International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 03 March 2025
Page No: 570 - 576
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0080

Precision: Paper Correction System Using AI

Ms.M Kowsalya[1], S Gowtham[2], S Guhan[3], S Logeshwar[4]
[1]Assistant Professor, Department of CSE, Erode Sengunthar Engineering College, Erode - 638057, India
[2,3,4]Students, Department of CSE, Erode Sengunthar Engineering College, Erode- 638057, India
Emails: *kowsalyamcse@esec.ac.in[1], gowthamgowtham7672@gmail.com[2], guhans983@gmail.com[3], logesh8211@gmail.com[4]*

Abstract
The education system has experienced a significant transformation due to technological advancement. Technology has made teaching much more interesting and informative through projectors, online tutorials, teaching videos, and animations. In classrooms, technology is now widely used in order to enhance learning experiences for students. The evaluation, however, tends to be traditional as exam papers are manually graded with a heavy reliance on the teacher's judgment. Replacing the manual way of grading with machine learning could very well tackle the inconvenience and errors human handling brings about. This project aims to address the massive amount of time and energy that are being applied to the manual correction of exam papers. Also, an online evaluation of answer sheets would provide a quicker method of assessment compared to offline grading.
Keywords: *Online Evaluation, Machine Learning, Automated Correction, Education Technology, Exam Paper Assessment.*

1. Introduction
The traditional method of evaluating student's descriptive answers requires faculty members to manually assess each response. This approach is not only time-consuming but also prone to inconsistencies and subjectivity, as different evaluators may assign varying marks to similar answers. Moreover, human emotions, fatigue, and workload can impact the fairness and accuracy of grading. As the number of students increases, the burden on faculty members grows, further slowing the evaluation process and increasing the likelihood of errors. To address these challenges, this study proposes an automated evaluation system using Natural Language Processing (NLP) and Optical Character Recognition (OCR) technologies. OCR, specifically the Google's open-source Tesseract engine, converts handwritten or printed text into an editable digital format, allowing for the seamless text processing.Python-Tesseract extends this functionality by supporting multiple image formats, making it adaptable to various answer script inputs. Additionally, it enhances the versatility of the system by allowing it to process different types of scanned documents, ensuring smooth integration with various academic assessment workflows.By leveraging these capabilities, the proposed approach enables efficient and reliable text extraction, forming a crucial foundation for automated grading. NLP techniques further enhance the system by enabling the intelligent assessment of textual responses, ensuring consistency, accuracy,and unbiased grading. These techniques analyze the system by enabling the intelligent assessment of textual responses, ensuring consistency, accuracy, and unbiased grading. This process ensures a fair and objective evaluation while minimizing discrepancies that may arise due to human subjectivity. By incorporating NLP-driven methods, the system can efficiently process large volumes of answer scripts, making the grading process more scalable and efficient for academic institutions. This research aims to improve the efficiency of student answer evaluation by minimizing human effort while maintaining a fair and scalable grading system. By integrating OCR and

NLP, the proposed approach ensures uniform assessment criteria, eliminates evaluator bias, and significantly reduces grading time. The originality of this work lies in leveraging AI-driven text recognition and language processing to automate and optimize the grading process, making it more reliable and scalable for academic institutions. [1-5]

1.1.Challenges of Manual Evaluation

This section discusses the challenges of manual evaluation, highlighting its inefficiencies and inconsistencies due to human subjectivity and workload constraints. Various studies have emphasized the need for automation in grading to improve accuracy and reduce bias.

1.2.Role of AI in Automated Evaluation

This section explores the technological advancements in OCR and NLP, particularly focusing on the role of Tesseract OCR and Python-Tesseract in digitizing handwritten responses. Additionally, it outlines how NLP techniques facilitate automated text analysis to ensure fair and precise evaluation.

2. Architecture

The proposed online evaluation system is designed to streamline the grading process for descriptive answers. It features a User Interface (UI) with two main portals: the Student Portal for submitting answer scripts and the Faculty Portal for managing evaluations and results. Students can upload answer scripts in various formats through the Input Module, while an Image Preprocessing Component enhances image quality for accurate recognition. The system uses Tesseract OCR, powered by Python-Tesseract, to convert handwritten or printed text into machine-readable formats to analysis the images. An NLP Module further refines the evaluation by analyzing extracted text, identifying key concepts, and scoring responses based on predefined criteria. A Database securely stores student responses and results, ensuring organized data management. The Automated Scoring System quickly generates scores, while the Feedback Generation Feature provides insights into student performance. Faculty members can review results through the Reporting Module, which includes a Results Dashboard and Performance Reports for students. Additionally, an Admin Module allows for user management, configuration of evaluation criteria, and system settings, making the evaluation process efficient, fair, and scalable.

3. Text Recognition

3.1.Handwriting Recognition

Offline handwriting recognition uses Optical Character Recognition (OCR) to convert images of handwritten, printed, or typed text into editable text. Popular OCR engines like Tesseract, GOCR, and Transym each have their strengths. The paper "An Overview of the Tesseract OCR Engine" explores Tesseract's capabilities, including an experiment on reading car license plates with preprocessing techniques.

3.2.Automatic Evaluation

The paper "A Dynamic Semantic Space Modelling Approach for Short Essay Grading" explores various NLP techniques, including tokenization, part of speech (POS) tagging, stemming, removal of stop words, correction of spelling errors, and handling of case sensitivity in text. Objective test evaluations typically require a single, correct answer from a set of possible responses.The paper titled "Automatic Answer-Sheet Evaluation: An OCR-Based Approach" discusses a method for automatically evaluating objective answer sheets using computer-based Optical Character Recognition (OCR) technology.The study "An Automatic Answering System with Template Matching" introduces an automated system designed to respond to frequently asked questions (FAQs) in both English and SMS language. The system can detect spelling mistakes, particularly errors involving incorrect vowel usage. It relies on pre-stored question templates and corresponding answers, which must be manually uploaded into the system. [6-10]

4. Image Acquisition Process

Image acquisition involves capturing a digital or digitized image as input. The most common devices used for this purpose are electronic tablets or digitizers, which typically use a digital pen. Handwritten characters can also be captured using

IRJAEH

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 03 March 2025
Page No: 570 - 576
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0080

other methods, such as scanners or photographs. The process of converting a document into an electronic format is achieved through scanning, which creates a digital image of the document. These images are typically in black and white and can be saved in various formats, including JPEG, BMP, or others. Once captured, the image is forwarded to subsequent stages for further processing. Image acquisition, or digitization, generates the digital image that is then sent to the pre-processing phase for additional handling.

4.1.Handwriting Recognition System

Handwriting recognition is a technique that allows a computer system to identify characters and symbols written in natural handwriting. In offline handwriting recognition, the completed text is provided in the form of an image. These images, whether of typed, handwritten, or printed text, can be captured from scanned documents or photos. The Optical Character Recognition (OCR) mechanism is then used to convert the image into editable text. This discusses the use of Google's open-source Optical Character Recognition (OCR) software, Tesseract. Tesseract is an OCR engine compatible with various operating systems and is available as free software under the Apache License. Since 2006, Google has sponsored its development. At that time, Tesseract was regarded as one of the most accurate open-source OCR engines available. Python-Tesseract is a Python library that serves as a wrapper for the Tesseract OCR Engine, enabling it to recognize and extract text embedded in images. In addition to its role as a wrapper, Python-Tesseract can function as a standalone script to invoke Tesseract, supporting various image formats like JPEG, PNG, GIF, BMP, TIFF, and others. In contrast, the Tesseract OCR engine by default only supports TIFF and BMP formats.

5. System Design

Preprocessing is the process of transforming data into a format that can be understood and processed by a computer. This step is crucial as it prepares raw data by cleaning, normalizing, and structuring it in a way that enhances the accuracy and efficiency of subsequent computational tasks. Proper preprocessing ensures that the system can effectively interpret and analyze the data. preprocessing phase typically involves removing irrelevant or noisy data, handling missing values, and converting text into machine-readable formats. Figure 1 shows Answer Sheet Evaluation System

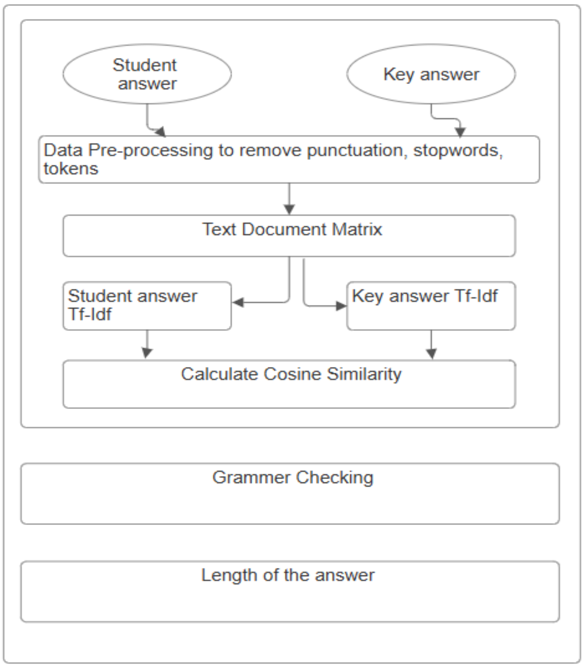

Figure 1 Answer Sheet Evaluation System

5.1.Database

The student's final score is determined by combining the marks allocated for individual sections, including keywords, grammar, and answer length. The computed score is then stored in the database. The MySQL database for the online evaluation system includes key tables: Students, Faculty, Answer Submissions, Evaluations, and Feedback. Each student can submit multiple answers, while faculty can evaluate these submissions and provide feedback. The Answer Submissions table tracks submission details, and the Evaluations table stores scores and comments from faculty. The Feedback table allows for detailed insights on student performance. This structure ensures efficient data management and retrieval for the evaluation process.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 03 March 2025
Page No: 570 - 576
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0080

5.2. Architectural

The architectural design provides an overview of the system's structure, describing its components and the flow of control and data between them. Arrows represent connections, while rectangular boxes denote functional units. Figure 5.1 illustrates the architectural diagram for this project, showcasing the entire process, from uploading question papers to evaluating student answer scripts. Figure 2 shows Architectural Diagram [11-15]

Figure 2 Architectural Diagram

The Figure 2 presents the architectural diagram for this project, depicting the complete workflow, starting from uploading question papers and progressing through the various stages of processing and evaluation, ultimately leading to the assessment of student answer scripts. The entire flow of this project-from uploading question papers to different stages of processing and evaluation until the assessment of student answer scripts and generation of final results-is depicted in Figure 5.3: the architecture plan for this project. Finally generates a similarity score and grades, followed by performance report generation.

5.3. Use case Diagram and Description

A use case defines a specific functionality that a system performs through its interactions with actors. It details the system's processes, including individual use cases and the roles of associated actors. For instance, the use case for uploading question papers involves faculty members entering relevant question details. Figure 3 shows Use Case Diagram for Question Paper, Figure 4 shows Use case diagram for Evaluation This use case illustrates the evaluation is process, where student answer scripts are assessed according to predefined criteria, and the results are displayed.

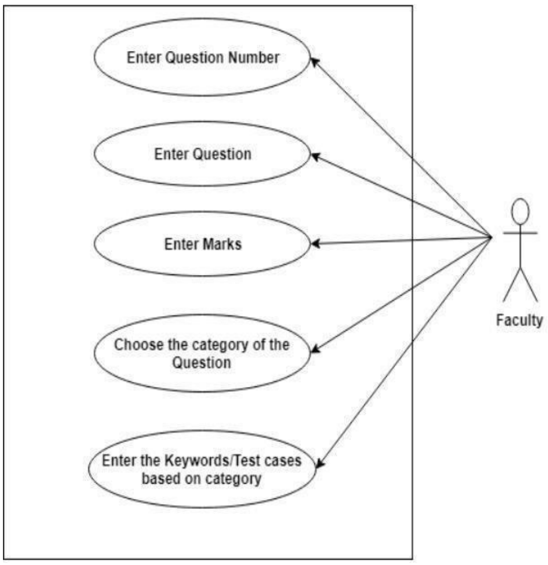

Figure 3 Use Case Diagram for Question Paper

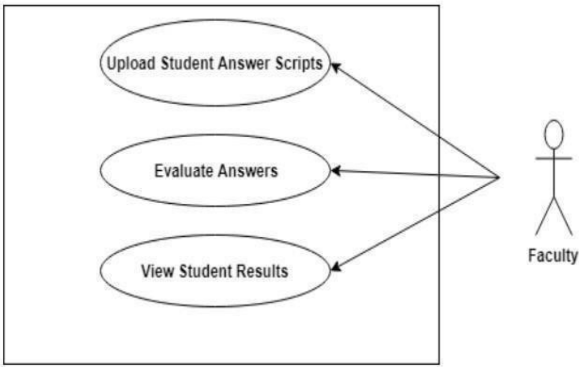

Figure 4 Use case diagram for Evaluation

5.4.Data Flow Diagram

A Data Flow Diagram (DFD) visually represents how data moves through a system, showing processes, data stores, external entities, and data flows. It provides a high-level overview of the system, helping identify inefficiencies and areas for improvement. DFDs are hierarchical, starting with a context diagram and breaking down into more details for deeper analysis. They are widely used in system design to enhance The Fig shows the data flow between each component in the system. Faculty uploads the question papers and key answers to the Question module. Faculty also uploads the answer scripts to the answer module. These processed answer scripts are given to the evaluate module. Then the

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 03 March 2025
Page No: 570 - 576
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0080

results are displayed in the result module. The data moves through the system, connecting different components. First, the faculty uploads question papers and key answers to the Question Module and submits answer scripts to the Answer Module. These answer scripts are then processed and sent to the Evaluate Module, where they are assessed. Once the evaluation is complete, the results are generated and displayed in the Result Module. Figure 5 shows Data Flow Diagram

Table1 Test Case

Test Numbers	Test Case ID	Test Case	Expected Results	Status
1	UT_1	Uploading questions into system	Question specific details are stored in the database	Uploaded
2	UT_2	Uploading answer script into system	Student answer has to be uploaded to the database	Uploaded
3	UT_3	Converting image to text	The image has to be converted into a text file	Converted
4	UT_4	Finding similarity between student answer and key answer	The similarity between answer and key answer is calculated out of 100	Evaluated
5	UT_5	Display the student result	To display the student results	Displayed
6	UT_6	Clear answer script from the database	To delete previous data	Deleted
7	UT_7	Delete questions from the system	To delete the questions from the system	Deleted

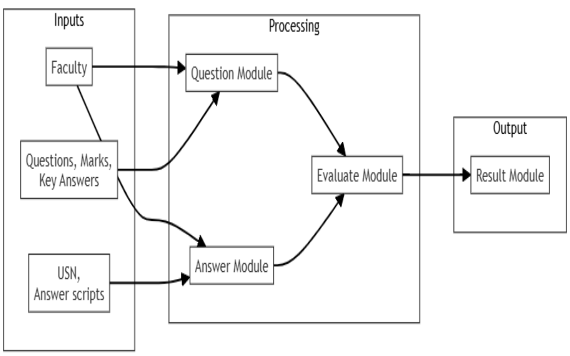

Figure 5 Data Flow Diagram

5.5.Test Case

In fact, testing has been defined in terms of actual results against the anticipated results and includes all that is required to know whether the system or application intended works. This is one aspect of testing, the most crucial process that can cope with the revelation of bugs or gaps or missing requirements relative to the initially set design and objectives. In addition to detecting bugs,Testing would be an assessment of the product's , reliability, and performance. This part of the process is vital for the development life cycle of the product and leads to a more robust and friendly final product.

Conclusion

This system is designed to evaluate subjective answers by comparing a student's response to the key answer provided by the university. Marks are allocated based on the alignment between the student's answer and the model answer. This approach can greatly benefit educational institutions by saving time and reducing the effort of manually checking large volumes of papers. The evaluation system assigns grades according to the percentage of similarity with the key answer.

Future Work

Future work will focus on developing a more advanced assessment system that can automatically evaluate a wide range of student responses, including mathematical formulas, code written in different programming languages, and graphical representations. The goal is to enhance the system's performance, accuracy, and efficiency, providing more reliable and insightful feedback to support student learning. Additionally, the system will be designed to adapt to diverse subjects and evaluation criteria ensuring broader applicability in educational settings.

Reference

[1]. Jianbo Xu, Wenhan Ding, and Hanbing Zhao, Based on improved edge detection algorithm for English text extraction and restoration from color images, IEEE Sensors Journal, vol. 03,2020.

[2]. Langcai Cao, Hongwei Li, Rongbiao Xie, and Jinrong Zhu, A text detection algorithm for images of student exercises based on CTPN and enhanced YOLOv3,IEEE Access, vol. 08, Aug. 2020.

[3]. Amirreza Fateh, Reza Tahmasbi Birgani, Mansoor Fateh, and Vahid Abolghasemi, Advancing multilingual handwritten numeral recognition with attention-driven transfer learning,IEEE Access, vol. 12, Mar. 2024.

[4]. Chucai Yi and Yingli Tian, Scene text recognition in mobile applications by character descriptor and structure configuration IEEE Transactions on Image Processing, vol. 23, no. 7, pp. 2972-2985, Jul. 2014.

[5]. Narayana Darapaneni, Sai Venkateshwaran, Sumathi,Gunasekaran,Malarvizhi,Subramaniyan,Nandini Ravi, and Asha,Handwritten form recognition using artificial neural network,in Proceedings of the 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), 2020.

[6]. Bilal Khan, Zohaib Ali Shah, Muhammad Usman, Inayat Khan, and Badam Niazi,Exploring the landscape of automatic text summarization: A comprehensive survey,IEEE Access, vol. 11,Oct. 2023.

[7]. Karina Korovai, Dmytro Zhelezniakov, Oleg Yakovchuk, Olga Radyvonenko, Nataliya Sakhnenko, and Ivan Deriuga, Handwriting enhancement: Recognition-based and recognition-independent approaches for on-device online handwritten text alignment, IEEE Access, vol. 12, pp. 99334-99345, Jun. 2024.

[8]. Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, and Mueen Uddin, Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR), IEEE Access, vol. 8, pp. 142642-142652, Jul. 2020.

[9]. Qiaokang Liang, Shao Xiang, Yaonan Wang, Wei Sun, and Dan Zhang, RNTR-Net: A robust natural text recognition network, IEEE Access, vol. 8, pp. 7719-7731, Jan. 2020.

[10]. Daniel Keysers, Thomas Deselaers, Henry A. Rowley, Li-Lun Wang, and Victor Carbune, Multi-language online handwriting recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1180-1192, Jun. 2017.

[11]. K. S.Raghunandan, Palaiahnakote

e ISSN: 2584-2137
Vol. 03 Issue: 03 March 2025
Page No: 570 - 576
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0080

Shivakumara, Sangheeta Roy, G. Hemantha Kumar, Umapada Pal, and TongLu, Multi-script-oriented text detection and recognition video/scene/born digital images,IEEE Transactions on Circuits and Systems for Video Technology, vol. 05, Nov -2018.

[12]. Tejasvee Bisen, Mohammed Javed, P. Nagabhushan, and Osamu Watanabe, Segmentation-less extraction of text and non-text regions from JPEG 2000 compressed document images through partial and intelligent decompression,IEEE Access, vol. 11, pp. 20673-20685, Feb. 2023

[13]. Syed Yasser Arafat and Muhammad Javed Iqbal,Urdu-text detection and recognition in natural scene images using deep learning,IEEE Access, vol. 8, pp. 96787-96795, May 2020.

[14]. Yi-Feng Pan, Xinwen Hou, and Cheng-Lin Liu, A hybrid approach to detect and localize texts in natural scene images,IEEE Transactions on Image Processing, vol. 20, no. 3, pp. 800-810, Mar. 2011.

[15]. Ibrar Hussain, Riaz Ahmad, Siraj Muhammad, Khalil Ullah, Habib Shah, and Abdallah Namoun, PHTI: Pashto handwritten text imagebase for deep learning applications, IEEE Access, vol. 10, pp. 113149-113162, Oct. 2022.