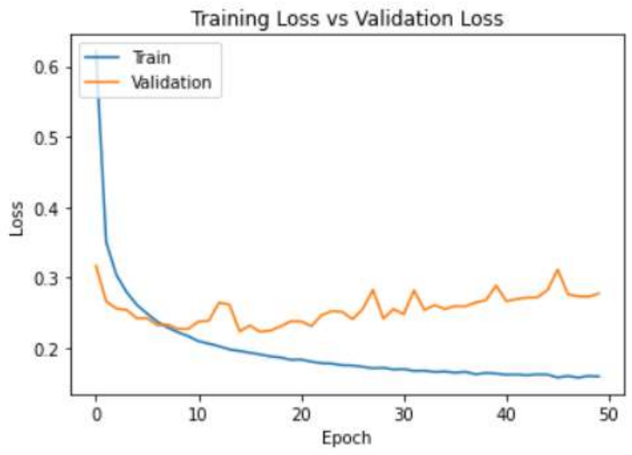


Fig 4.10 Training Accuracy Vs Validation Accuracy

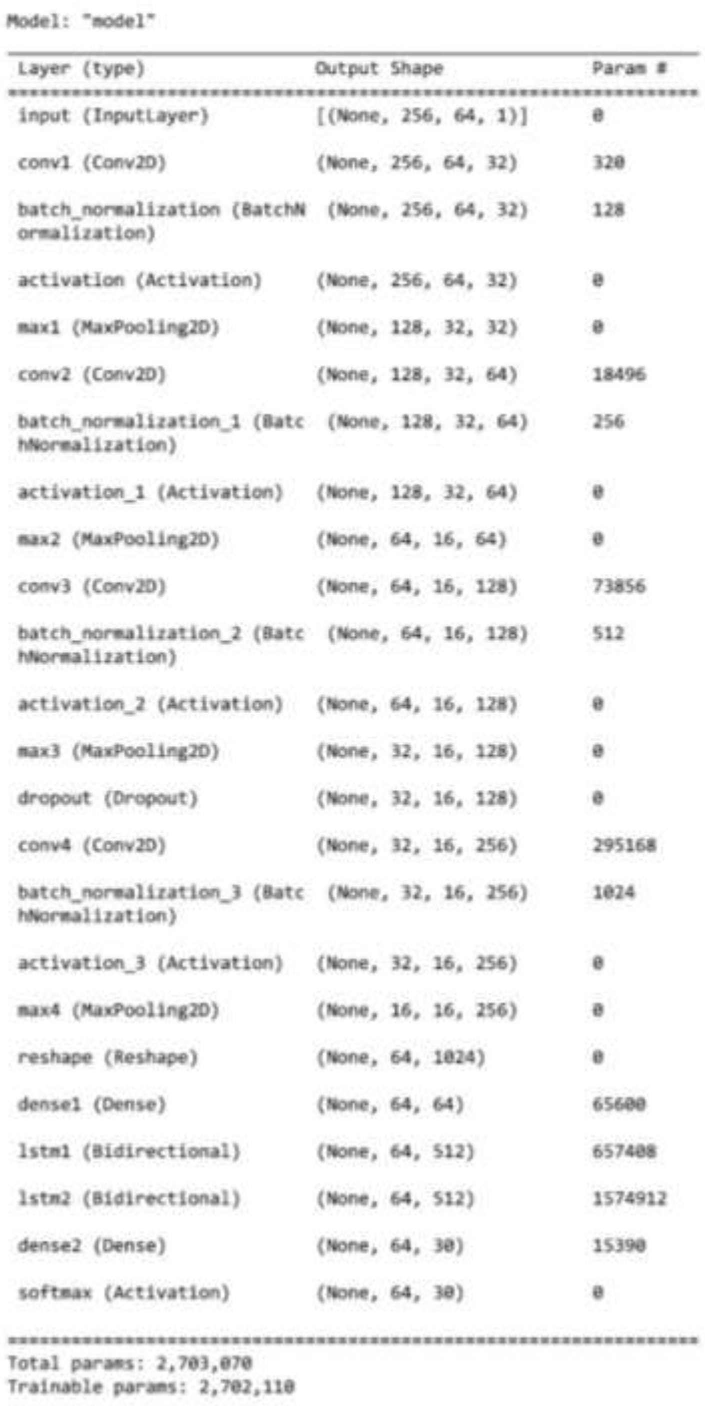
Fig 4.11 Training Loss Vs Validation Loss

4.1.3 CNN-RNN Model

CNN: The CNN layers receive the input image as input. These layers have been taught to extract from the image the necessary features. Every layer has three operations. First, there is the convolution process, which applies a filter kernel to the input that is 55 in size for the first two layers and 33 for the final three layers. The non-linear RELU function is then utilised. A pooling layer then outputs a condensed version of the input after summarising image regions. Feature maps (channels) are created while the image height is shrunk by two in each layer, resulting in an output feature map (or sequence) that is 32x256 in size.

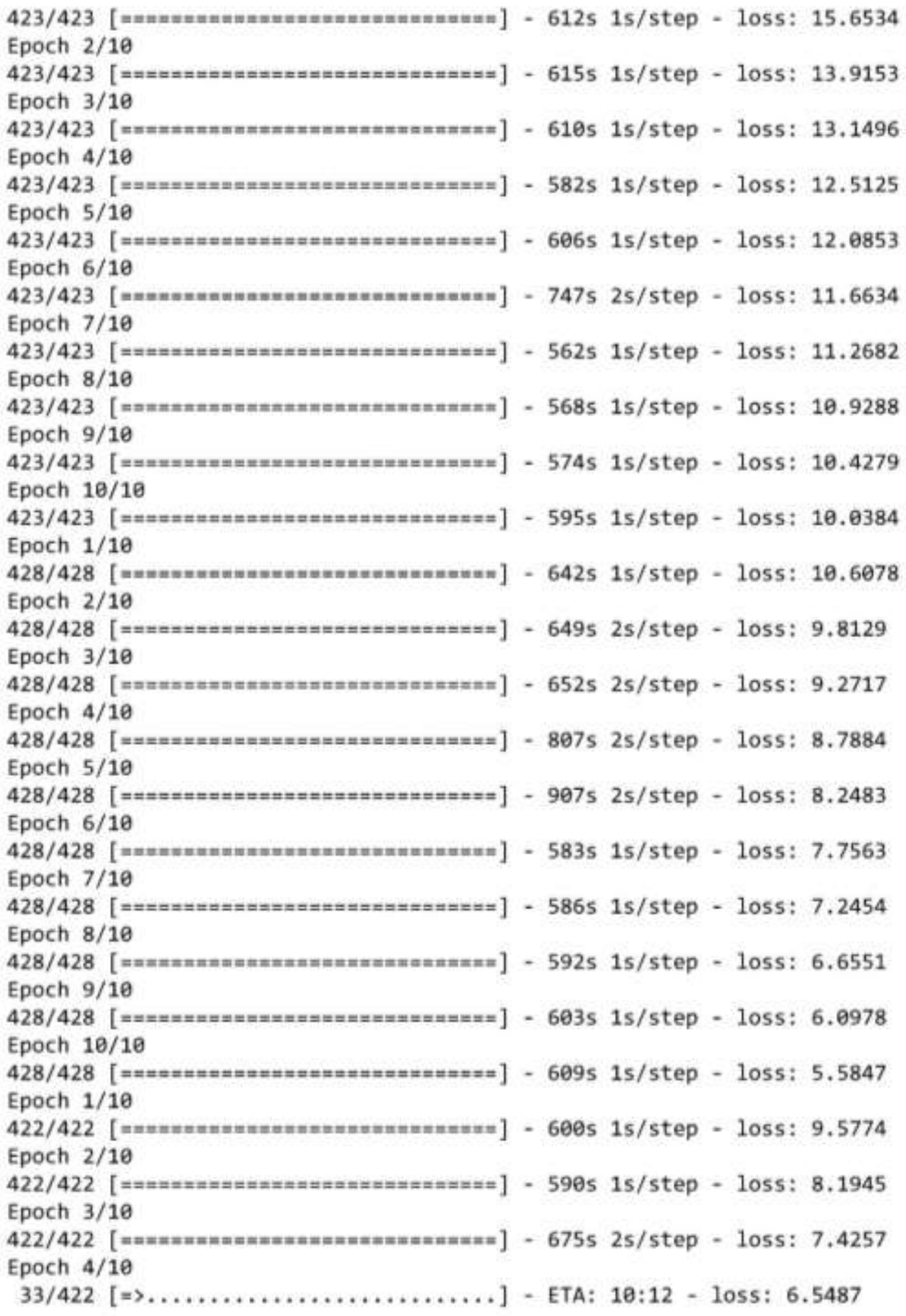
RNN: Each time step in the feature sequence has 256 features, and the RNN propagates important information across this sequence. The well-known Long Short-Term Memory (LSTM) implementation of RNNs is chosen since it has more robust training features than a vanilla RNN and can transmit information over greater distances. A matrix with a dimension of 3280 is used to map the RNN output sequence. The IAM dataset has 79 different characters, and the CTC procedure requires an additional character (the CTC blank label), making 80 entries for each of the 32 time-steps.

CTC: The CTC computes the loss value when training the NN using the ground truth text and RNN output matrix. The CTC only receives the matrix during inference, which it uses to decode into the final text. The maximum character length for the recognised text and the ground truth text is 32..



```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input (InputLayer)          [(None, 256, 64, 1)]      0

 conv1 (Conv2D)              (None, 256, 64, 32)       320

 batch_normalization (BatchN  (None, 256, 64, 32)      128
 ormalization)

 activation (Activation)     (None, 256, 64, 32)       0

 max1 (MaxPooling2D)         (None, 128, 32, 32)       0

 conv2 (Conv2D)              (None, 128, 32, 64)       18496

 batch_normalization_1 (Batc  (None, 128, 32, 64)      256
 hNormalization)

 activation_1 (Activation)   (None, 128, 32, 64)       0

 max2 (MaxPooling2D)         (None, 64, 16, 64)        0

 conv3 (Conv2D)              (None, 64, 16, 128)       73856

 batch_normalization_2 (Batc  (None, 64, 16, 128)      512
 hNormalization)

 activation_2 (Activation)   (None, 64, 16, 128)       0

 max3 (MaxPooling2D)         (None, 32, 16, 128)       0

 dropout (Dropout)           (None, 32, 16, 128)       0

 conv4 (Conv2D)              (None, 32, 16, 256)       295168

 batch_normalization_3 (Batc  (None, 32, 16, 256)      1024
 hNormalization)

 activation_3 (Activation)   (None, 32, 16, 256)       0

 max4 (MaxPooling2D)         (None, 16, 16, 256)       0

 reshape (Reshape)           (None, 64, 1024)          0

 dense1 (Dense)              (None, 64, 64)            65600

 lstm1 (Bidirectional)       (None, 64, 512)           657408

 lstm2 (Bidirectional)       (None, 64, 512)           1574912

 dense2 (Dense)              (None, 64, 30)            15390

 softmax (Activation)        (None, 64, 30)            0

=================================================================
Total params: 2,703,070
Trainable params: 2,702,110
```

Fig 4.12 Model Summary



```
423/423 [==============================] - 612s 1s/step - loss: 15.6534
Epoch 2/10
423/423 [==============================] - 615s 1s/step - loss: 13.9153
Epoch 3/10
423/423 [==============================] - 610s 1s/step - loss: 13.1496
Epoch 4/10
423/423 [==============================] - 582s 1s/step - loss: 12.5125
Epoch 5/10
423/423 [==============================] - 606s 1s/step - loss: 12.0853
Epoch 6/10
423/423 [==============================] - 747s 2s/step - loss: 11.6634
Epoch 7/10
423/423 [==============================] - 562s 1s/step - loss: 11.2682
Epoch 8/10
423/423 [==============================] - 568s 1s/step - loss: 10.9288
Epoch 9/10
423/423 [==============================] - 574s 1s/step - loss: 10.4279
Epoch 10/10
423/423 [==============================] - 595s 1s/step - loss: 10.0384
Epoch 1/10
428/428 [==============================] - 642s 1s/step - loss: 10.6078
Epoch 2/10
428/428 [==============================] - 649s 2s/step - loss: 9.8129
Epoch 3/10
428/428 [==============================] - 652s 2s/step - loss: 9.2717
Epoch 4/10
428/428 [==============================] - 807s 2s/step - loss: 8.7884
Epoch 5/10
428/428 [==============================] - 907s 2s/step - loss: 8.2483
Epoch 6/10
428/428 [==============================] - 583s 1s/step - loss: 7.7563
Epoch 7/10
428/428 [==============================] - 586s 1s/step - loss: 7.2454
Epoch 8/10
428/428 [==============================] - 592s 1s/step - loss: 6.6551
Epoch 9/10
428/428 [==============================] - 603s 1s/step - loss: 6.0978
Epoch 10/10
428/428 [==============================] - 609s 1s/step - loss: 5.5847
Epoch 1/10
422/422 [==============================] - 600s 1s/step - loss: 9.5774
Epoch 2/10
422/422 [==============================] - 590s 1s/step - loss: 8.1945
Epoch 3/10
422/422 [==============================] - 675s 2s/step - loss: 7.4257
Epoch 4/10
 33/422 [=>............................] - ETA: 10:12 - loss: 6.5487
```

Fig 4.13 Model Train

4.2 Image Processing

In image processing we converted the color image to black and white and then we detected the specific word using OpenCV and then I cropped it and we saved it in a folder.

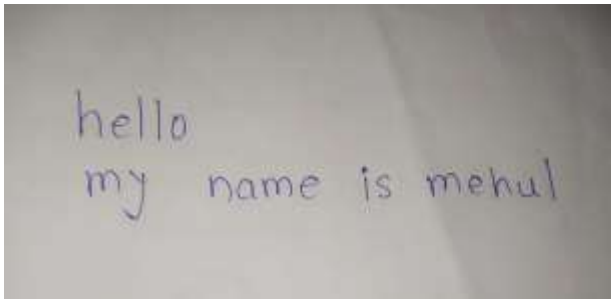


Fig 4.14 Original Image Image

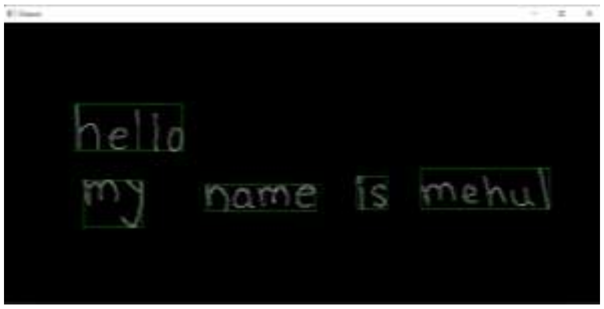


Fig 4.15 Detect Word From Image

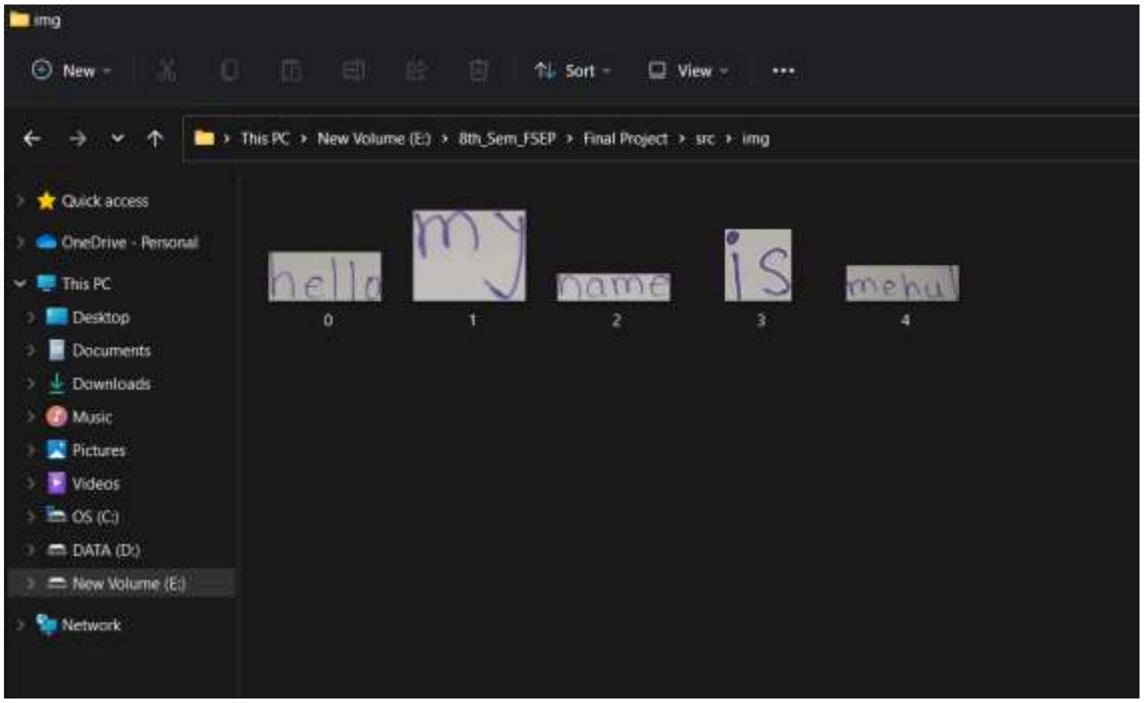


Fig 4.16 Crops Images

4.3 MCQ Test Checker

This is the first page of our MCQ Test Checker

This page is shows up whenever user opens the project. Through this page user can access the different types of functionality that we have made such as Uploading image, Live detection, Handwritten MCQ Test checking, Storing data into database, Visualizing result ,converting handwritten attendance into an Excel file.

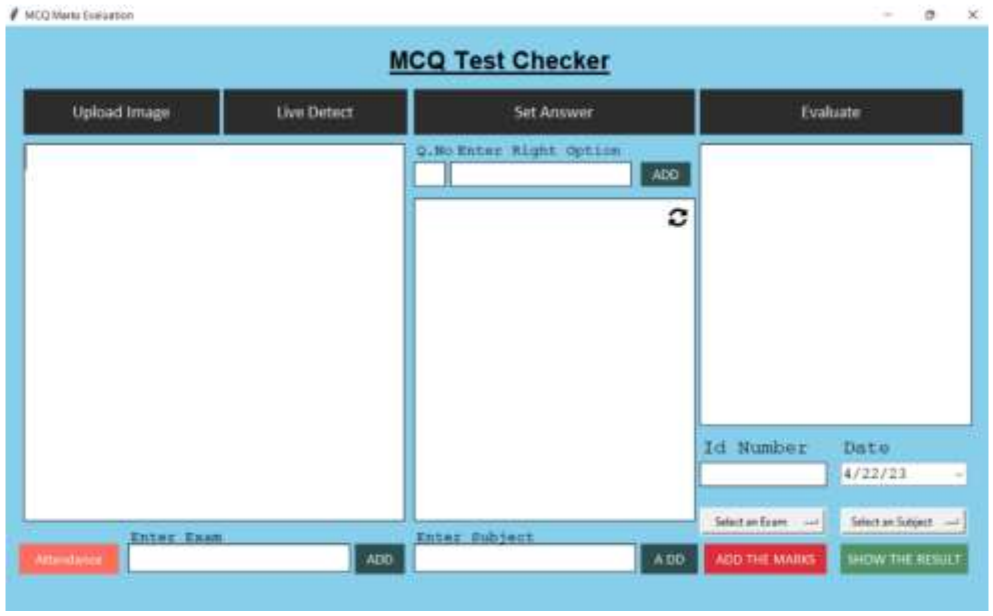
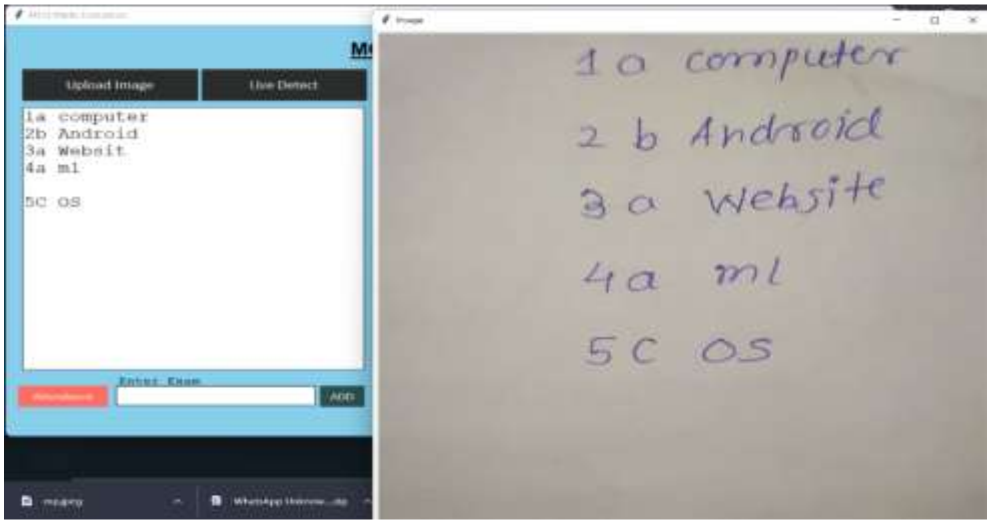


Fig 4.17 MCQ Test Cheaker

4.3.1 Upload Answer Sheet

By clicking on the upload image button the user can upload and recognized the answer sheet.



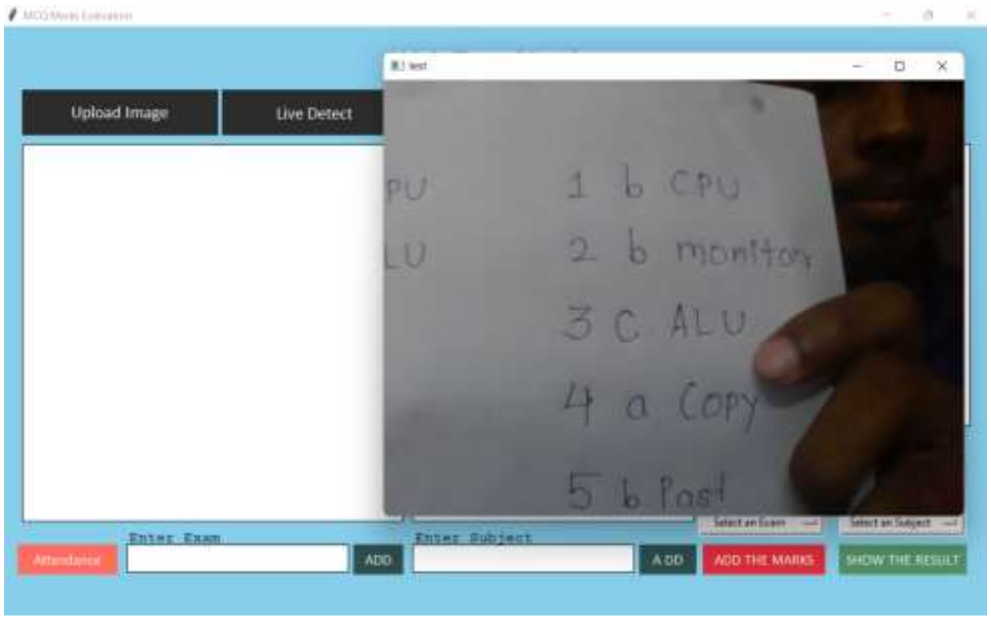
```
1a computer
2b Android
3a Websit
4a ml

5C os
```

```
1 a computer
2 b Android
3 a Website
4 a ml
5 C OS
```

Fig 4.18 Upload Answer Sheet

4.3.2 Live Detect

User can also take answer sheet or attendance sheet through camera.

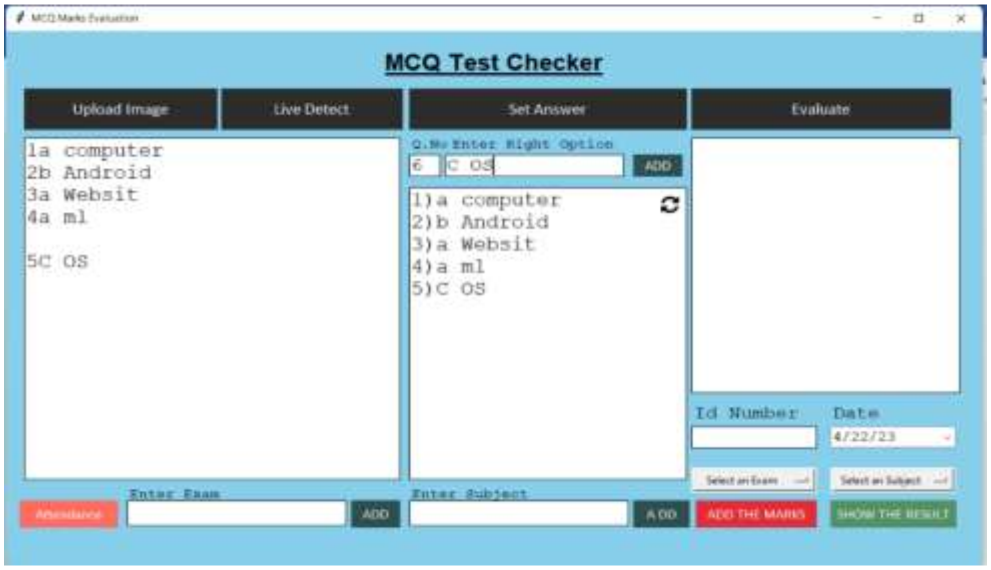


```
1 b CPU
2 b monitor
3 C ALU
4 a Copy
5 b Post
```

Fig 4.19 Live Detection

4.3.3 Set Answer

For evaluating marks we have to set correct answer so that system will give marks according set answer.



```
1a computer
2b Android
3a Websit
4a ml

5C OS
```

Q.No: 6 Enter Right Option: C OS

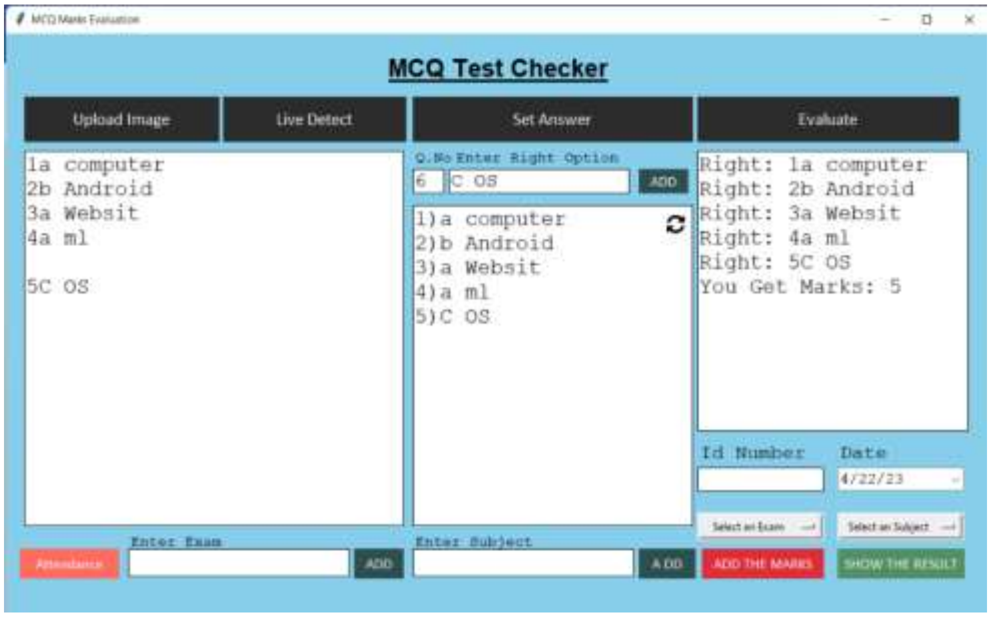
```
1)a computer
2)b Android
3)a Websit
4)a ml
5)C OS
```

Date: 4/22/23

Fig 4.20 Set Answer

4.3.4 Mark Evaluate

For evaluating marks user have to click on Evaluate Button.



```
1a computer
2b Android
3a Websit
4a ml

5C OS
```

Q.No: 6 Enter Right Option: C OS

```
1)a computer
2)b Android
3)a Websit
4)a ml
5)C OS
```

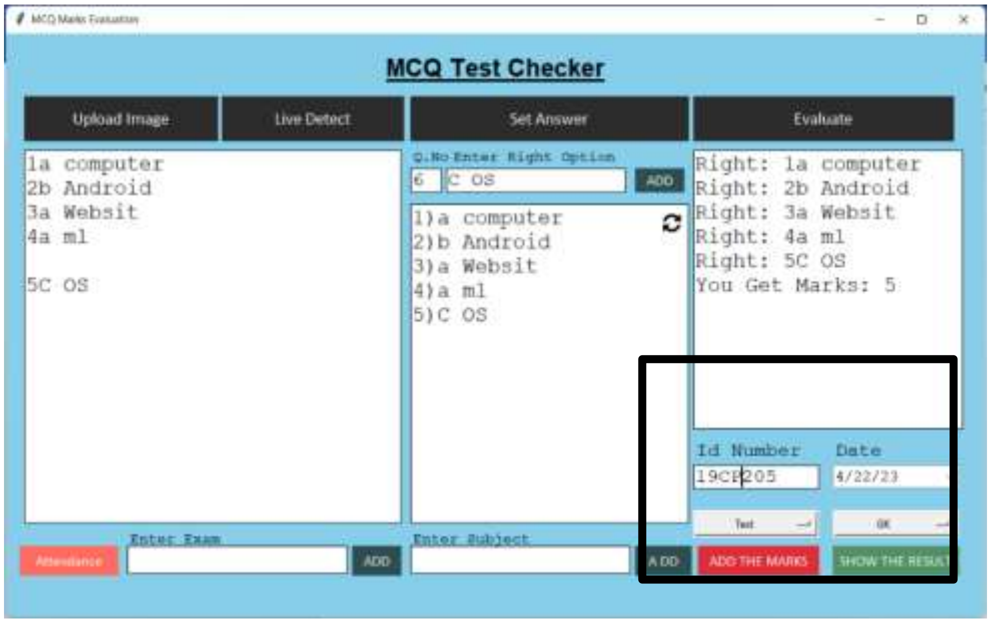
```
Right: 1a computer
Right: 2b Android
Right: 3a Websit
Right: 4a ml
Right: 5C OS
You Get Marks: 5
```

Date: 4/22/23

Fig 4.21 Mark Evalute

4.3.5 Stored Result

User can store result into database by entering their ID Number, Date, Exam name and Subject name.



```
1a computer
2b Android
3a Websit
4a ml

5C OS
```

Q.No Enter Right Option

6 | C OS

```
1)a computer
2)b Android
3)a Websit
4)a ml
5)C OS
```

```
Right: 1a computer
Right: 2b Android
Right: 3a Websit
Right: 4a ml
Right: 5C OS
You Get Marks: 5
```

Id Number: 19CP205
Date: 4/22/23

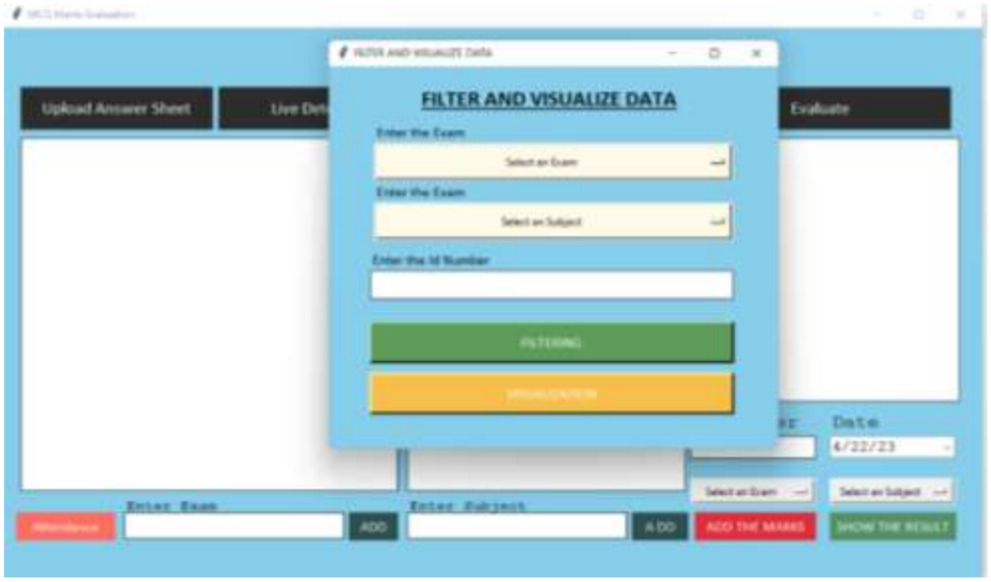
Enter Exam

Enter Subject

Fig 4.22 Store Result

4.4 Filter and visualize data

In this page users can filter and visualize data according to them by selecting Exam, Subject and Id Number.



Enter the Exam

Enter the Exam

Enter the Id Number

Date: 4/22/23

Enter Exam

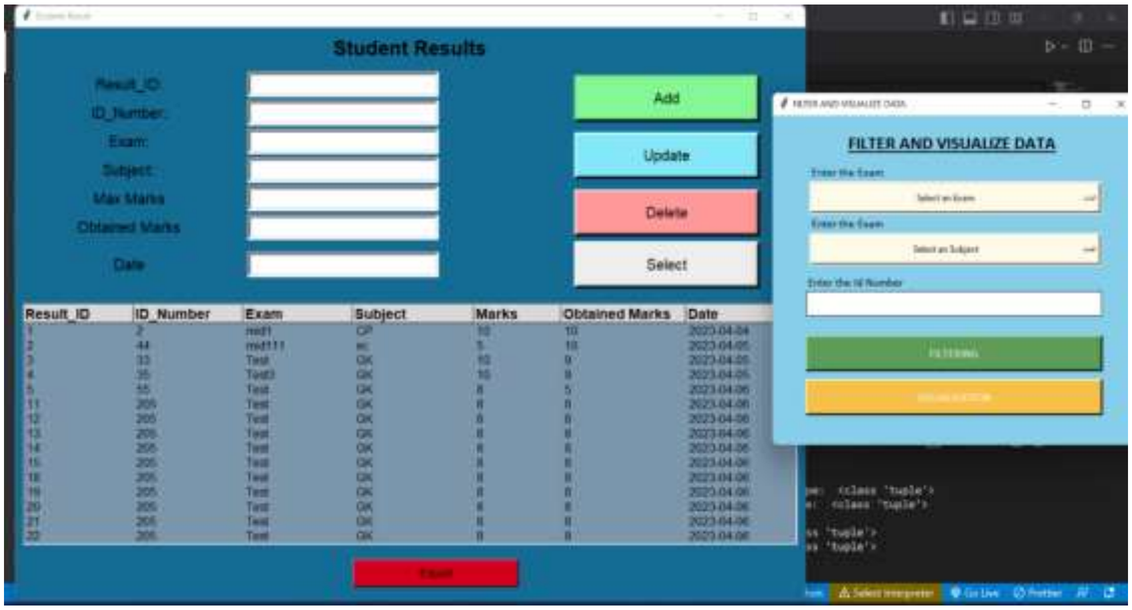
Enter Subject

Fig 4.23 Filter and visualize data

4.4.1 Result Page

Users can filter results and according to user can see the result. User can add manually result also and if user wants update or delete then user can do.

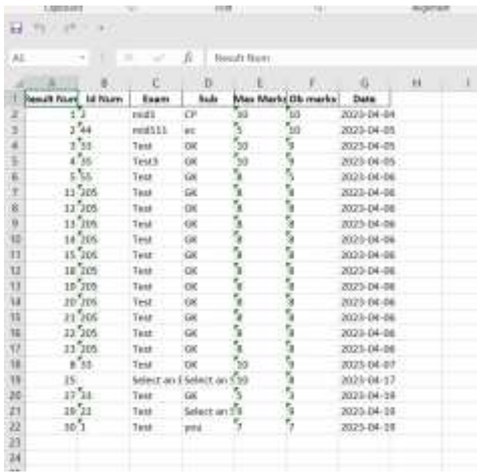
User can also export the result sheet in the pdf or excel format by click export button



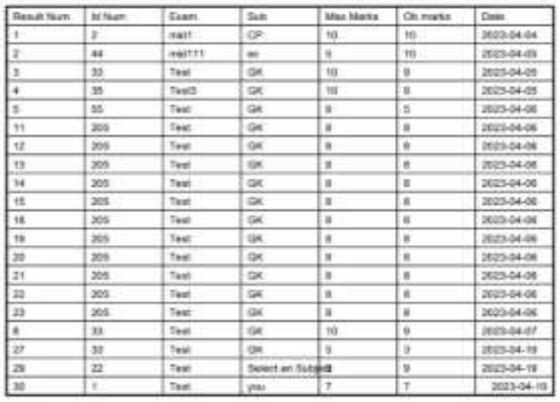
Result_ID	ID_Number	Exam	Subject	Marks	Obtained Marks	Date
1	2	mid1	CP	10	10	2023-04-04
2	44	mid111	ec	5	10	2023-04-05
3	33	Test	GK	10	9	2023-04-05
4	35	Test3	GK	10	9	2023-04-05
5	55	Test	GK	8	5	2023-04-06
11	205	Test	GK	8	8	2023-04-06
12	205	Test	GK	8	8	2023-04-06
13	205	Test	GK	8	8	2023-04-06
14	205	Test	GK	8	8	2023-04-06
15	205	Test	GK	8	8	2023-04-06
18	205	Test	GK	8	8	2023-04-06
19	205	Test	GK	8	8	2023-04-06
20	205	Test	GK	8	8	2023-04-06
21	205	Test	GK	8	8	2023-04-06
22	205	Test	GK	8	8	2023-04-06

```
pe:  <class 'tuple'>
e:  <class 'tuple'>
ss 'tuple'>
ss 'tuple'>
```

Fig 4.24 Result Page



	A	B	C	D	E	F	G
1	Result Num	Id Num	Exam	Sub	Max Marks	Ob marks	Date
2	1	2	mid1	CP	10	10	2023-04-04
3	2	44	mid111	ec	5	10	2023-04-05
4	3	33	Test	GK	10	9	2023-04-05
5	4	35	Test3	GK	10	9	2023-04-05
6	5	55	Test	GK	8	5	2023-04-06
7	11	205	Test	GK	8	8	2023-04-06
8	12	205	Test	GK	8	8	2023-04-06
9	13	205	Test	GK	8	8	2023-04-06
10	14	205	Test	GK	8	8	2023-04-06
11	15	205	Test	GK	8	8	2023-04-06
12	18	205	Test	GK	8	8	2023-04-06
13	19	205	Test	GK	8	8	2023-04-06
14	20	205	Test	GK	8	8	2023-04-06
15	21	205	Test	GK	8	8	2023-04-06
16	22	205	Test	GK	8	8	2023-04-06
17	23	205	Test	GK	8	8	2023-04-06
18	8	33	Test	GK	10	9	2023-04-07
19	25		Select an E	Select an S	10	8	2023-04-17
20	27	33	Test	GK	5	3	2023-04-19
21	29	22	Test	Select an S	8	9	2023-04-19
22	30	1	Test	you	7	7	2023-04-19



Result Num	Id Num	Exam	Sub	Max Marks	Ob marks	Date
1	2	mid1	CP	10	10	2023-04-04
2	44	mid111	ec	5	10	2023-04-05
3	33	Test	GK	10	9	2023-04-05
4	35	Test3	GK	10	9	2023-04-05
5	55	Test	GK	8	5	2023-04-06
11	205	Test	GK	8	8	2023-04-06
12	205	Test	GK	8	8	2023-04-06
13	205	Test	GK	8	8	2023-04-06
14	205	Test	GK	8	8	2023-04-06
15	205	Test	GK	8	8	2023-04-06
18	205	Test	GK	8	8	2023-04-06
19	205	Test	GK	8	8	2023-04-06
20	205	Test	GK	8	8	2023-04-06
21	205	Test	GK	8	8	2023-04-06
22	205	Test	GK	8	8	2023-04-06
23	205	Test	GK	8	8	2023-04-06
8	33	Test	GK	10	9	2023-04-07
27	33	Test	GK	5	3	2023-04-19
29	22	Test	Select an Subject	8	9	2023-04-19
30	1	Test	you	7	7	2023-04-19

Fig 4.25 Convert Result into Excel Sheet

Fig 4.26 Convert Result into PDF

4.4.2 Visualization

Users can visualize the result data. Users can visualize the performance of the student by entering id number. User will visualize student performance in the line or bar chart form.

User can visualize the result of the whole class by entering the subject or exam name as per the performance of the whole student.

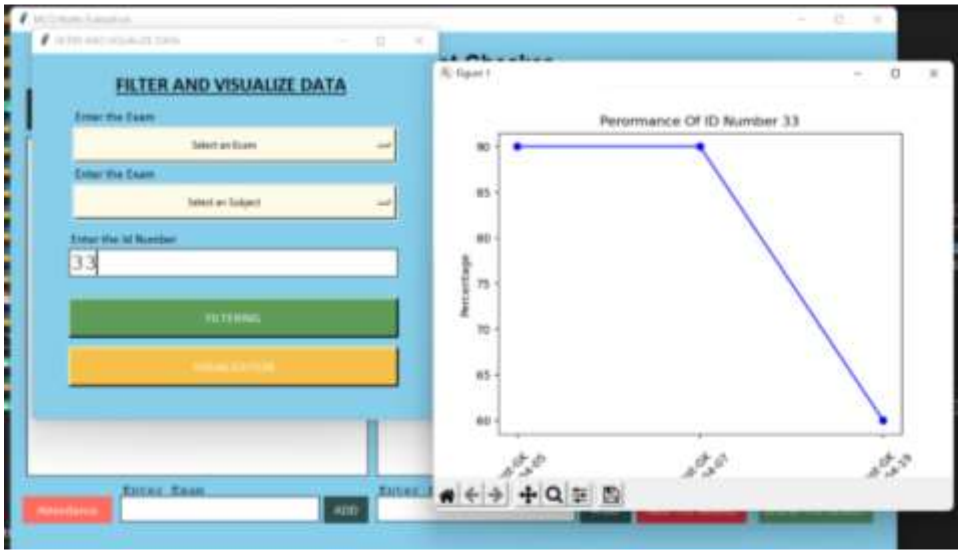


Fig 4.27 Line Graph

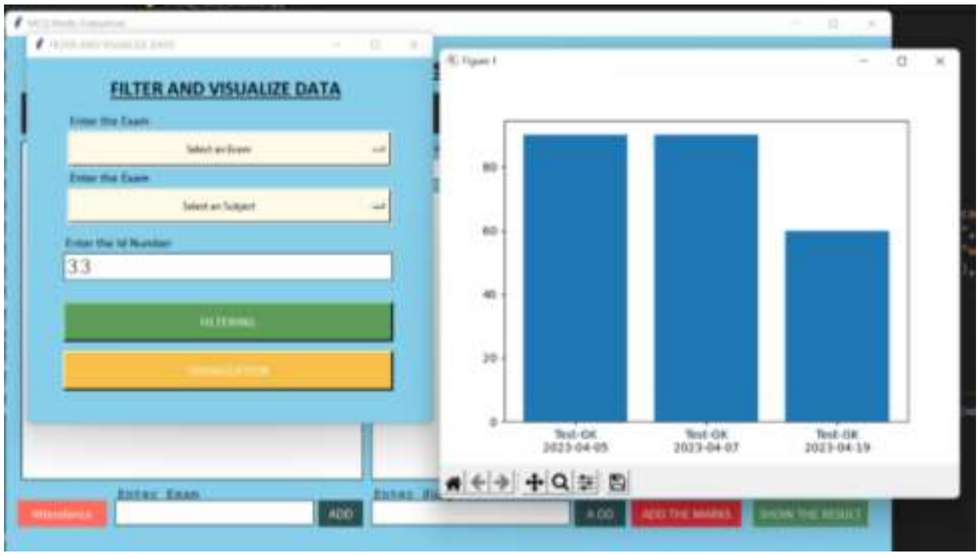


Fig 4.28 Bar Graph

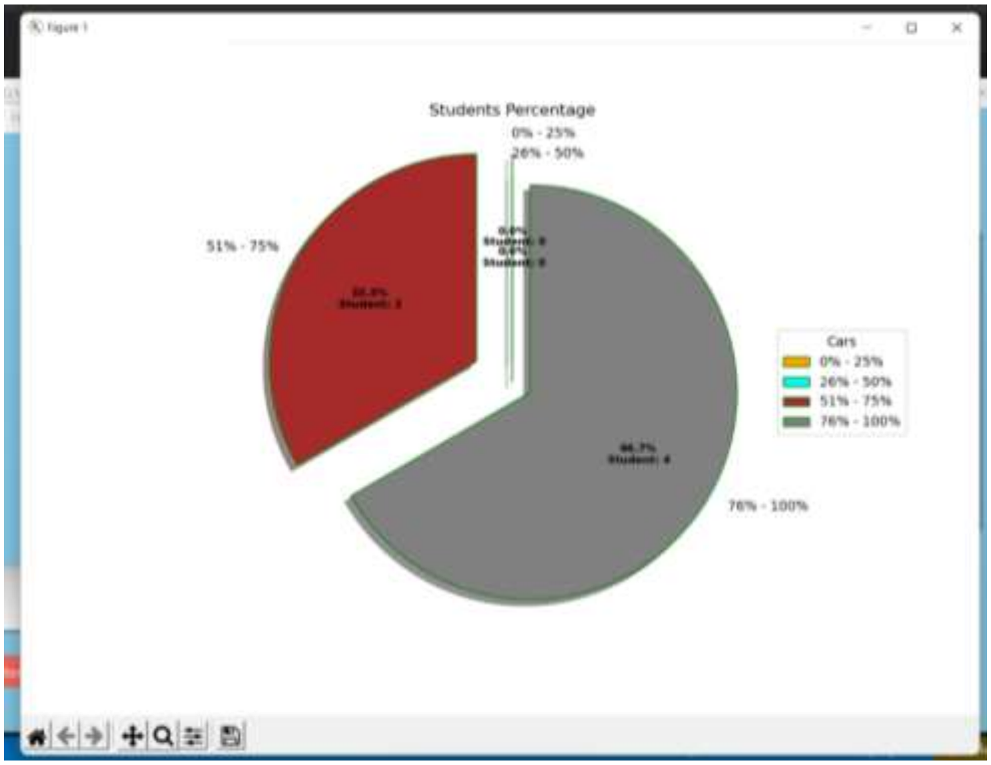
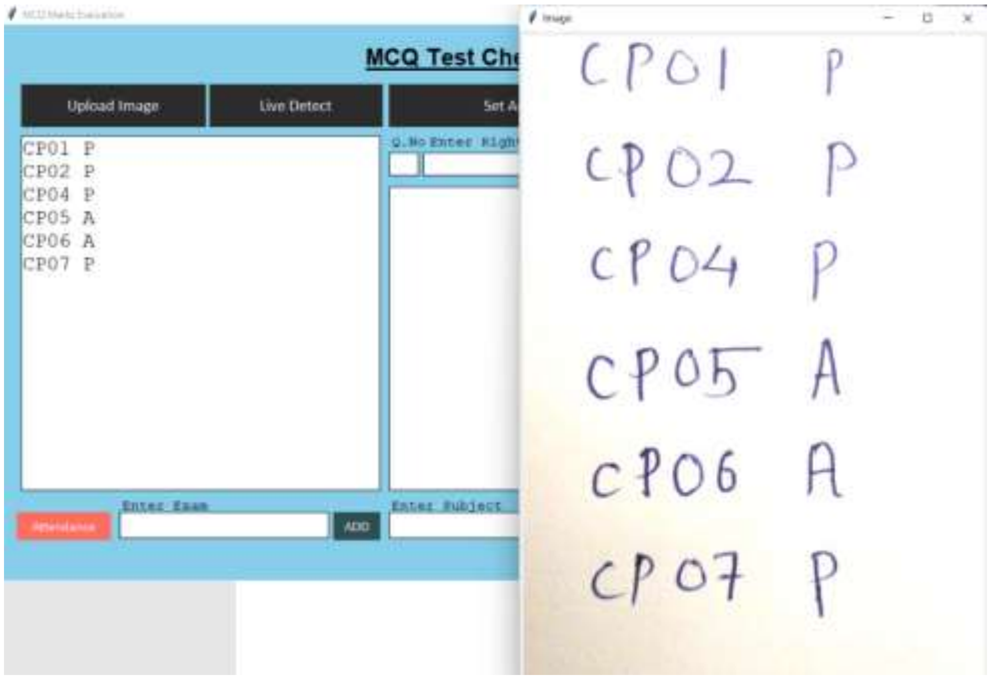


Fig 4.29 Pie Chart

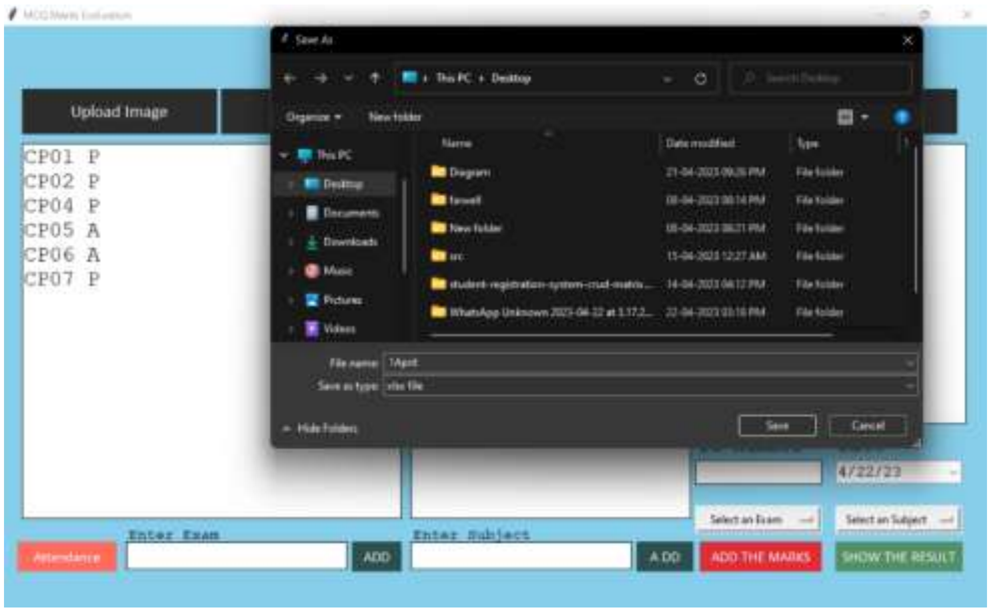
4.5 Digitalize Attendance

If a user wants to digitalize attendance, user have to upload an attendance sheet and then click attendance button to convert it to excel or pdf format.



```
CP01 P
CP02 P
CP04 P
CP05 A
CP06 A
CP07 P
```

Fig 4.30 Digitalize Attendance

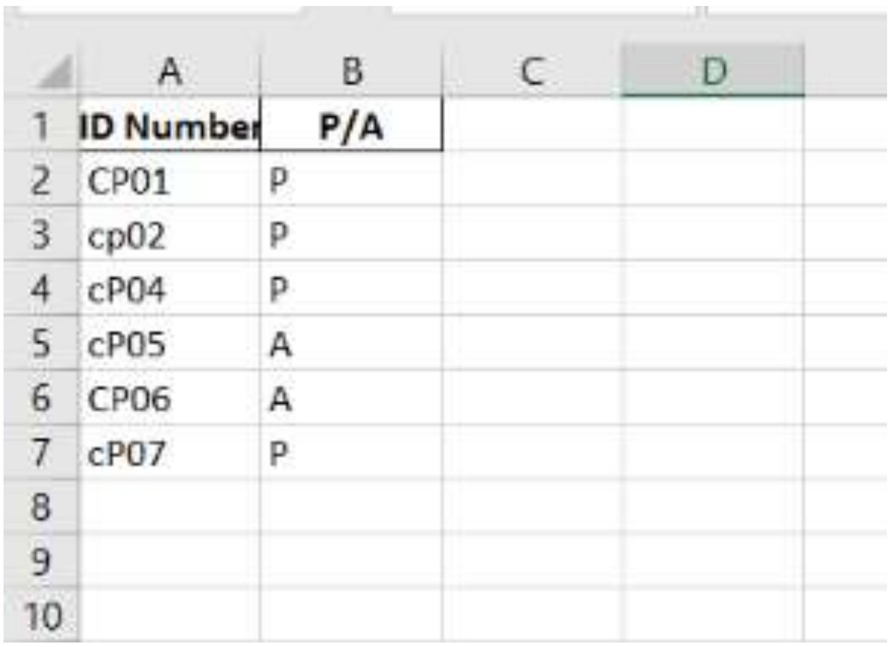


```
CP01 P
CP02 P
CP04 P
CP05 A
CP06 A
CP07 P
```

File name: 1April
Save as type: xlsx file

4/22/23

Fig 4.31 Save Attendance



	A	B	C	D
1	ID Number	P/A		
2	CP01	P		
3	cp02	P		
4	cP04	P		
5	cP05	A		
6	CP06	A		
7	cP07	P		
8				
9				
10				

Fig 4.32 Digitalize Attendance into excel format

5. Conclusions and Future Scope

Conclusions

In conclusion, the "Handwritten Text Recognizer" project focuses on developing an innovative approach that leverages deep learning techniques to accurately recognize and convert handwritten text into digital text. This project has various applications, such as digitizing historical documents, document digitalization, healthcare and pharmaceutical, and digitizing handwritten notes and attendance sheets. The system can also be used for language translation, checking MCQ tests, and storing data in a database, which can be filtered and visualized to analyze student results/marks.

The project involves creating a large-scale annotated dataset of handwritten text samples, training and evaluating deep learning models using state-of-the-art techniques, and conducting extensive experiments to assess the performance and adaptability of the proposed approach. The goal is to develop an HWR system that performs with high accuracy, is adaptable to different writing styles, and can be applied to practical applications.

Overall, this project can significantly contribute to the field of HTR by developing an efficient and effective system that can accurately recognize and convert handwritten text into digital text, ultimately facilitating digitization, translation, and analysis of handwritten documents.

Future Scope

In the future we can increase the accuracy of the model by using larger datasets and more numbers epoch.

We can implement more applications with the help of handwritten text recognition.

We can work on different languages of handwritten text like Hindi, Gujarati Marathi.

References:

[01] Existing Solutions:

[001] ABBYY FineReader PDF **:** https://pdf.abbyy.com/

[002] MYscript : https://www.myscript.com/

[003] microsoft office lens : https://support.microsoft.com/en-us/office/office-lens-for-windows-577ec09d-8da2-4029-8bb7-12f8114f472a

[004] Wondershare : https://pdf.wondershare.com/

[005] GoodNotes 5 : https://apps.apple.com/us/app/goodnotes-5/id1444383602

[02] Python: https://en.wikipedia.org/wiki/Python_(programming_language)

[03] Python logo: shorturl.at/ghlz8

[04] Open cv: https://www.learnopencv.com/histogram-of-orientedgradients/ –

[05] Open cv logo: shorturl.at/jCQR4

[06] Object Detection: https://www.geeksforgeeks.org/detect-an-object-with-opencv-python/

[07] Object Detection logo: shorturl.at/lmDGS

[08] OCR: https://en.wikipedia.org/wiki/Optical_character_recognition

[09] OCR logo: shorturl.at/qK045

[10] Tesseract : https://pytesseract.readthedocs.io/

[11] Tesseract Logo : https://t3.ftcdn.net/jpg/03/43/73/46/360_F_343734651_Bk3iP4m198NJp5ZrZ-tjpwo1PnCtiXplZ.jpg

[12] Tensorflow : https://www.tensorflow.org/

[13] Tensorflow Logo : https://upload.wikimedia.org/wikipedia/commons/thumb/a/ab/TensorFlow_logo.svg/2560px-TensorFlow_logo.svg.png

[14] Keras : https://keras.io/

[15] Keras Logo : https://keras.io/img/logo.png

[16] Image Processing : https://docs.opencv.org/4.x/d2/d96/tutorial_py_table_of_contents_imgproc.html

[17] Image Processing logo : https://qualitastech.com/image-processing/image-processing-and-its-steps/

[18] Jupyter Notebook : https://jupyter.org/try

[19] Jupyter Notebook logo : https://jupyter.org/assets/share.png

[20] VS Code : https://code.visualstudio.com/

[21] VS Code logo : \https://code.visualstudio.com/brand

[22] Drow io. : https://app.diagrams.net/

[23] CNN Image : https://production-media.paperswithcode.com/method_collections/cnn.jpeg

[24] RNN Image : https://www.simplilearn.com/ice9/free_resources_article_thumb/Simple_Recurrent_Neural_Network.png

[25] MNIST Dataset : https://storage.googleapis.com/tfds-data/visualization/fig/mnist-3.0.1.png

[26] IMA Dataset : https://fki.tic.heia-fr.ch/databases/iam-handwriting-database