The LeftOvers

Automated Script Checker

Category : Education

AI HACKATHON
EMBRACING THE FUTURE

Project Overview

Manual Script Checking is a **Time Consuming** and **Error Prone** process

We propose **AUTOMATED SCRIPT CHECKER**

Our solution **Automates** the process of **Handwritten Script Checking** through the use of **AI** technology

Technical Overview

Problem Statement

In Bangladesh’s resource-constrained classrooms, traditional manual grading of Bangla-script exams:

- Is highly **time-consuming**, forcing teachers to spend hours on script checking
- Diverts educators from teaching, research, and professional growth
- Introduces **bias** (handwriting neatness, personal preferences) and errors under heavy workloads
- Fails to deliver detailed, **individualized feedback** in a timely manner
- Struggles with Bangla script complexity and high student-to-teacher ratios, further delaying results
- Incur **additional costs** when institutions hire extra graders and risk miscalculations in score tabulation

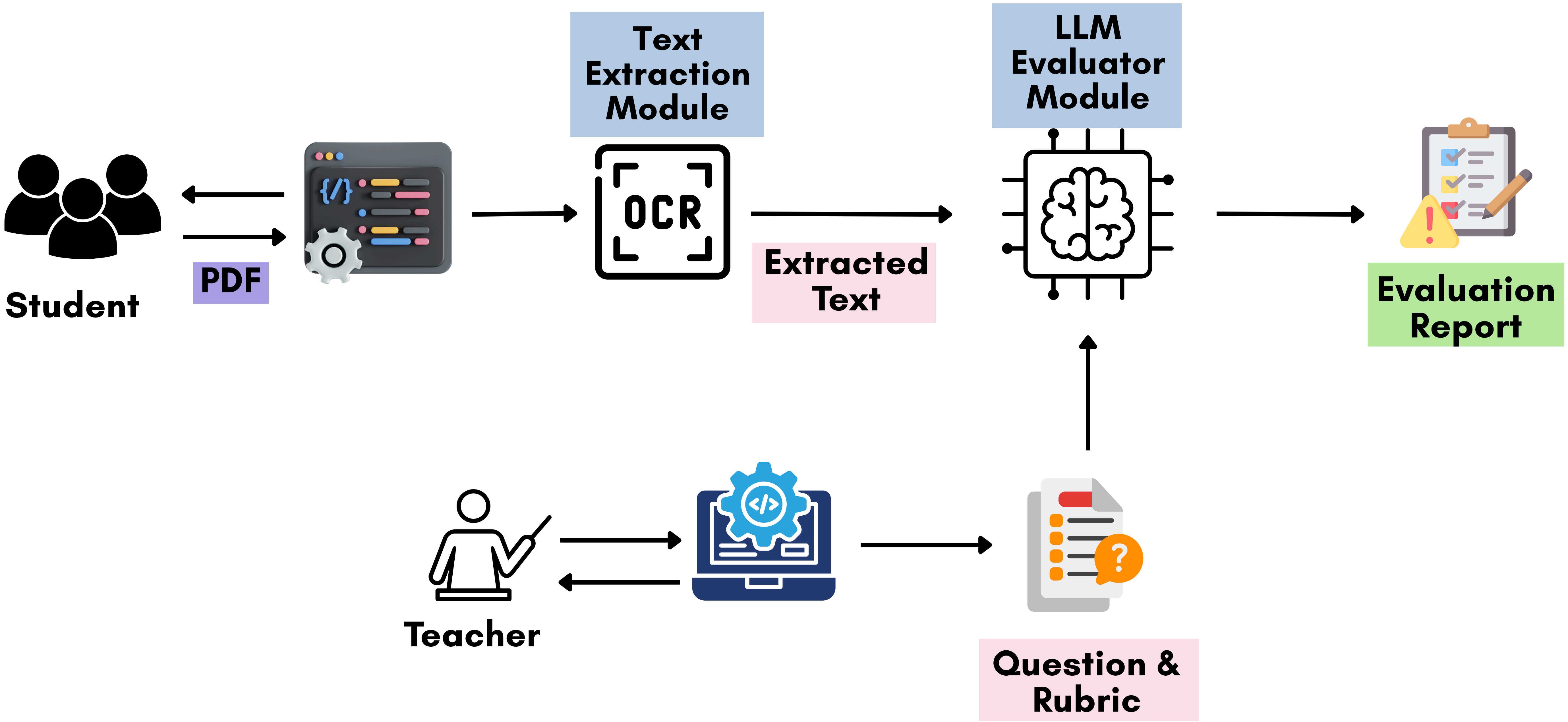
An AI-driven grading system can eliminate these inefficiencies—ensuring fast, fair, accurate, and cost-effective evaluation.

Solution Approach

Answer Evaluation Pipeline

- **Upload:** Students submit handwritten answer scripts as PDF files.
- **Text Extraction:** OCR technology converts the PDF into images and extracts the text.
- **Evaluation:** The extracted text, along with the question and rubric, is sent to the LLM for analysis.
- **Scoring:** The LLM assesses the answer based on the rubric, assigning marks and providing detailed explanations for each criterion.
- **Result Display:** The final scores and feedback are presented clearly on the frontend interface.

Overall Pipeline



Student

PDF

Text Extraction Module

Extracted Text

LLM Evaluator Module

Evaluation Report

Teacher

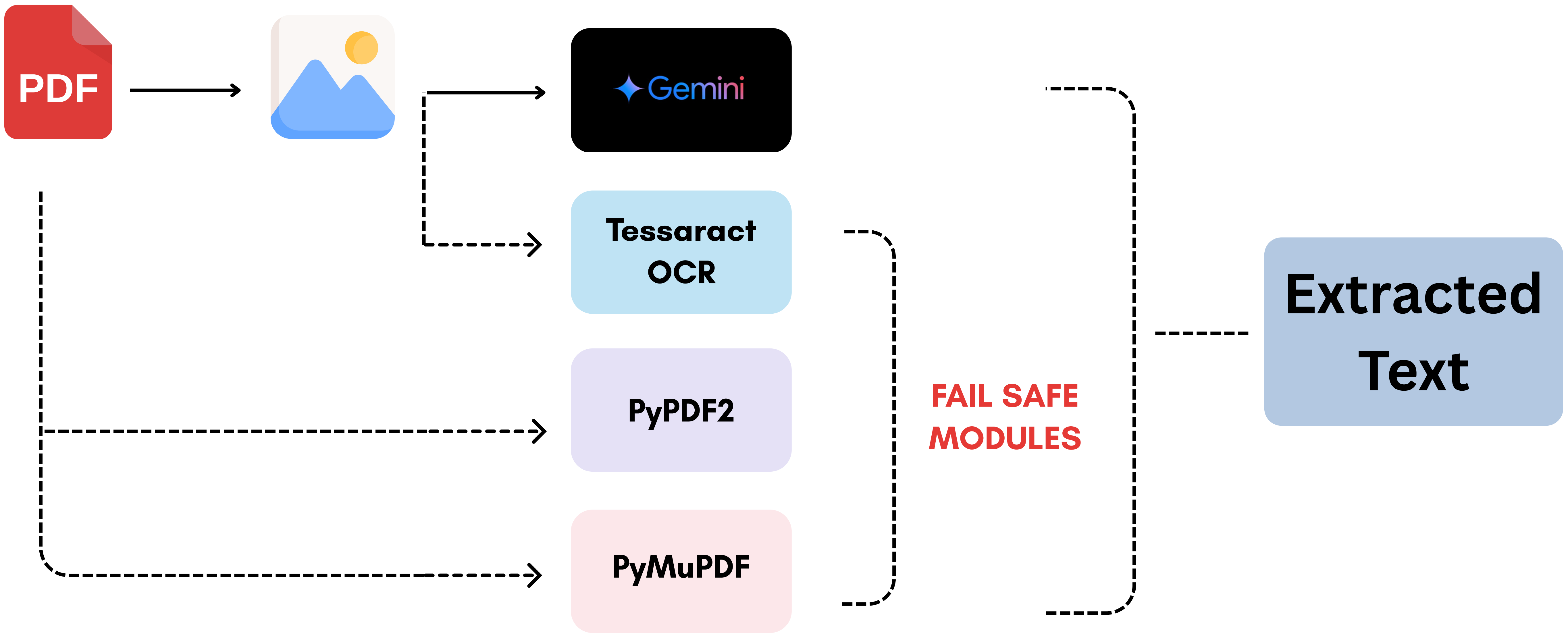
Question & Rubric

Solution Details

OCR Module

- **PDF Ingestion**: Students upload their exam scripts as PDF submissions.
- **Primary OCR Engine:** We run each PDF through **Gemini 2.5 Flash** to extract Bangla (and English) text from scanned images.
- **Fallback OCR Sequence:** If Gemini fails or produces low-confidence output, we automatically invoke the OpenAI OCR endpoint; if that still falls short, we cascade to traditional tools—Tesseract, then PyPDF2, then PyMuPDF—to maximize text-capture accuracy.
- **Text Storage:** All extracted text (with metadata on which engine succeeded) is saved in our database for auditing and retrieval.
- **Interactive Self-Evaluation UI**: From the web interface, students can request an instant “self-grade”—the stored text is sent to our LLM grading module, which reviews answers against rubrics and returns marks and feedback in real time.

OCR Module Overview

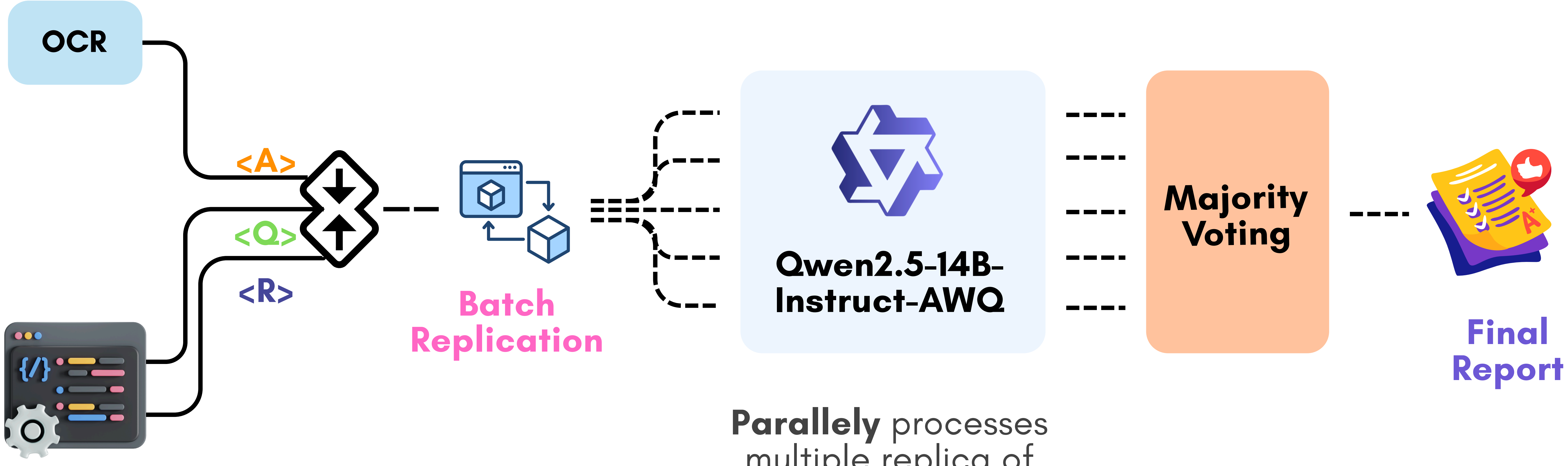


AI Solution Details

LLM Module

- **Batch Creation:** A dictionary containing the question, answer, and rubric with allocated marks is duplicated to form a batch.
- **Parallel Processing:** The batch is sent to the LLM for independent, parallel evaluation of each element.
- **Evaluation Output:** For each batch element, the LLM generates results including rubric text, assigned marks, total rubric marks, and detailed explanations for the scoring.
- **Majority Voting:** The batch outputs undergo a majority voting process to select the most consistent evaluation, ensuring accurate marking and explanations.
- **Result Delivery:** The final evaluation is sent to the frontend for display.

LLM Evaluation Module



Parallely processes multiple replica of **<Q,R,A> Tuple**

AI Model Details

Model Architecture:

Base Architectire:

Transformer-based, following the LLaMA design principles.

Parameter Count:

14 billion parameters.

Context Length:

Supports up to 128K tokens.

Quantization:

Utilizes AWQ to reduce model size and improve inference speed

AI Model Details

Libraries:

Python's internal libraries:

re, typing and collections

External Libraries:

Pytorch, vllm

Future Improvements

Model Accuracy:

Train on larger, diverse datasets and fine tuning for better precision

Efficiency:

Optimize code and use parallel processing(more powerful GPU) for faster evaluations.

Fairness:

Regularly audit to reduce biases and ensure equitable evaluations. up to 128K tokens.

Multilingual OCR:

Improve OCR to accurately extract text from images in various languages for broader usability.

Integration with Academic Workflows

Link the system with platforms like Moodle or other learning management systems (LMS) for automated grading and feedback, enhancing educational efficiency.