A Major Project Report (Stage – I)

On

“PAPER EVALUATION USING ARTIFICIAL INTELLIGENCE”

Submitted in Partial Fulfilment of the Academic Requirement for the Award of the Degree

of

BACHELOR OF TECHNOLOGY

In

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Submitted By:

M.SAICHARAN 22R01A0540

Under the Esteemed Guidance of

S. Shilpa

(Assistant Professor, Dept. of CSE)

CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMUS)

Approved by AICTE, Affiliated to JNTUH, Accredited by NAAC with ‘A+’ Grade

Kandlakoya(v), Medchal Dist.- 501401

www.cmrithyderabad.edu.in

2025-2026

CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMUS)

Approved by AICTE, Affiliated to JNTUH, Accredited by NAAC with 'A+' Grade

Kandlakoya(v), Medchal Dist.- 501401

www.cmrithyderabad.edu.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that a Major Project (Stage-1) entitled with "**Paper Evaluation using Artificial Intelligence**" is being submitted by

M.SAICHARAN 22R01A0540

To JNTUH, Hyderabad, in partial fulfilment of the requirement for award of the degree of **B. Tech** in **Computer Science and Engineering** and is a record of a bonafide work carried out under our guidance and supervision. The results in this project have been verified and are found to be satisfactory. The results embodied in this work have not been submitted to have any other University for award of any other degree or diploma.

Signature of Guide	**Signature of Project Coordinator**	**Signature of HOD**
S. Shilpa	**Mr. P. Krishnaiah**	**Dr. K. Pradeep Reddy**
(Assistant Professor, CSE)	(Assistant Professor, CSE)	(Head of Department)

Signature of External Faculty

ACKNOWLEDGEMENT

We are extremely grateful to **Dr. M Janga Reddy**, Director, **Dr. B. Satyanarayana,** Principal and **Dr. K. Pradeep Reddy,** Head of Department, Department of Computer Science and Engineering, CMR Institute of Technology for their inspiration and valuable guidance during entire duration.

We are extremely thankful to **Mr. P. Krishnaiah,** Assistant Professor, Major Project Coordinator and internal guide **"S. Shilpa",** Assistant Professor Dept. of Computer Science and Engineering, CMR Institute of Technology for their constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided directly or indirectly for every step towards success.

M.SAICHARAN **22R01A0540**

ABSTRACT

"**Paper Evaluation Using AI**" is a comprehensive project aimed at transforming the traditional approach to academic paper assessment by integrating the capabilities of Artificial Intelligence. For decades, manual paper evaluation has been the primary method used in educational and research institutions. Although effective in many cases, manual evaluation is inherently slow, labour-intensive, and susceptible to human error. Factors such as evaluator fatigue, inconsistent judgment, unintentional bias, and variation in interpretation often lead to unreliable scoring. Furthermore, the rising number of students and research submissions has created a need for a more scalable and efficient evaluation process. This project addresses these concerns by proposing an AI-based system that ensures accuracy, transparency, and uniformity across all evaluated papers.

The system incorporates advanced **Natural Language Processing (NLP)** and **Machine Learning (ML)** techniques to analyze the content of each paper in a detailed and structured manner. Instead of relying solely on grammar checks or keyword matching, the AI system evaluates multiple qualitative dimensions that reflect true academic writing quality. These include **grammatical correctness, clarity, structure, coherence, relevance to the topic, semantic consistency, originality of ideas, sentence formation, logical flow, and vocabulary richness**. By assessing each of these parameters holistically, the system generates a score that closely aligns with human assessment standards.

A dataset of **expert-reviewed papers** is used to train the model. This allows the AI system to understand how experienced evaluators judge academic writing and what patterns or features indicate high-quality responses. The NLP pipeline employs techniques such as **tokenization, lemmatization, syntactic parsing, contextual word embeddings, semantic similarity analysis, and topic detection** to interpret both the structure and meaning of the text. Meanwhile, ML algorithms like **K-Nearest Neighbors (KNN)** support accurate score prediction by comparing new submissions with known scoring patterns.

Experimental results from the project indicate that the AI system achieves a **strong positive correlation with human evaluations**. The automated evaluation method reduces overall grading time by more than **70%**, making it highly beneficial in situations where thousands of papers must be assessed within short deadlines. In addition, the system significantly reduces subjectivity, ensuring fair and unbiased evaluation for all students regardless of evaluator mood, experience, or workload.

The project is implemented using **Python**, with support from libraries and frameworks such as **NLTK, spaCy, Scikit-learn, TensorFlow, Pandas, and NumPy**. **MongoDB** is used for efficient storage and retrieval of evaluation data. OCR capabilities are included to allow the system to process scanned documents without requiring manual transcription.

INDEX

LIST OF FIGURES

1. INTRODUCTION

In today's education system, the number of students, assignments, and written examinations is increasing every year. Teachers and exam evaluators often spend long hours reviewing answer papers, checking grammar, understanding the content, and giving marks. This manual process is slow, tiring, and sometimes inconsistent because different evaluators may interpret the same answer differently. Human factors such as fatigue, stress, or time pressure can also influence the quality of evaluation. As a result, many institutions face difficulties in giving quick, fair, and uniform results to all students. With growing academic demands, it has become important to look for smarter and more efficient ways to evaluate written papers.

At the same time, technology—especially Artificial Intelligence—is developing very fast. AI systems are now able to understand human language, identify patterns, and make decisions based on data. Natural Language Processing (NLP), which is a branch of AI, allows computers to read and analyze text in a meaningful way. Machine Learning models can learn from examples and improve their accuracy over time. These technologies are already being used in areas like grammar checking, content analysis, summarization, and even professional writing tools. This shows that AI has the ability to support and improve many tasks that previously required human effort.

The project **"Paper Evaluation Using AI"** is developed with the aim of using these technologies to make the evaluation process faster, fairer, and more reliable. The system is designed to read answer papers, examine the quality of writing, check grammar and sentence structure, understand the relevance of content, and give a score based on patterns learned from expert-evaluated papers. By reducing human involvement in repetitive tasks, the system helps teachers save time and focus more on teaching and interacting with students. It also removes subjectivity and ensures that every student receives equal and unbiased evaluation.

Another advantage of this system is that it can handle large volumes of papers without getting tired or making errors. With the help of OCR (Optical Character Recognition), it can even read scanned or handwritten papers, making it suitable for real classroom environments. As educational institutions grow and examinations become more demanding, AI-based evaluation can play an important role in improving the overall assessment process.

Overall, this project introduces a modern and practical solution to the challenges faced in manual paper checking. By combining simplicity with intelligent automation, **"Paper Evaluation Using AI"** aims to support teachers, speed up result processing, and bring more fairness and consistency to academic evaluation.

2. OBJECTIVES OF THE PROJECT

The main objective of this project is to develop an AI-based Paper Evaluation System that can automatically assess student answer scripts with accuracy, fairness, and consistency. The system aims to reduce the manual workload of teachers, speed up the evaluation process, and improve the quality of overall academic assessment. By integrating machine learning and natural language processing techniques, the project intends to create an efficient, reliable, and scalable solution for modern educational institution

Primary Objectives

1. **Automate the evaluation of student answer scripts using AI**
   This objective focuses on creating an AI-driven system that can read, interpret, and understand written student responses with high accuracy. The model will evaluate answers based on content correctness, relevance to the question, clarity of explanation, and completeness of information. By training the system on a large dataset of well-graded responses, the project aims to make the evaluation process faster, more consistent, and free from human fatigue or scoring variations. This automation helps ensure that every student receives fair and unbiased marking.

2. **Classify student responses into structured scoring categories**
   The project aims to develop a reliable classification method that groups answers into predefined quality levels. The AI model will categorize responses into one of the following groups:
   •Excellent
   •Good
   •Satisfactory
   •Needs Improvement
   This classification allows teachers to quickly identify the strengths and weaknesses of each response. It also ensures that evaluation remains uniform, especially when handling large volumes of scripts. The categories act as a standardized framework that simplifies the grading process and brings clarity to how marks are distributed.

3. **Provide instant scoring and meaningful feedback for improved learning**
   This objective focuses on generating immediate results once an answer is submitted. The system will not only provide a score but also offer specific feedback on where the student performed well and where improvement is needed. Such instant feedback supports better learning, especially during

practice sessions, online assessments, or formative evaluations. It also significantly reduces the workload on teachers by automating repetitive tasks and speeding up overall result processing.

4. **Enhance AI accuracy through continuous training and ensemble techniques** To ensure reliability and high accuracy, the project will explore multiple AI models and deep-learning algorithms. A stacked ensemble approach will be used to combine the strengths of individual models, leading to more stable predictions. This objective includes training the system on diverse writing styles, different subjects, and multiple answer formats. Continuous model updates and error-analysis cycles will further improve performance and ensure the system adapts to new types of questions or writing patterns.

5. **Identify key writing features that influence the scoring process** This objective focuses on analyzing the most important factors that affect how a student's answer is evaluated. These may include keyword usage, conceptual depth, sentence clarity, grammatical accuracy, organization of ideas, and overall coherence. By identifying these features, the system becomes more transparent and easier for educators to understand. Such insights also help teachers guide students more effectively by highlighting which skills need improvement for higher academic performance.

6. **Highlight student performance patterns for academic planning and decision-making** The system will analyze large sets of evaluated answers to detect common mistakes, recurring weaknesses, and overall performance trends. This allows teachers and institutions to understand which topics students struggle with the most and where additional teaching support is required. By identifying these patterns, schools and colleges can plan remedial sessions, modify teaching strategies, and improve curriculum delivery. This objective ultimately helps in achieving better learning outcomes and more efficient academic management.

3. LITERATURE SURVEY

A literature survey is essential to understand how previous researchers have approached automated paper evaluation and what limitations still exist. Over the years, different technologies such as OMR, OCR, machine learning, deep learning, and natural language processing (NLP) have been used to improve the accuracy and speed of answer-sheet evaluation. Each technique has contributed to automation in its own way, but none of them provided a complete and reliable solution for descriptive answer checking. This section reviews the major categories of research done in this field and highlights the gaps that motivated the need for an AI-based paper evaluation system.

3.1 OMR and Image Processing Systems

Early research focused heavily on Optical Mark Recognition (OMR) systems, mainly used for exams with multiple-choice questions. These systems detect filled bubbles and compare them with a predefined answer key.
OMR-based techniques are highly efficient in terms of speed and accuracy, making them suitable for large-scale examinations. Many studies highlight that OMR reduces human error, supports high-volume answer sheets, and provides quick results.

However, the major limitation is that OMR works only for objective-type exams and strictly follows fixed templates. Any deviation in shading, alignment, or paper quality can reduce accuracy. Because of these limitations, researchers realized that OMR alone cannot be used for descriptive or handwriting-based evaluations. This limitation created a need for smarter systems that can understand text, diagrams, or long answers.

3.2 OCR-Enabled Text Digitization Models

As exams began including descriptive questions, researchers moved towards Optical Character Recognition (OCR) to convert handwritten text into machine-readable form. OCR allowed examiners to partially automate evaluation by reading paragraphs or sentence-based answers.

Studies show that OCR works well for printed text but struggles when dealing with handwritten answers due to differences in writing style, spacing, alignment, and clarity. Even advanced OCR techniques such as Tesseract and CNN-based OCR models fail when handwriting is messy or when technical terms are used. Additionally, OCR only extracts text—it does not understand the meaning or correctness of the answer. This made OCR insufficient as a complete solution for answer-sheet evaluation.

3.3 AI and NLP-Based Semantic Evaluation (BERT, GPT, Transformers)

To address the shortcomings of OCR, researchers explored Natural Language Processing (NLP) and transformer-based deep learning models such as BERT, RoBERTa, GPT, and other semantic models. These models are capable of understanding meaning, grammar, coherence, keywords, and context in descriptive answers.

Studies highlight that these models can compare student responses with reference answers and generate similarity scores based on semantic understanding rather than just word matching. Transformer-based models have significantly improved accuracy in automated essay and descriptive answer evaluation.

However, challenges still remain. These systems require large amounts of training data, domain-specific fine-tuning, and careful handling to avoid biased scoring. Additionally, some models behave like "black boxes," making it difficult to explain how they arrived at a particular score.

3.4 Machine Learning Algorithms (SVM, KNN, Decision Trees)

Before the rise of deep learning, researchers used traditional machine learning models for text evaluation. Methods such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees, and Naïve Bayes were commonly applied.

These models used manually extracted features like:

- Sentence length
- Keyword density
- Vocabulary richness
- Grammar indicators
- N-gram patterns

While these models helped automate scoring to some extent, they could not capture deeper meaning or logical flow within answers. Another major drawback was the need for manual feature engineering, which made the process time-consuming and inconsistent. Researchers also noted that these models perform poorly when answer structures vary widely.

3.5 Hybrid OCR + NLP + AI Approaches

Recent research moves towards hybrid systems that combine OCR for text extraction and NLP/AI for semantic scoring. These integrated models use OCR to convert handwritten text into digital form and then apply transformer models to evaluate meaning and correctness.

Studies show that hybrid approaches improve the overall reliability of automated evaluation by using the strengths of multiple technologies. They can handle diverse handwriting, evaluate long descriptive answers, and provide near-human scoring consistency. These systems also reduce the examiner's workload and speed up the evaluation process significantly.

However, this approach still requires high-quality datasets, powerful computational resources, and properly trained models to ensure fairness and accuracy.

3.6 Persistent Research Gaps and Challenges

Although research has progressed significantly, various gaps remain:

- **Handwriting Variability:** messy or cursive handwriting still reduces OCR accuracy.
- **Contextual Understanding:** AI still struggles with subject-specific answers that require expert reasoning.
- **Bias and Fairness:** automated scoring systems can unintentionally favour certain writing styles or vocabulary levels.
- **Dataset Requirements:** high-quality annotated datasets are needed to train robust models.
- **Explainability:** many AI models do not clearly explain why a specific score was given.

These limitations highlight the need for a modern system that integrates improved OCR, advanced NLP, and AI-based evaluation techniques to provide accurate, transparent, and reliable assessment of both objective and descriptive exam answers.

4. SYSTEM ANALYSIS

4.1 Existing System

Traditional methods for evaluating student answer scripts are primarily manual and supported only by basic digital tools. These approaches are limited in speed, consistency, and scalability, and they struggle to provide meaningful feedback or to handle large volumes of scripts reliably.

Characteristics of Existing Systems

- **Manual Evaluation by Teachers:**
  Most institutions still rely on human examiners to read and grade answers. While human graders can judge nuance and reasoning, this method is slow and prone to inconsistency across evaluators and time.
- **OMR-Based Systems for Objective Tests:**
  Optical Mark Recognition (OMR) systems are effective for multiple-choice questions but cannot handle short answers, long descriptive responses, or problem-solving steps.
- **OCR-Only Approaches for Digitization:**
  Some workflows use OCR to convert scanned answer sheets to text. OCR helps digitize content but does not evaluate meaning, correctness, or argument quality. Handwriting variability often leads to OCR errors.
- **Rule-Based or Keyword Matching Tools:**
  Older automated tools use fixed rules or keyword presence to award marks. These systems are unable to interpret context, logic, or conceptual depth.

Limitations of Existing Systems

- **Lack of Semantic Understanding:**
  Current methods cannot reliably assess conceptual correctness, argument coherence, or depth of explanation.
- **Inconsistent Scoring and Bias:**
  Manual marking varies between evaluators; rule-based systems favour certain phrasing and can introduce bias.
- **Poor Handling of Handwriting and Noisy Inputs:**
  OCR errors, illegible handwriting, and diverse answer formats reduce automation accuracy.

- **No or Minimal Feedback for Learners:**
  Traditional grading provides little actionable feedback that students can use to improve
- **Limited Scalability and Speed:**
  Manual correction becomes impractical for large classes or mass examinations due to time and resource constraints.
- **Low Explainability in Some Automated Tools:**
  When simple ML models are used without interpretability methods, it is hard to explain why a particular score was given.

4.2 Proposed System

The proposed system addresses these gaps by combining OCR, advanced Natural Language Processing (NLP), and machine learning (including ensemble methods) to deliver accurate, fair, and explainable automated evaluation for objective and descriptive answers. It supports fast processing, consistent scoring, and constructive feedback generation.

System Objectives

- Automate scoring of objective and descriptive answers while matching human-like judgement.
- Provide consistent, unbiased marking and meaningful feedback.
- Handle scanned/handwritten inputs reliably.
- Offer transparency through explainability tools and reports.
- Scale to large exam volumes and multiple subjects.

Key Components

- **Advanced OCR Module:**
  High-quality OCR (with preprocessing: denoising, line/word segmentation) to convert scanned or handwritten answers to text. Includes handwriting-specific models or post-correction heuristics to reduce errors.
- **NLP & Text Processing Pipeline:**

Tokenization, lemmatization, POS-tagging, syntactic parsing, and contextual embeddings (transformer-based) to extract semantic features, coherence signals, and topical relevance.

- **Feature Extraction & Scoring Engine:**
  Extracts linguistic and domain features — e.g., content coverage vs. reference answer, keyword density, concept presence, logical flow, grammar quality, answer length, and rubric-aligned indicators.
- **Machine Learning & Deep Models:**
  Uses a mix of traditional ML (SVM, KNN, tree-based) and deep models (LSTM, Transformer/BERT variants, CNN for local patterns) to predict scores and classifications.
- **Ensemble / Stacked Meta-Classifier:**
  Combines outputs from multiple base models into a robust meta-classifier to improve accuracy and reduce variance.
- **Feedback Generator:**
  Produces targeted, actionable feedback (grammar tips, missing points, structure suggestions) mapped to rubric items to help students improve.
- **Interpretability & Explainability Tools:**
  Permutation feature importance, SHAP/Integrated Gradients, and attention visualizations to explain model decisions and show which answer parts influenced scores.
- **Admin & Reviewer Interface:**
  Web-based UI for uploading scripts, reviewing AI-suggested scores, correcting where needed, and retraining models on corrected examples.
- **Database & Storage:**
  Secure storage for scanned images, extracted text, scoring histories, rubrics, and training datasets (e.g., MongoDB or similar).

Data Sources & Training Materials

- **Expert-graded answer datasets:** historical graded papers and rubrics for supervised training.
- **Domain-specific reference answers / marking schemes** for comparison and semantic alignment.
- **Pretrained language models** fine-tuned on academic answer corpora to boost domain understanding.

Key Features & Advantages

- **Hybrid OCR + NLP approach:** converts handwriting to text and applies semantic evaluation for descriptive responses.
- **Support for multiple question types:** MCQs, short answers, long descriptive answers, and stepwise problem solutions.
- **High throughput & reduced reviewer time:** evaluation time reduced substantially compared to fully manual marking.
- **Explainable and fair scoring:** model interpretability reduces "black box" concerns and allows post-hoc checks for bias.
- **Continuous learning:** system can be updated with corrected outputs to improve over time.
- **Analytics & Reporting:** dashboards to show class performance, common errors, question-wise difficulty, and trends.

Limitations & Mitigation Strategies

- **OCR errors with poor handwriting:** mitigate via handwriting-adapted OCR models, human-in-the-loop verification, and confidence thresholds.
- **Domain adaptation needs:** fine-tune models for each subject or course and maintain subject-specific rubrics.
- **Potential bias:** use diverse training data, bias auditing, and explainability checks.
- **Resource requirements:** plan for computational resources and incremental deployment (pilot → scale).

5. REQUIREMENT SPECIFICATION

The requirement specification defines the essential needs of the system. It includes functional and nonfunctional requirements, along with the hardware and software necessary for implementing the climatebased cropland suitability prediction system.

Functional Requirements

- Import and preprocess climate & terrain datasets
- Classify cropland type
- Train ML models on historical data
- Forecast cropland suitability for 2050 under SSP pathways
- Generate probability heatmaps
- Provide feature-importance analysis

Non-Functional Requirements

- High scalability for geospatial datasets
- High accuracy and reliability of ML models
- Interpretability of predictions
- Efficient runtime for ensemble forecasting
- Data consistency across climate models

Hardware Requirements

The hardware requirements depend on the size of datasets and complexity of ML models.

Minimum Recommended:

- **RAM:** 16 GB or more
- **CPU:** Multi-core processor
- **Storage:** Adequate space for climate datasets (tens of GBs)

Reference System Used in Research Paper:

- 1 TB RAM
- Intel Xeon Processor
- NVIDIA RTX GPU

Software Requirements

The system requires the following software tools and libraries:

Programming Language

- Python

Machine Learning Libraries

- TensorFlow or PyTorch
- Scikit-learn

Geospatial Processing Libraries

- NumPy, Pandas, Matplotlib - for data handling and visualization

Database

- SQLite - lightweight

6. PROPOSED METHODOLOGY

1. Data Collection & Pre-Processing

The system begins by collecting scanned images of students' handwritten or printed answer sheets. These images are prepared for evaluation through several enhancement techniques:

- **Grayscale Conversion:** Reduces the image to shades of gray, making text easier to isolate.
- **Noise Removal:** Eliminates smudges, shadows, ink bleed, or artifacts that could interfere with text recognition.
- **Segmentation:** Separates the answer sheet into logical sections (questions, answers, diagrams) for accurate processing.

These steps ensure that the image quality is optimized for precise OCR extraction and downstream evaluation processes.

2. OCR & Text Extraction

After pre-processing, the images are passed through **Optical Character Recognition (OCR)**. This stage:

- Converts handwritten or printed text into machine-readable digital text.
- Identifies lines, paragraphs, and answer regions.
- Supports extraction for both **subjective questions** (essays, explanations) and **objective questions** (MCQs, short answers).

This is a crucial foundation that enables the use of NLP and machine learning for automated evaluation.

3. NLP & ML-Based Evaluation

Once the text is extracted, the system uses Natural Language Processing (NLP) and Machine Learning (ML) to analyze and evaluate the content:

- **Text Preprocessing:** Tokenization, stop-word removal, stemming/lemmatization.
- **Grammar & Structure Analysis:** Checks sentence flow, grammar accuracy, clarity, and coherence.
- **Semantic Similarity:** Compares the student's answer with ideal reference answers using advanced language models.

- **Objective Question Evaluation:** Image-based classification or ML models (KNN, SVM, etc.) determine correct answers for MCQs or short responses.

This ensures a consistent, bias-free, and concept-oriented evaluation of answers.

4. Result Generation & Feedback

After processing, the system produces:

- **Scores** for every question
- **Detailed performance analytics** such as accuracy, strengths, and areas of improvement
- **Visual performance reports** accessible through a user-friendly interface
- **Secure data storage in MongoDB** for future reference and audits

This module enhances speed, accuracy, and transparency in evaluation while reducing educators' workload.

MODEL TRAINING AND PREDICTION STEPS

1. Training Data Preparation

Large datasets consisting of scanned answers, reference solutions, exam rubrics, and labeled scores are collected.
The data is cleaned and structured:

- Removal of unwanted noise
- Conversion of answers into numeric features (TF-IDF, embeddings, BERT vectors)
- Associating each answer with its correct marks

This creates a strong foundation for reliable model learning.

2. Feature Extraction

The system identifies critical features from each answer, such as:

- Grammar quality
- Vocabulary usage
- Concept clarity and topic relevance

- Semantic similarity with ideal solutions
- Logical flow and length
- For MCQs: detected option, bounding boxes, pattern matching

These features represent the student's responses in a measurable form.

3. Model Selection

Different types of questions require different models:

- **Classification models:** SVM, Logistic Regression, Random Forest (for objective questions)
- **Similarity-based models:** KNN, cosine similarity (for short answers)
- **Deep learning models:** BiLSTM, Transformers, BERT (for long subjective answers)

The system selects the model that provides the highest accuracy for each question type.

4. Model Training

During training:

- The model learns patterns from thousands of graded responses
- It adjusts weights and parameters to minimize scoring errors
- Validation tests ensure the model generalizes well to new answers

Training continues until optimal accuracy and stability are achieved.

5. Model Evaluation

Before deployment, the model is tested on unseen datasets for:

- Accuracy
- Scoring consistency
- Ability to handle different writing styles
- Detection of key concepts
- Precision in identifying correct/incorrect responses

Fine-tuning is done if performance is not satisfactory.

6. Prediction (Scoring New Answers)

When a new student answer sheet is uploaded:

1. OCR extracts the text
2. Text preprocessing and feature extraction occur
3. Features are passed into the trained model
4. The model predicts:
    - The score
    - Mistakes
    - Missing content
    - Grammar/relevance issues
    - MCQ correctness

This entire process happens instantly and automatically.

7. Final Output Generation

The system presents:

- Final marks
- Question-wise breakdown
- Strengths and weaknesses
- Suggestions for improvement
- Comparison with class performance

All results are organized clearly and stored for future reference.

7. SYSTEM DESIGN

7.1 Architecture / Block Diagram

The architecture of the system is based on a pipeline that starts from data input and ends with useful outputs for decision-makers.

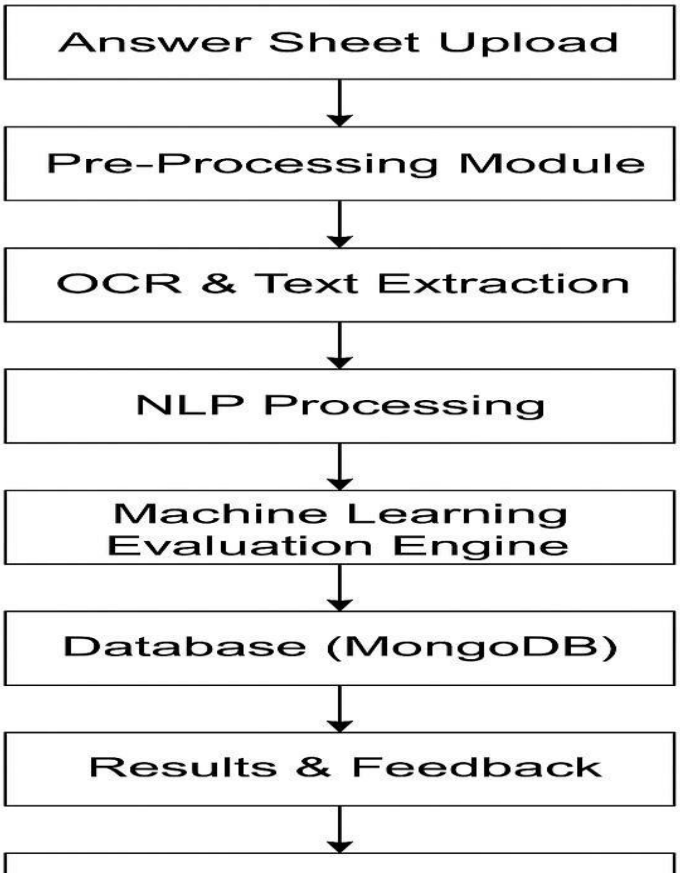

Figure -7.1.1. *System Architecture / Block Diagram for Paper Evaluation using AI*

Description of the Architecture

The architecture of the "Paper Evaluation Using AI" system is designed as a modular pipeline, ensuring efficient processing of scanned answer sheets from input to final score generation. Each component plays a critical role in automating the evaluation process through OCR, NLP, and machine learning technologies.

1. Input Layer – Answer Sheet Upload

- Teachers/students upload scanned handwritten or printed answer sheets.

- System supports JPG, PNG, and PDF formats.
- Files are sent to the preprocessing module for enhancement.

2. Pre-Processing Module

- Converts images to **grayscale** for easier text isolation.
- Removes **noise, blur, ink bleeding**, and unwanted marks.
- Performs **segmentation** to separate answers, questions, and diagrams.
- Ensures clean input for accurate OCR performance.

3. OCR & Text Extraction Layer

- Reads and extracts handwritten or printed text from the scanned sheets.
- Converts visual text into machine-readable digital text.
- Works for both **subjective (long answers)** and **objective (MCQs)**.

4. NLP Processing Unit

- Processes extracted text using:
  - Tokenization
  - Stop-word removal
  - Stemming/Lemmatization
  - Text normalization
- Prepares text for semantic analysis and similarity matching.

5. Machine Learning Evaluation Engine

- Uses trained ML/DL models such as:
  - **KNN** for short answers and MCQs

  - **BERT, LSTM, Transformers** for long subjective answers

- Evaluates:
  - Grammar
  - Concept relevance
  - Sentence coherence
  - Semantic similarity with ideal answers
  - MCQ correctness

6. Database Layer (MongoDB)

- Stores:
  - Uploaded answer sheets
  - Extracted text
  - NLP outputs
  - Model predictions
  - Final scores and analytics

7. Results & Feedback Generator

- Creates question-wise marking reports.
- Highlights mistakes, missing content, and improvement areas.
- Generates performance analytics such as comparative scores and accuracy.
- Ensures transparent and explainable evaluation.

8. User Interface (UI)

- Web interface for teachers, admins, and students.

- Displays results, analytics, evaluated answers, and feedback.
- Supports secure login and role-based access.

7.2 UML Diagrams

1. Use Case Diagram

Purpose

The Use Case Diagram highlights *who* interacts with the system and *what* operations they perform. It reflects external interaction without exposing internal logic.

Primary Actors

1. Teacher

- Uploads scanned answer sheets to the system.
- Initiates evaluation.
- Views and downloads the results.
- Monitors performance analytics for multiple students.

2. Student

- Views individual evaluation results.
- Checks feedback and improvement suggestions.

3. AI Evaluation System

- Automatically performs OCR, NLP, ML scoring.
- Responsible for generating the evaluation report.

4. Database System

- Stores scanned sheets, extracted text, scores, and reports.
- Facilitates retrieval of past evaluations.

Main Use Cases

1. **Upload Answer Sheets**

   The teacher uploads bulk or single answer sheets. The system validates file type and quality.

2. **Pre-process Images**

   The system removes noise, improves clarity, and segments answers to ensure accurate OCR extraction.

3. **Extract Text Using OCR**

   AI converts handwritten/printed content into readable digital text.

4. **NLP-Based Cleaning & Understanding**

   The extracted text is tokenized, cleaned, normalized, and analyzed for semantic meaning.

5. **ML-Based Evaluation**

   Machine learning models score answers by comparing them with a trained knowledge base.

6. **Generate Results & Feedback**

   Detailed performance report with scores, strengths, and improvement advice.

7. **View & Download Reports**

   Teachers and students can access the results in a user-friendly format.

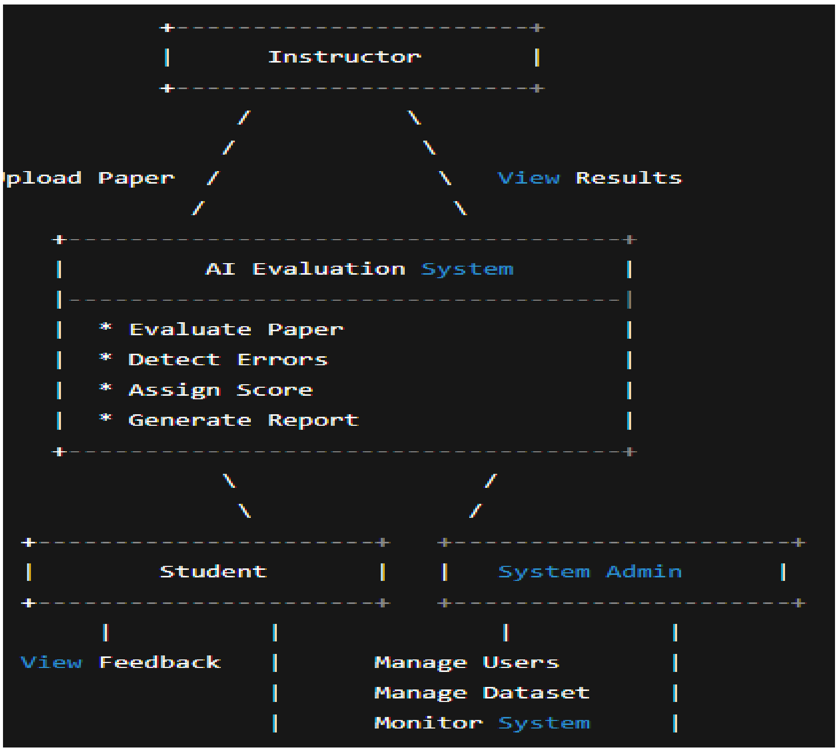

Figure -7.2.1. UML: Use Case Diagram

2. Class Diagram

A class diagram for a paper evaluation system using AI shows the main classes involved in uploading papers, evaluating them, and generating feedback.

1. User

- **Attributes**: userId, name, role
- **Methods:** uploadPaper(), viewResults()

2. Paper

- **Attributes:** paperId, title, content
- **Methods:** getContent()

3. AIModel

- **Attributes**: modelName, version
- **Methods:** analyzeText(), generateScore()

4. Evaluation

- **Attributes:** score, feedback, plagiarismPercent
- **Methods**: generateReport()

Relationships

- **User → Paper:** uploads (Association)
- **Paper → Evaluation:** paper has one evaluation (Composition)
- **Evaluation → AIModel:** uses AI model for scoring (Dependency)

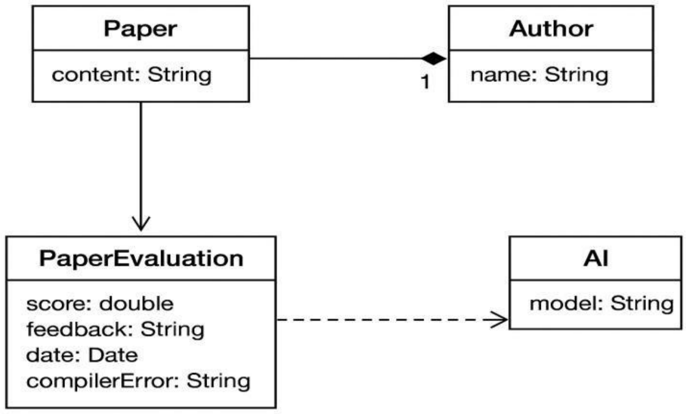

Figure -7.2.2. UML: Class Diagram

3. Activity Diagram

The activity diagram represents the **entire workflow**, showing how the system processes a sheet from input to final output.

Step-by-Step Activity Flow

1. **Start**
   Teacher initiates the process by logging into the interface.
2. **Upload Answer Sheet**
   Sheets are provided in image format.
3. **Image Pre-processing**
   System improves image clarity using noise removal and segmentation.
4. **OCR Extraction**
   Extracts readable text from each answer.
5. **Text Cleaning with NLP**
   Removes unwanted data → normalizes text → finds meaning.
6. **Machine Learning Evaluation**
   Model compares extracted text with ideal responses and calculates scores.
7. **Score & Feedback Generation**
   Evaluation engine creates detailed insights.

8. **Store Results in Database**

   Ensures the results are permanently saved.

9. **Display Results**

   Students and teachers can view, save, and analyze results.

10. **End**

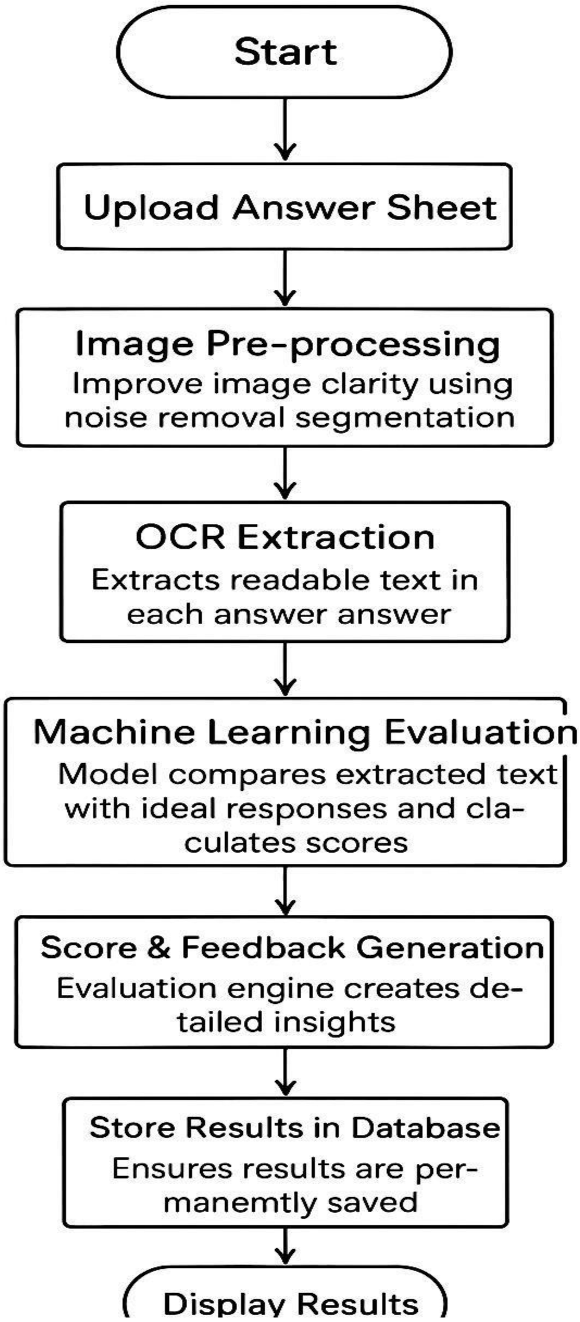

Figure -7.2.2. UML: Activity Diagram

4. Sequence Diagram

A **sequence diagram** shows **how different components interact step-by-step over time** in the paper-evaluation workflow. It highlights the **order of messages**, **data flow**, and **processing sequence** between the user and the AI-based system.

Main Participants

1. **Student / User** – uploads the answer script.
2. **Paper Evaluation System (Front-end)** – receives the file and sends it to backend.
3. **AI Evaluation Engine** – performs text extraction, scoring, and feedback generation.
4. **Database** – stores answers, model scores, rubrics, and results.
5. **Evaluator / Teacher** – optionally reviews or approves final marks.

Flow of Interactions

1. **Student uploads the answer sheet** → sent to the paper evaluation system.
2. System **forwards the file** to the **AI Engine**.
3. AI Engine performs:
    - OCR/text extraction
    - Keyword & concept matching
    - Similarity analysis with model answers
    - Grammar/content quality check
    - Marks generation
4. AI Engine **stores results** (scores + feedback) into the **database**.
5. System **fetches results** and shows them to the **teacher for verification**.
6. Teacher may **approve or modify marks**.
7. Final evaluated score is **returned to the student**.

Purpose

This sequence diagram helps understand **exact data flow**, ensures **clarity in system design**, and verifies that each component works in proper order for accurate and automated evaluation.

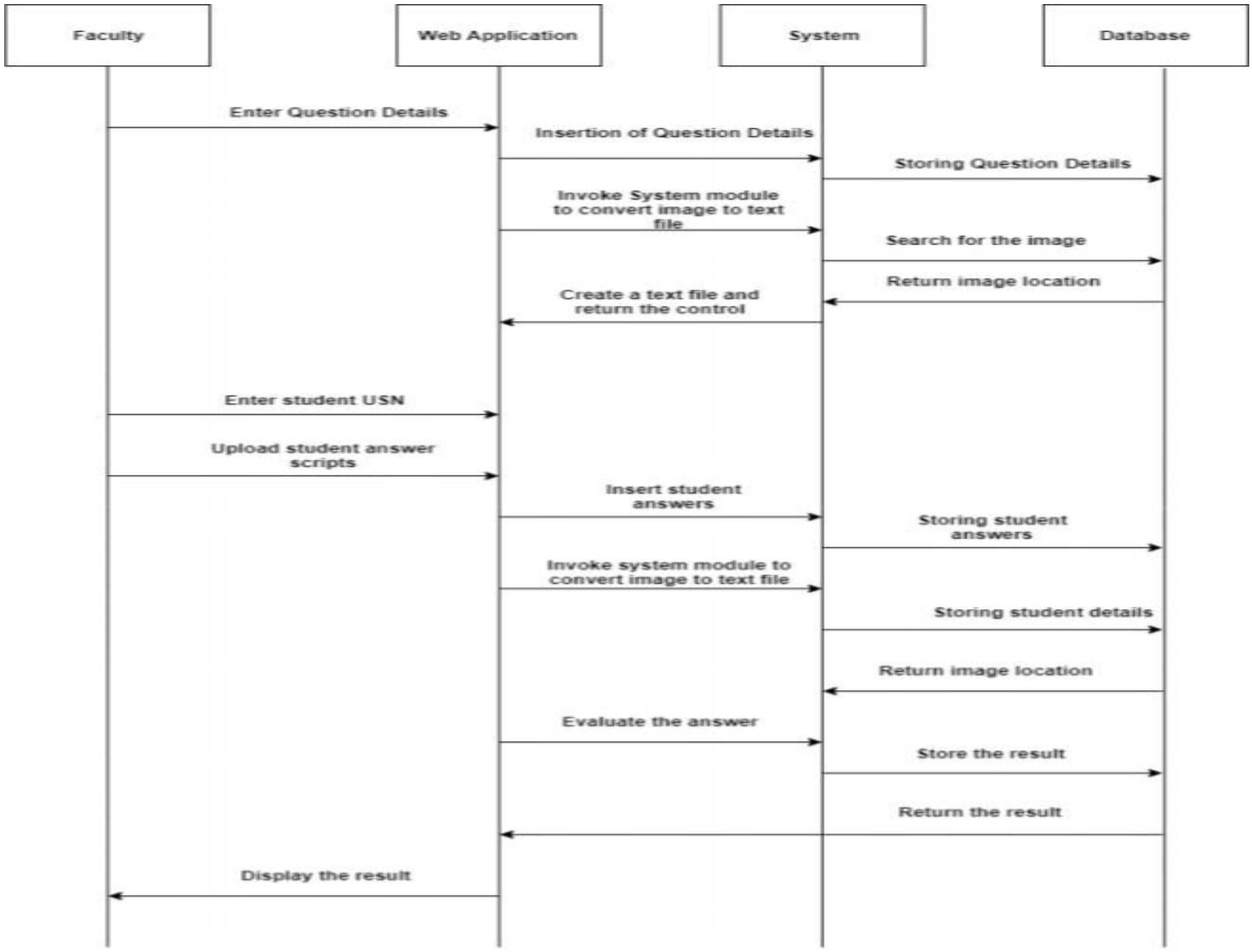

Figure -7.2.4. UML: Sequence Diagram

8. CONCLUSION

The *"Paper Evaluation Using AI"* project aims to modernize and improve the traditional method of evaluating student answer sheets. For many years, manual checking and OMR-based systems have been the primary methods used in schools and colleges. While these methods work for objective questions, they are slow, tiring, and often inconsistent when it comes to descriptive answers. Human evaluators may differ in how they interpret responses, and large volumes of papers can lead to fatigue-based errors.

The proposed AI-based system overcomes these challenges by combining OCR for text extraction, NLP for understanding the content, and machine learning models such as KNN for automated scoring. OCR helps convert handwritten answers into digital text, which allows the system to analyze responses even when students write in different styles. NLP techniques, supported by tools such as NLTK and SpaCy, help the system understand meaning, grammar, structure, and relevance of the answer. Machine learning then compares the student's response with trained data and assigns marks more consistently and objectively.

This integrated approach significantly reduces evaluation time, increases accuracy, and makes the marking process more standardized. Teachers no longer need to manually read every line, and institutions can process thousands of papers more efficiently. Storing evaluated responses in databases like MongoDB further helps with record management, re-evaluation, and result generation.

In conclusion, this project proves that AI can greatly improve the speed, fairness, and reliability of paper evaluation systems. As technology continues to grow, future improvements—such as enhanced handwriting recognition, deeper semantic understanding, and adaptive scoring models—can make AI-based evaluation even more intelligent and trustworthy. This system not only supports teachers but also contributes to building a more transparent and efficient educational assessment process.

9. REFERENCES

Datasets

- Automatic Student Assessment Prize (ASAP) dataset for essay scoring — very commonly used. SpringerLink+2The Science and Information Organization+2
- DREsS: Dataset for Rubric-based Essay Scoring on EFL Writing. arXiv

Machine Learning & AI Libraries

- **Scikit-learn**: Widely used for ML algorithms such as KNN, SVM, etc. Wikipedia
- **CatBoost**: A high-performance gradient boosting library useful for structured data. Wikipedia
- **TensorFlow**: For deep learning / neural network-based scoring (you already plan to use this).

NLP Libraries

- **NLTK (Natural Language Toolkit)**: Useful for tokenization, POS tagging, parsing, and feature extraction. Wikipedia
- **SpaCy**: For efficient, production-grade NLP, and integration with ML frameworks. Wikipedia

Research Papers

- Azahar, M. Z. A. & Ghauth, K. I. "A Hybrid Automated Essay Scoring Using NLP and Random Forest Regression" — demonstrates combining NLP feature extraction and ML regression. Atlantis Press
- Suresh, A. & Jha, M. "Automated Essay Grading using Natural Language Processing and Support Vector Machine" — uses SVM + NLP features for scoring essays. IJCAT
- Vajjala, S. "Automated assessment of non-native learner essays: Investigating the role of linguistic features" — explores which linguistic features (lexical, syntactic, discourse) are predictive. arXiv
- Xiao, C., Ma, W., Song, Q., et al. "Human-AI Collaborative Essay Scoring: A Dual-Process Framework with LLMs" — recent work using large language models and human-AI collaboration. arXiv
- *A survey on deep learning-based automated essay scoring an*