IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141170

AutoGrad

Miss. Raheen Rafique Bagwan[1], Miss. Akansha Anil Sasane[2],

Miss. Riya Chandrakant Chawate[3], Miss. Rutuja Atul Kavitake[4]

Department of Computer Engineering,

Al- ameen Educational & Medical Foundation's College of Engineering, Koregaon Bhima, Pune[1-4]

Abstract: In the era of rapidly expanding student populations and increasing academic work- loads, traditional methods of evaluating handwritten answer sheets have become inef- ficient, inconsistent, and resource-intensive. AutoGrad addresses these challenges by leveraging cutting-edge Generative AI to automate the assessment process with high accuracy and scalability. The system integrates the Gemini model for Optical Character Recognition (OCR), effectively digitizing diverse handwriting styles, and the LLaMA-7B language model for semantic answer evaluation.AutoGrad introduces a novel Mixture-of-Experts (MoE) architecture to significantly reduce character recognition errors and uses adaptive thresholding to fine-tune evaluation rigor based on question types. The solution further integrates a rule-based and AI-driven hybrid evaluation engine, ensuring both factual correctness and semantic coherence in student answers. With a Flask-based user interface, vector similarity matching, and a real-time feedback generation system, AutoGrad offers an end-to-end, scalable solution for academic institutions.Empirical results from real-world deployments show a reduction in grading time and a correlation with faculty evaluations. AutoGrad not only automates evaluation but enhances it—providing detailed feedback, promoting personalized learning, and supporting academic integrity at scale.

Keywords: Automated Grading, Generative AI(or LLM), Optical Character Recognition(OCR), Semantic Evaluation, Scalable Assessment, Mixture-of-Experts(MoE), Distributed Processing

I. INTRODUCTION

Assessment is a fundamental pillar of the educational system, directly impacting the learning process, student performance analysis, and academic progression. However, the traditional approach of manually evaluating handwritten answer sheets is labor-intensive, time-consuming, and prone to human bias and inconsistency. As student enrollment increases and academic institutions strive to maintain evaluation quality, the limitations of manual assessment have become more pronounced.

To address these challenges, AutoGrad proposes an AI-driven automated grading system designed to evaluate handwritten answer sheets with high accuracy, speed, and semantic understanding. Unlike conventional grading software that is limited to struc- tured formats such as multiple-choice questions (MCQs) or typed responses, AutoGrad is tailored for unstructured, handwritten responses, which are more common in theoretical and technical examinations

The system integrates state-of-the-art Generative AI models—namely the Gemini model for Optical Character Recognition (OCR) and the LLaMA language model for semantic answer evaluation. The Gemini model, enhanced through a Mixture-of-Experts (MoE) architecture, is optimized for various regional handwriting patterns, significantly improving character recognition accuracy. Meanwhile, the LLaMA model ensures a deeper contextual understanding of student responses, beyond mere keyword matching.

AutoGrad not only reduces the workload of educators but also ensures consistent and transparent evaluation, offering actionable feedback for students. The system's modular and scalable design enables easy deployment across academic institutions, making it a practical solution for the evolving needs of modern education.

II. RELATED WORK

The field of Automated Grading has historically been limited to structured tasks like multiple-choice and programming, struggling with the complexities of handwritten, subjective responses. Traditional methods, including conventional OCR and Rule-Based Grading, fail to reliably handle diverse handwriting or achieve true semantic understanding. Existing commercial solutions often lack the necessary scalability and transparency for large institutions. The "AutoGrad" project tackles this by integrating cutting-edge Generative AI technologies. It uses the Gemini model with a Mixture-of-Experts (MoE) architecture for highly accurate OCR and a fine-tuned LLaMA-7B model for deep contextual evaluation. This

DOI: 10.17148/IJARCCE.2025.141170

approach moves beyond simple keyword checks, enabling the system to assess logical coherence and infer meaning, thus providing a scalable and transparent solution for modern education.

III. METHODOLOGY

The AutoGrad system aims to revolutionize the process of academic assessment. It will combine the interpretative depth of large language models, the pattern recognition strength of advanced OCR, and the operational flexibility of modular system design to create an end-to-end solution for modern educational evaluation.

The proposed system follows a structured process consisting of the following steps:
1. Input and Submission
2. Central Processing
3. Preprocessing
4. Output and Storage
5. PDF Upload and Preprocessing
6. OCR Layer – Gemini Model with Mixture-of-Experts

IV. ALGORITHM USED

A. Gemini Model with Mixture-of-Experts (MoE): MoE enhances the Gemini model to perform Optical Character Recognition (OCR). It accurately digitizes diverse handwriting styles by dynamically routing input to specialized sub-networks based on writing features.

B. LLaMA-7B Language Model: This model is the engine for Semantic Answer Evaluation. It performs deep contextual analysis to assess a student's response for coherence, completeness, correctness, and logical progression.

C. Vector Similarity Matching (ChromaDB): Student answers are converted into numerical vectors (embeddings) and compared to model answers in a vector database. It uses Cosine Similarity to assign a score based on the conceptual alignment of the student's answer.

D. YOLOv8 Object Detection Model: Used in the Preprocessing Module to perform Page Segmentation. It identifies and isolates individual answer regions for specific questions before sending them to the OCR.

E. Rule-Based Evaluation: This component is used for Numeric / Objective answers. It applies predefined logic to check for exact mathematical correctness or specific expected outputs, ensuring alignment with simple marking guidelines.

V. DATABASE DESIGN

The system uses a MongoDB database for securely storing all backend data. This NoSQL architecture allows for flexible, scalable, and efficient data handling. Sensitive data such as file uploads and evaluations are encrypted using TLS/SSL protocols to ensure privacy and security. The database maintains records of uploaded answer sheets, question metadata, model answers, rubrics, and the resulting scores and feedback.

VI. SYSTEM ARCHITECTURE

System Architecture Design and Workflow Overview
AutoGrad's end-to-end architecture is designed using a modular, scalable microser- vices paradigm. Each functional unit runs as an independent service containerized using Docker and orchestrated using Kubernetes, allowing seamless deployment, scaling, and fault isolation. The entire pipeline can be deployed on institutional servers or integrated into cloud platforms such as AWS, Azure, or GCP based on organizational needs.

The workflow is as follows:
1. User Upload Interface: Faculty logs into a secure dashboard to upload answer sheets and question metadata.
2. Preprocessing Module: Uploaded PDFs are parsed, enhanced, segmented, and tagged.
3. Distributed OCR Engine: Gemini OCR instances run in parallel to digitize segmented answers.
4. Evaluation Pipeline: Each answer is evaluated in a distributed queue using NLP and LLM services.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.141170

5. Feedback and Scoring: Results are saved, rendered on dashboards, and made available for export and download.
AutoGrad uses asynchronous processing with task queues (e.g., Celery with Redis or RabbitMQ) to handle large batches efficiently without blocking the user interface. This allows simultaneous grading of thousands of answer sheets without performance degra- dation.

Scalability and Parallelism
Scalability is a core design goal of AutoGrad. The system includes:
-Horizontal Scalability: AutoGrad services are stateless, allowing horizontal scaling based on load. OCR, NLP, and LLM components can run across multiple GPUs or TPUs in parallel.
-Batch Processing Engine: The platform supports scheduled batch evaluations, allowing nightly runs for large institutions or real-time on-demand grading for small classes
-Load-Balanced Evaluation Nodes: Evaluation servers dynamically assign workloads to available resources based on GPU availability, memory footprint, and active thread count.

By deploying evaluation, the system can handle spikes in demand such as exam seasons or admission tests.

Security and Data Privacy
In academic settings, security and privacy are critical. AutoGrad incorporates com- prehensive security protocols, including:
-End-to-End Encryption: All file uploads, evaluations, and report downloads are en- crypted using TLS/SSL protocols.
-Role-Based Access Control (RBAC): Separate permission levels are defined for students, faculty, admins, and auditors. Only authorized users can access specific parts of the data or evaluation reports.
-Audit Logging: Every action taken on the system—from sheet upload to final grade export—is logged and timestamped for compliance and audit purposes.
-Data Retention Policies: Institutions can define how long student data is stored. Auto- Grad supports auto- deletion policies and full compliance with data protection laws like GDPR and India's DPDP Act.

Customization and Extensibility
Recognizing that different institutions, departments, and instructors may have unique grading styles, AutoGrad is built to be fully customizable:
-Custom Rubric Editor: Faculty can define detailed rubrics using a visual interface—specifying keyword weightages, logical components, required reasoning steps, or custom evaluation tags.
-Pluggable LLM Backend: While LLaMA-7B is the default LLM, institutions can swap in other open models like Mistral, Claude, or even proprietary LLMs through a simple adapter module.
-Plugin Support: AutoGrad includes support for external plugins such as plagiarism detectors, diagram matchers (via computer vision), or automated code checkers for pro- gramming exams.
This flexibility ensures that AutoGrad evolves alongside pedagogical and regulatory changes.

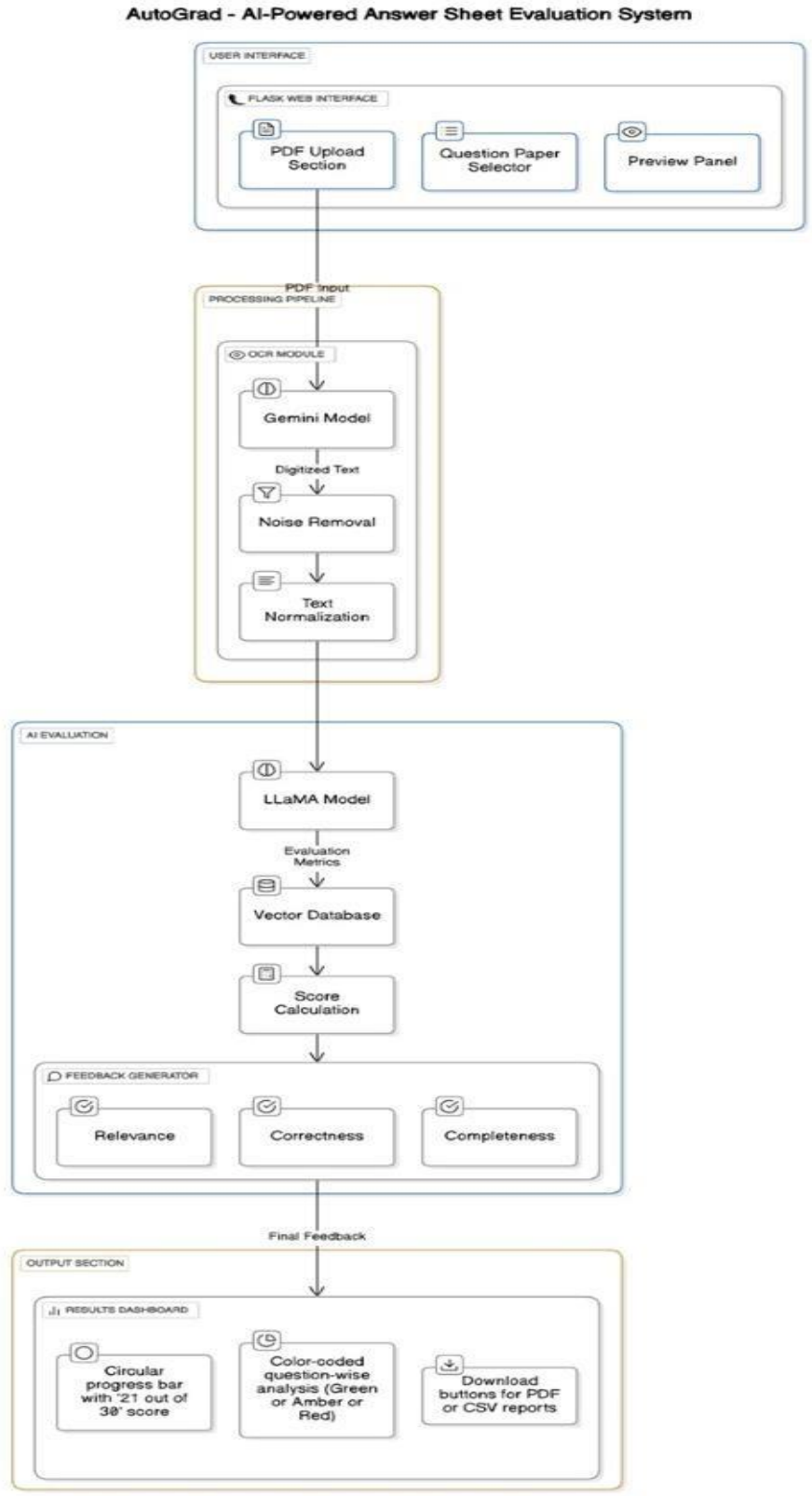
Diagram and Code-Based Answer Evaluation
Many technical subjects involve diagrams, flowcharts, and pseudocode. AutoGrad includes special handling for these cases:
-Diagram Recognition**: Integrated computer vision modules detect flowcharts, state diagrams, circuit schematics, and labeled figures. Using pretrained convolutional models (ResNet, EfficientNet), it compares student diagrams with annotated model diagrams for accuracy, completeness, and labeling correctness.
-Code Block Grading: For programming-related answers, code snippets are extracted and passed through a custom-built evaluator that checks:
- Syntax correctness
- Logic structure
- Output prediction
- Time-space complexity estimation
-Hybrid Content Evaluation: Answers with mixed content (e.g., theory + code or dia- gram + explanation) are processed using composite pipelines that assign weighted scores to each segment.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141170



Architecture

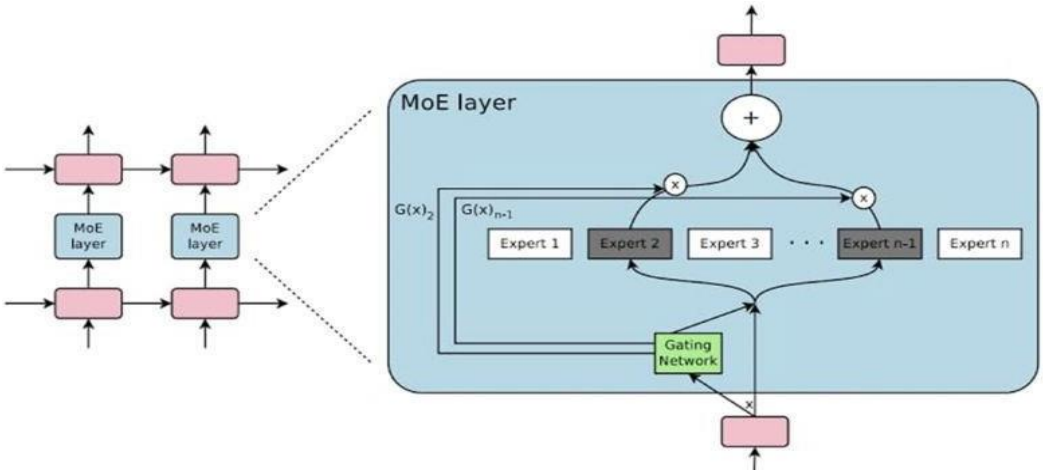
Architecture The architecture of AutoGrad is built on a scalable, modular pipeline that leverages Mixture-of-Experts (MoE) layers to optimize performance in OCR and semantic evalua- tion. The diagram below illustrates the internal working of the MoE layer used in both the Gemini OCR module and the LLM-based grading engine.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141170



AutoGrad is an advanced AI-based grading system engineered to process, interpret, evaluate, and report on handwritten academic responses with high precision and ef- ficiency. Its architecture is modular, scalable, and optimized for performance using cutting-edge technologies such as Mixture-of-Experts (MoE) models, large language mod- els (LLMs), and high-throughput distributed processing. The entire pipeline supports real-time and batch processing and has been designed with flexibility, extensibility, and institutional integration in mind. The architecture of AutoGrad consists of the following core components:

• Input Layer
• Image Preprocessing
• MoE-Based OCR System (Gemini)
• Semantic Answer Evaluation (LLM + Vector Matching) Feedback and Reporting
• Scalability and Efficiency

Each of these components has been optimized for specialized tasks and communicates with the others via well-defined APIs or message queues. Below is a detailed look into each component and its internal mechanisms.

Input Layer

The input layer serves as the entry point for document ingestion and processing. It is built using Flask, a lightweight web framework for Python, which provides an intuitive and secure interface for faculty members and academic administrators. Users interact with a web portal to upload scanned PDFs of handwritten answer sheets and corresponding question papers. Each upload can be tagged with metadata such as exam name, subject code, assessment type (quiz, midterm, final), and time window.

The system supports bulk uploads and drag-and-drop functionality. Backend valida- tions ensure the PDFs are not corrupted and are scanned with adequate resolution for 22 OCR processing. Uploaded documents are stored in a versioned object store, allowing rollback and traceability. The input layer also handles authentication and access control, using JWT tokens or OAuth integrations with institutional login systems such as LDAP, Shibboleth, or Google Workspace.

Image Preprocessing

After receiving scanned answer sheets, AutoGrad performs a series of preprocessing steps to clean, normalize, and prepare the images for OCR.

This module includes noise reduction using Gaussian and bilateral filters to remove scanner dust and image artifacts adaptive thresholding and binarization to convert the image into a clean black-and-white format, improving contrast for OCR alignment correc- tion and deskewing using Hough Line Transform and affine transformations page border removal and margin trimming to focus only on relevant content areas.

The preprocessed images are then passed through a YOLOv8-based segmentation engine. YOLOv8 is a real-time object detection model trained specifically on academic documents. It detects question-wise answer regions, tables, diagrams, and handwritten math expressions. Each identified region is cropped and saved as an individual image patch. Segmentation is followed by semantic tagging, where each region is labeled with the corresponding question number, page index, and positional coordinates. This facilitates question-wise processing and improves the evaluation accuracy. In the backend, all image operations are performed using OpenCV and TorchVision pipelines, running on CUDA for acceleration. This ensures preprocessing does not be- come a bottleneck even during peak loads.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141170

MoE-Based OCR System (Gemini)

AutoGrad employs a state-of-the-art Gemini OCR engine, enhanced by a Mixture- of-Experts (MoE) architecture, to digitize the handwritten content from answer regions. Traditional OCR systems often struggle with the diversity of handwriting styles, regional scripts, and mathematical notation. The MoE-based approach addresses these challenges with specialization and adaptability.

The OCR system comprises the following subcomponents: Gating Network: This neural controller analyzes incoming handwriting samples and assigns routing scores to each expert model based on handwriting style features such as slant, stroke density, and curvature. It is trained using reinforcement learning techniques for dynamic optimization.

Expert Models: Each expert is a transformer-based model fine-tuned on a specific handwriting style.

Examples include cursive writing expert, block lettering expert, math- focused expert, Devanagari-influenced script expert, and multilingual expert for regional scripts.

Dynamic Routing: For each image patch, the gating network selects the top-k expert models. These experts process the input in parallel. Their outputs are then combined using a learned aggregation function that assigns weighted scores to each expert's pre- diction.

Final Output Generation: The aggregated outputs are decoded into UTF-8 text us- ing a beam search decoder with character-level attention, producing a highly accurate transcription of the handwritten content.

This architecture enables the system to deal with variations in writing style across different regions, age groups, and educational levels. It also drastically reduces computa- tional overhead since only a subset of experts is active per input.

Semantic Answer Evaluation (LLM + Vector Matching)

Once the handwritten content is digitized, the next step is to evaluate the meaning, correctness, and completeness of each answer. This is performed using a hybrid semantic engine that combines natural language understanding, keyword analysis, vector similarity matching, and deep contextual reasoning.

The semantic evaluation pipeline consists of the following stages: Keyword Extraction and Vectorization: Using domain-specific models fine-tuned with technical and academic corpora, the system extracts core concepts from the student's answer. Named entity recognition, noun phrase chunking, and topic modeling are performed using SpaCy and custom transformers. The extracted concepts are vectorized using embeddings generated from a BERT-based encoder trained on textbooks and lecture notes.

Model Answer Repository: Model answers and rubrics are stored in ChromaDB, a high-performance vector database. Each model answer is encoded into vector space. Additional metadata includes associated marks, rubrics, keywords, and answer structures.

Cosine Similarity Matching: The vectorized student answers are compared to model answers using cosine similarity. A similarity score is generated that reflects conceptual alignment between the student's response and the model expectation.

LLM-Based Contextual Grading: The same answer is passed to a fine-tuned LLaMA- 7B model. The LLM is guided using question-specific prompts that include rubrics, expected logic flow, reasoning checkpoints, and common misconceptions. The model evaluates each answer for structural coherence, logical flow, relevance, and completeness.

Score Aggregation: The cosine similarity score and the LLM's grading output are combined using a rule- based or learnable aggregation model. Final marks are calculated by mapping the combined score onto the rubric-defined mark distribution.

This hybrid system ensures that grading is both objective and contextually nuanced. It handles factual, descriptive, and analytical questions equally well, while being robust to variations in wording and answer structure.

Feedback and Reporting AutoGrad places significant emphasis on feedback, transparency, and reporting. The goal is not only to score answers but to foster learning and institutional accountability. Student Feedback Interface: - -each answer is color-coded based on accuracy (green for correct, amber for partial, red for incorrect) - tooltips and callouts explain the reasoning behind deductions (e.g., missing definitions, incorrect diagrams, unsupported logic) - students receive suggestions such as "mention algorithm name" or "state assumptions clearly" Faculty Dashboard: - question-level analytics such as average scores, distribution curves, and standard deviations – heatmaps showing which concepts were poorly understood across the class - AI-flagged irregularities including possible copy-paste patterns, duplicated logic, or unnatural answer flow Admin Reporting: -PDF and CSV reports for every batch, customizable by date, subject, or class - NBA- and NAAC-compliant formats for academic audits - support for anonymized exports for peer-review and moderation. All feedback is version-controlled and stored in an institutional record system with access logs for audit purposes.

Scalability and Efficiency

AutoGrad is engineered for high throughput and responsiveness, suitable for institu- tions evaluating thousands of students simultaneously. Key strategies include:

MoE Parallelization: Since only a few expert paths are activated per input, MoE architectures reduce both latency and energy consumption. GPU threads are allocated only where needed. Distributed Processing Queue: AutoGrad uses Celery

DOI: 10.17148/IJARCCE.2025.141170

and Redis to distribute tasks across multiple worker nodes. Each OCR, evaluation, or report generation task runs independently and can be retried in case of failure.
Asynchronous Processing: All backend services use non-blocking I/O and asynchronous HTTP requests (via FastAPI workers) to ensure a responsive frontend. Containerized Deployment: Each service is deployed in a Docker container and man- aged using Kubernetes Services auto-scale based on CPU and GPU utilization metrics, ensuring uninterrupted service even during peak exam seasons.
High Availability: Services are deployed across availability zones with load balancing and failover strategies. A centralized logging and monitoring system based on Prometheus and Grafana provides real-time visibility into system health.

VII. RESULT AND ANALYSIS

The results are presented based on three core functional areas: Image Processing (OCR), Grading Accuracy (LLM), and Efficiency.
Accuracy Results:
Image & Text Quality: 93%
Grading Accuracy: 80%
Operational Efficiency: 15%

VIII. FUTURE SCOPE

The AutoGrad system will be upgraded with dedicated deep learning models (e.g., CNNs for handwriting, few-shot LLMs for evaluation), real-time adaptive grading APIs driven by human feedback, and cross-platform integration capabilities for seamless deployment within existing academic ecosystems (LMS/SIS).

IX.CONCLUSION

The AutoGrad system represents a significant step forward in the field of automated academic evaluation. By integrating advanced Optical Character Recognition (OCR), deep learning-based handwriting analysis, and semantic grading via large language mod- els (LLMs), the system effectively addresses the inefficiencies and inconsistencies associ- ated with traditional manual assessment methods.The use of Mixture-of-Experts (MoE) architecture allows AutoGrad to dynamically select specialized models for different types of handwriting and answer structures, enhancing accuracy and scalability.

Moreover, the semantic grading engine, powered by vector similarity and fine-tuned LLMs, ensures fair and context-aware evaluation, even when student answers deviate from the standard pattern but are conceptually correct.Through the implementation of MongoDB, the sys- tem achieves a flexible, scalable, and efficient backend that supports rapid data retrieval, analysis, and report generation.

AutoGrad not only saves valuable faculty time but also delivers faster and more transparent feedback to students, promoting a better learning experience.

REFERENCES

[1]. A. Khan and M. Matskin,“Cost modelling and optimization for cloud: agraph- based approach,”JournalofCloudComputing,vol.13,no.1,2024.[Online]. - https://doi.org/10.1186/s13677-024-00709-6
[2]. C. Anantaram, G. Nagaraja, and K. Narayanan, “Verification of accuracy in rule-based systems for automated grading,”Data Knowledge Engineering,vol.10,pp. 115–138, 2019.
[3]. Google Deep Mind,“Gemini1.5ProTechnicalReport,”2023.[Online]. – https://deepmind.google/gemini
[4]. H. Aldriye and A. Alkhalaf, “Automated grading systems for programming assignments: A literature review, ”International Journal of Advanced Computer Science and Applications,vol.10,no.3,pp.1–8,2019.
[5]. M. Messer, N. C. C. Brown, and K. M. Smith,“A systematic review of automated grading and feedback tools in programming education,”ACM Transaction son Computing Education,vol.24,no.2,pp.1–43,2024.
[6]. Z. Yangetal.,“CC-OCR:A comprehensive and challenging bench mark for evaluating large multi modal models in literacy tasks,”Proc.ofCVPR,2024.[Online]. - https://arxiv.org/abs/2412.02210

