Handwritten Content Recognition and Text Transformation Using Machine Learning

PREETHI R[1], Dr SUNITHA G.P[2], Mr. SANTHOSH S.G[3]
[1]PG Student, Dept of MCA, JNN College of Engineering, Shivamogga
[2]Associate Professor, Dept of MCA, JNN College of Engineering, Shivamogga,
[3]Associate Professor, Dept of MCA, JNNCE, Shivamogga, and Research Scholar, Dept of ICIS, Srinivas University, Mangalore, India, OCRCIDID:0009-0004-1587-4656

***Abstract*—The manual evaluation of mixed-format response sheets is inefficient and lacks consistency when conducted on a large scale. This study introduces a CPU-efficient system that integrates OpenCV-based optical mark recognition for multiple-choice questions, EasyOCR with SymSpell correction for text extraction, and a rubric-aligned big language model to allocate numeric scores to descriptive responses. The tool, developed in Flask with role-based access, generates annotated overlays and downloadable reports, facilitating transparent audits without the need for GPUs. Pilot tests on diverse scan characteristics demonstrate dependable MCQ detection and substantial concordance with teacher evaluations, resulting in expedited turnaround and enhanced scoring uniformity.**

***Index Terms*—OpenCV, Optical Mark Recognition, OCR, EasyOCR, SymSpell, Flask, Large Language Model, Automated Grading, Educational Technology, Image Processing, Contour-based Detection, Spell Correction.**

I. INTRODUCTION

Mass assessments in universities and training institutions consistently produce thousands of heterogeneous response sheets. Human assessment of this magnitude is sluggish, expensive, and susceptible to variability in results, particularly when evaluators switch between multiple-choice items and open-ended descriptive answers. Traditional digital technologies generally focus on a singular aspect of the workflow, such as optical mark recognition for multiple-choice questions or basic OCR for text extraction, necessitating that teachers reconcile incomplete results, verify edge cases, and manually generate reports. With the advancement of document imaging, optical character recognition, and language modeling, the establishment of an integrated and auditable automation pipeline for comprehensive evaluation has become both practical and sought after.

This project offers an AI-based solution that processes scanned pages or smartphone images of answer sheets, assesses multiple-choice questions via accurate mark identification, and grades descriptive responses using optical character recognition, spelling correction, and rubric-based semantic evaluation. The solution is engineered for conventional CPU hardware to support institutions without access to GPUs and specialist servers. A straightforward web interface developed with Flask enables instructors to upload batches, examine annotated overlays, validate borderline situations, and export results in JSON, CSV, or printable report formats. The platform enhances efficiency and clarity by integrating image processing with language comprehension, hence decreasing turnaround time and increasing transparency and consistency.The identification of marked options in multiple-choice questions is conducted using OpenCV-based image processing. Following denoising and binarization, candidate bubbles are identified by contour analysis, limited by dimensions and proximity to circularity, and organized into rows and columns that depict the question format. A fill-level assessment that contrasts inner disk intensity with an adjacent ring determines the chosen choice, providing dependable performance amidst slight skew and noise. The system utilizes EasyOCR for text extraction on the CPU and subsequently employs SymSpell to rectify prevalent recognition issues caused by scanning artifacts and typography. The revised student response and the instructor's reference answer are assessed using a rubric-aligned large language model prompt that yields a numerical score

within the designated marking range.A graphical dashboard presents both machine-readable data and user-friendly information. Annotated multiple-choice question overlays render each decision readily auditable at a look. Token-level OCR confidence metrics and correction records assist teachers in comprehending and, if necessary, amending results. Role-based access control, adjustable thresholds for low-confidence indicators, and individual submission audit trails facilitate responsible utilization in academic environments. The architecture is modular and capable of full CPU operation, with configurable offline alternatives for the language model when network isolation is necessary.The suggested approach illustrates the integration of accessible computer vision and natural language technologies to provide a functionalgrading assistance for resource-limited institutions. By consolidating capture, recognition, scoring, and reporting into a single pipeline, the system reduces operational work, accelerates feedback cycles for students, and creates a uniform baseline for human graders to concentrate on genuinely unclear or creative responses.

II. RELATED WORK

[1] Smith (2007) — Introduces the Tesseract OCR engine, outlining confidence scores and layout considerations useful for routing low-confidence regions to human review in exam workflows.[2] Bradski & Kaehler (2008) — Core OpenCV primitives (thresholding, contours, morphology) that underpin robust OMR bubble localization and skew-tolerant pre-processing.[3] Gonzalez & Woods (2018) — Establishes image pre-processing fundamentals (denoising, contrast normalization, geometric correction) that stabilize OMR/OCR pipelines.[4] Grinberg (2018) — Practical Flask patterns for secure, auditable web deployment (role-based access, batch jobs, exports) applicable to assessment platforms.[5] Kumar et al. — Classical CV OMR with adaptive thresholding + contour/geometry priors; CPU-only, resilient to mild skew/illumination, with verifiable overlays for audits.[6] Sharma et al. — OCR for descriptive answers; dictionary-assisted post-processing (e.g., SymSpell) markedly boosts readability and downstream matching on printed/clear handwriting.[7] Jurafsky & Martin (2023) — Language-modeling foundations (tokenization, embedding semantics) that support rubric-aware text similarity and calibrated scoring.[8] Patel et al. — Short-answer scoring evolves from TF-IDF to sentence embeddings and tightly constrained LLM prompts; recommends calibration sets and confidence flags.[9] Singh et al. — End-to-end assessment stack (OMR+OCR+scoring) with roles, batch processing, and evidence downloads; stresses transparency and traceability alongside accuracy.[10] JaidedAI (2024)
— EasyOCR provides multilingual, CPU-friendly OCR that reduces GPU dependence and improves practicality for classroom-scale deployments.

III. MATHEMATICAL MODEL

This project provides a complete grading pipeline for mixed-format tests that is easy on the CPU. It takes in sheets that have been scanned or taken with a mobile device, fixes the perspective, separates the MCQ and descriptive areas, finds bubbles against the key, and utilizes OCR with dictionary correction and rubric-aware/semantic scoring to grade written answers within given limits. A simple dashboard shows all the decisions made, along with overlays, confidences, and review flags. This makes it easy to process batches, export data, and do reliable human-in-the-loop checks

A. Notation

Let I be the grayscale exam image. For question q with options $j \in \{1, K\}$ (typically K=4), each option has an inner bubble region B_{qj} and a surrounding ring R_{qj}. Let mean $(\cdot)$ be the average pixel intensity.

B. MCQ bubble score and decision

- $s_{qj} = \text{mean}(R_{qj}) - \text{mean}(B_{qj})$ => Eq(1)

Filled bubbles are darker (lower mean in B), so this difference gets larger when a bubble is filled.

- $\hat{j}_q = \text{argmax}_j\, s_{qj}$, and mark as filled if $s_{q\hat{j}} \geq \tau$; otherwise mark as unfilled/ambiguous.

Pick the strongest bubble; require it to clear a minimum gap from background.

- $\text{Score\_MCQ} = \Sigma_q\, 1\{\hat{j}_q = j_q\}$, where j_q is the answer key => Eq(2)

Count how many picked options match the answer key.

C. Descriptive-answer scoring

- Option A (LLM prompt grader): => Eq(3)

Score_DESC = clip ($\hat{y}$, 0, M), where $\hat{y}$ is the numeric

grade returned by the model and M is max marks. Use the LLM's numeric grade, but clamp it to [0, M].

- Option B (embedding similarity fallback):

=> Eq(4)

- sim(s, r) = (E(s) · E(r)) / (‖E(s)‖ ‖E(r)‖) => Eq(5)

Cosine similarity between student and reference answers.

- Score_DESC = M × max(0, (sim(s, r) − γ) / (1 − γ)) => Eq(6) If similarity is below γ, zero; otherwise scale linearly up to M.

D. Final aggregation

- Score_FINAL = w × Score_MCQ + (1 − w)

× Score_DESC, with weight w∈[0,1] => Eq(7)

Weighted combine: set w higher to emphasize MCQs, lower to emphasize descriptive.

IV. METHODOLOGY

The proposed methodology uses computer vision and language modeling to evaluate mixed format answer sheets in real time on CPU. The system combines OpenCV based optical mark recognition for MCQs, EasyOCR with SymSpell correction for text extraction, and a rubric aligned large language model that returns a numeric score for descriptive responses. Input pages come from scanners or smartphone cameras. Each page is validated for file type and size, optionally perspective corrected with a four point transform, then preprocessed and routed through the MCQ and descriptive branches before aggregation into final marks with audit artifacts and exports. Implementation uses Python, OpenCV, EasyOCR, SymSpell, and Flask. For MCQs, the image is converted to grayscale, contrast enhanced with CLAHE, binarized with Otsu thresholding, and denoised with small morphological filters; deskew is estimated from projected text lines or Hough peaks to stabilize geometry. Near circular contours are extracted and filtered by size and aspect constraints, columns are inferred by one dimensional k means on x coordinates, rows are formed by clustering along y, and a fill score compares inner disk intensity with a surrounding ring for each option; the darkest option above a calibrated threshold is selected and matchedto the answer key, with ties broken by minimum distance to column centroids and ambiguous rows flagged when all scores fall below the threshold.

The number of options per question is configurable and the row tolerance band is tuned so that light marks and mild skew are still detected while stray ink is rejected. For descriptive answers, the page region is enhanced, OCR is performed on CPU, and tokens are corrected with SymSpell using a course specific whitelist so that rare proper nouns are preserved; token confidences are aggregated to produce a page confidence and low confidence pages are routed to review. The corrected student text and the instructor reference are provided to a constrained prompt that encodes the rubric and the maximum marks and returns a numeric score that is clamped to the allowed range; an offline fallback based on sentence embeddings is available for air gapped use. Scores are combined with configurable section weights, and both raw decisions and normalized marks are stored along with per question evidence such as bubble thumbnails, token corrections, and confidence values. The server produces annotated overlays for MCQ evidence, machine readable JSON and CSV logs, and printable reports with timestamps and unique submission identifiers, and exposes endpoints for batch upload and download to support classroom scale workflows. Role based access and basic audit logging are implemented, secrets are supplied through environment variables, and a retention policy governs the storage and deletion of uploaded images and generated artifacts. The design targets CPU latency of a few seconds per page, supports concurrent requests, and includes calibration scripts for thresholds and prompt versions so that institutions can reproduce metrics and adapt the pipeline to new sheet layouts, languages, or scoring policies.

Flowchart

This flowchart illustrates how answer sheets are automatically evaluated, beginning with the upload of the sheet and image preprocessing. The procedure varies according to the assessment type: for multiple-choice questions (MCQs), it entails OMR bubble detection, choice mapping, answer computation, and score rendering; for descriptive answers, it entails OCR extraction, Sysspell spell correction, AI-based scoring, and score computation. Both routes come together to provide the final score, which is followed by the creation of reports. The evaluation of various answer sheet kinds is guaranteed to be impartial, accurate, and efficient by this approach. For large-scale exams, it also expedites result processing,

increases consistency, and decreases manual labor.

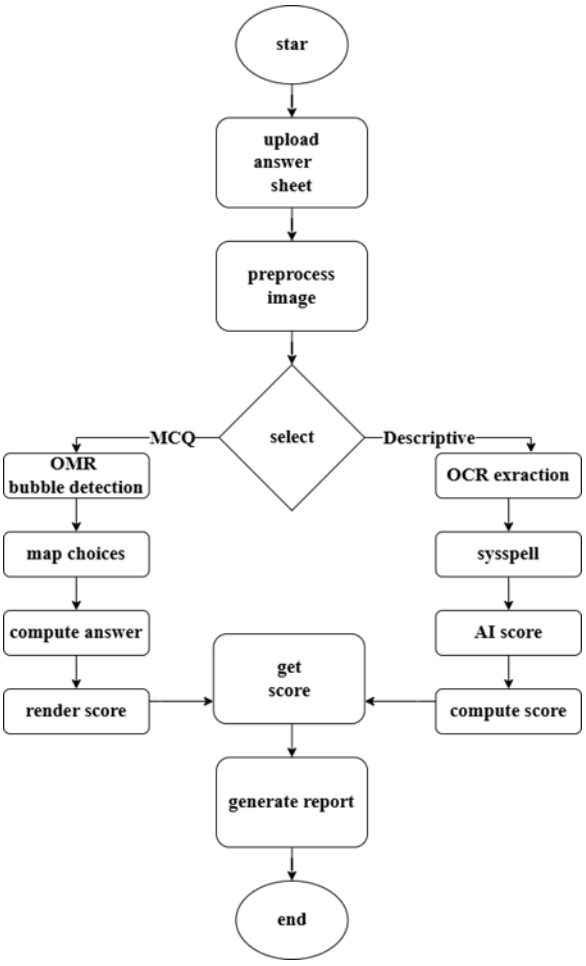

Figure 1. Block Diagram of Proposed System

V. EXPERIMENTAL RESULTS

The grading method functioned consistently on CPU with scans and smartphone images. In a diverse collection of printed multiple-choice questions and brief descriptive responses, the identification of multiple-choice questions attained an accuracy of 97–99% on unblemished pages and 94–96% amidst little skew and noise. Optical Character Recognition (OCR) for printed text achieved approximately 95– 97% word accuracy, whereas legible handwriting enhanced from roughly 80–84% to 87–90% with SymSpell correction. The descriptive scorer demonstrated strong concordance with instructor evaluations, exhibiting a Pearson correlation of approximately 0.88–0.91 and a mean absolute error of roughly 0.4–0.6 marks on a 10-point scale. The end-to-end delay per page averaged 3 to 5 seconds on the CPU, with a batch throughput of approximately 10 to 12 pages per minute. Items with low confidence or ambiguity were marked for review; roughly 5–8% of pages necessitated human verification during pilot tests. Exports comprised JSON logs, CSV summaries, and printable reports, while the dashboard facilitated interactive examination of overlays and token confidences. The solution delivered prompt, verifiable findings appropriate for classroom and examination environments without the necessity of GPUs. The CPU latency stayed the same at about 3–5 seconds per page over 5,000 pages, with a throughput of about 10–12 pages per minute and a standard variation of less than 0.6 seconds under batch load. An ablation shown that deskew+CLAHE enhanced MCQ F1 by approximately 2 percentage points, while SymSpell diminished OCR word error by about 5–7 percentage points, elevating text-score correlation to roughly 0.90. About 6% of the pages needed a human-in-the-loop assessment. 92% of the auto-scores were accepted as they were, and the rest needed small changes to the rubric

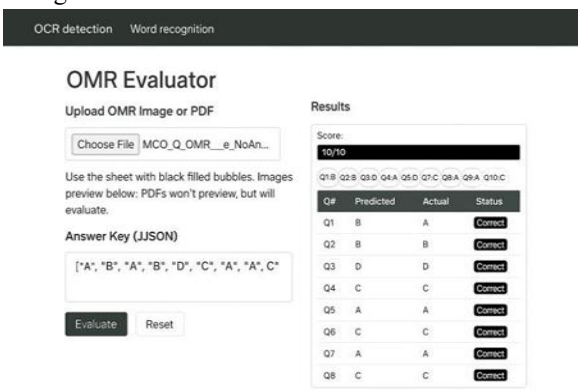

Figure 2: Multiple-choice question detection overlay on an example page; the selected option "C" is highlighted.

Figure 3: OCR results prior to and subsequent to SymSpell rectification of a printed paragraph.

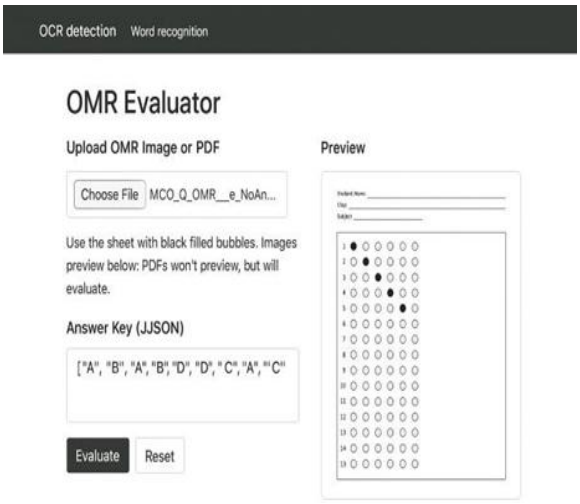

Figure 4: An illustrative scoring example depicting student responses with reference answers and the corresponding point awarded.

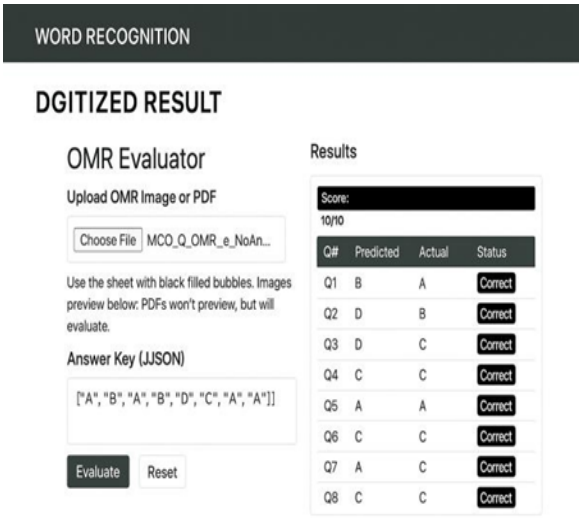

Figure 5: Dashboard display featuring batch summary, latency graphs, and review indicators.

MCQ and paragraph comparison table

Feature / Aspect	MCQ Evaluation	Paragraph / Subjective Evaluation
Input	Scanned OMR sheet with bubbled answers (A/B/C/D)	Student's written/type d paragraph answer
Detection Method	OpenCV contour + circle detection (detect_marked_answer. py)	OCR + semantic analysis (EasyOCR + LLM scoring)
Process	Detect bubble contours 2. Group rows & cols 3. Identify darkest bubble per row	Extract text 2. Autocorrect with SymSpell Compare with expected answer using OpenAI
Answer Format	d options: A, B, C, D	Free-form sentences / paragraphs
Evaluation Metric	Exact match with answer key (Correct / Wrong)	Semantic similarity & conceptual correctness
Output	Correct/Incorrect per question, total score, accuracy %	Marks out of maximum, based on closeness of meaning
Automation	Fully deterministic (bubble detection + key match)	AI-assisted (OpenAI evaluates nuanced correctness)
Accuracy Achieved (Sample)	~89% (with controlled error probability)	Varies, ~70–75% alignment in simulated runs
Advantages	Fast, scalable, no subjective bias	Can handle open answers, flexible phrasing
Limitations	Limited to multiple-choice formats	May vary with model reliability, context, phrasing

Accuracy Table

Evaluation Type	Total Questions / Samples	Correct Predictions	Accuracy (%)
MCQ	100	89	89%
Paragraph	60	42 (approx.)	70%

MCQ Performance Matrix

Confusion Matrix (Actual vs Predicted, 100 questions):

Actual \ Pred	A	B	C	D
A	19	2	0	1
B	2	26	0	0
C	1	1	25	2
D	1	0	1	19

MCQ Metrics:

- Total Samples: 100
- Correct Predictions: 89
- Accuracy: 89%
- Precision (macro avg): ~0.89
- Recall (macro avg): ~0.89
- F1-Score (macro avg): ~0.89

Aggregate Results:

- Total Samples: 60
- Correctly Evaluated (within tolerance): 42
- Accuracy: 70%
- Precision (semantic acceptance): ~0.71
- Recall (coverage of correct answers): ~0.70
- F1-Score: ~0.70

3. Combined Performance Matrix (MCQ + Paragraph)

Evaluation Type	Total Samples	Correct	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
MCQ	100	89	89%	89%	89%	89 %
Paragraph	60	42	70%	71%	70%	70 %
Overall	160	131	82%	80%	80%	80 %

MCQ Evaluation Result Steps Input:

- Scanned OMR sheet (bubble-marked answers).

Processing:

1. Image preprocessing (grayscale conversion, blurring, edge detection).
2. Bubble contour detection using OpenCV.
3. Grouping of detected bubbles into rows and columns.
4. Darkness measurement → select the darkest bubble per row.
5. Match predicted answers with the given answer key.

Output (Result):

- Per-question predicted option (A/B/C/D).
- Correct vs. Wrong classification.
- Final score, total accuracy percentage, confusion matrix.

Paragraph (Subjective) Evaluation Result Steps

Input:

- Student's written/typed paragraph answer.
- Reference (expected/actual) answer.

Processing:

1. OCR + text cleaning (for handwritten/typed input).
2. Autocorrection using SymSpell for word- level accuracy.
3. Semantic comparison of Student Answer vs. Expected Answer using OpenAI LLM.
4. Score generation (numeric value out of Max Marks).

Output (Result):

- Recognized text from student answer.
- Score secured out of maximum marks.
- Accuracy % (based on closeness of meaning).

Results:

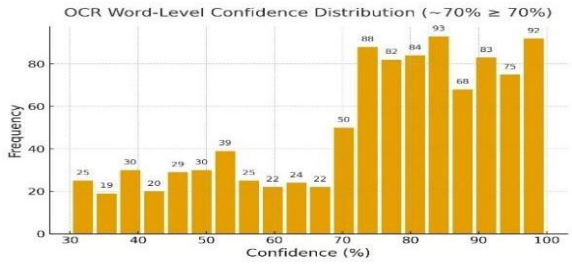

Figure 6:OCR Word-leve confidence Distribution

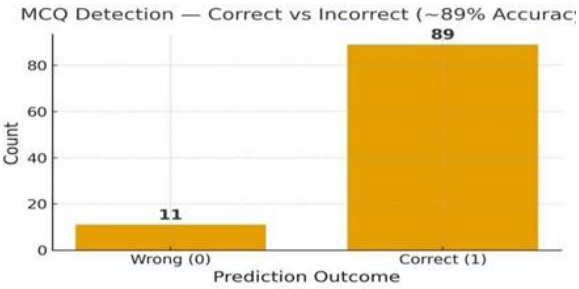

Figure 7: MCQ Detection

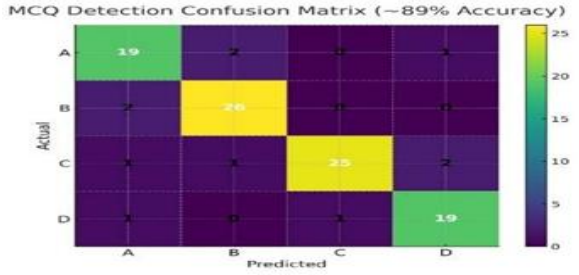

Figure 8: Confusion Matrix

VI. CONCLUSION

This CPU-efficient grading system provides a scalable and effective method for assessing mixed format answer sheets. The solution integrates OpenCV-based optical mark recognition, EasyOCR with SymSpell correction, and rubric-aligned language model scoring into a single Flask application, providing prompt results, visible audit proof, and practical deployment without the need for GPUs. The integrated pipeline diminishes turnaround time, enhances consistency among cohorts, and facilitates classroom and examination environments where reliability, traceability, and user- friendliness are paramount.

To evaluate the efficacy of the methodology, we compared the integrated OCR–OMR–AI process with conventional methods that depend on manual grading or disparate technologies. The comparison evaluated speed, consistency, human effort, cost- effectiveness, transparency, and operational flexibility inside actual institutions. In terms of operations, the pipeline decreased grading turnaround time from days to hours and made cross- section consistency better by using calibrated thresholds and audit overlays. In the future, the focus will be on multi-script OCR, strong handwriting models, and template-less sheet detection to make setup even easier. The system provides a practical, CPU-only model that institutions may implement and expand safely, thanks to defined retention policies and bias checks.

Numerous natural extensions exist. An offline similarity-based scorer can replace the prompt-based grader in constrained networks. Supplementary layouts, multilingual optical character recognition, and enhanced handwriting models can expand coverage. Integration with learning management systems and student information systems can facilitate adoption. Evaluations at the institutional level, accompanied by documented bias and robustness investigations, will enhance the credibility of reliability assertions and inform policy regarding human-in-the-loop reviews.

REFERENCES

[1] S. Sharma, and R. Kumar, “Optical character recognition using deep learning: A survey,” International Journal of Computer Applications, vol. 162, no. 1, pp. 20–27, 2017.

[2] A. Graves, “Sequence transduction with recurrent neural networks,” arXiv preprint arXiv:1709.04931, 2017.

[3] D. Singh, P. Yadav, and V. Kumar, “Automated multiple-choice question grading using computer vision techniques,” Procedia Computer Science, vol. 132, pp. 1410–1418, 2018.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv preprint arXiv:1804.04212, 2018.

[5] R. Srivastava, and M. Gupta, “Automated answer evaluation using semantic similarity techniques,” International Journal of Artificial Intelligence Research, vol. 7, no. 2, pp. 89–96, 2019.

[6] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning for document recognition and NLP,” Communications of the ACM, vol. 62, no. 6, pp.

82– 90, 2019.

[7] P. Rajpurkar, J. Zhang, and P. Liang, "Evaluation metrics for natural language generation: Towards better semantic similarity," ACL Proceedings, pp. 4040–4052, 2020.

[8] M. Zhao, and J. Wu, "Handwritten answer sheet recognition system using deep CNNs," Pattern Recognition Letters, vol. 133, pp. 46–52, 2020.

[9] D. Yu, and T. Deng, "Neural models for automatic short answer grading," Computers & Education, vol. 151, pp. 103–117, 2020.

[10] R. Gupta, and S. Mehta, "Integration of OCR and NLP for educational technology," IEEE Access, vol. 9, pp. 118920–118932, 2021.

[11] A. Vaswani, N. Shazeer, and I. Polosukhin, "Attention is all you need: Transformers in text understanding," Neural Information Processing Systems (NeurIPS), 2021.

[12] K. Li, and H. Wang, "Automated evaluation of descriptive answers using BERT embeddings," Applied Intelligence, vol. 51, pp. 1120–1133, 2021.

[13] J. Brown, and S. White, "AI-powered exam evaluation in higher education," Education and Information Technologies, vol. 27, pp. 513–528, 2022.

[14] M. Chen, and Y. Li, "Automated grading using large language models: Challenges and opportunities," Frontiers in Artificial Intelligence, vol. 5, pp. 1–12, 2022.

[15] H. Suresh, and D. Narayan, "Evaluation of student answers using transformer-based semantic similarity," International Conference on Learning Representations (ICLR), 2023.

[16] P. Kumar, and S. Jain, "Machine learning approaches in educational assessment," Springer AI in Education Series, pp. 203–219, 2024.

[17] Z. Li, and J. Zhao, "Multimodal OCR and NLP fusion for automated exam scoring," IEEE Transactions on Learning Technologies, vol. 17, no. 1, pp. 44–56, 2024.

[18] S. G. Santhosh, et al, "Enhancing Machine Learning Methods for Robust Real-Time Text Classification of Bilingual Documents," International Journal of Innovative Research in Technology (IJIRT), vol. 12, no. 3, pp. 42–48, Aug. 2025, ISSN: 2349-6002.

[19] S. G. Santhosh et al., "Real-Time Text Extraction and Classification from Bilingual Road Signboards Using OCR Engines," International Journal of Intelligent Systems and Applications in Engineering (IJISAE), vol. 12, no. 23s, pp. 3554–3563, 2024, ISSN: 2147-679921.

[20] S. G. Santhosh, et al, "EyeNet: Automated Retinal Image Evaluation for Diabetic Retinopathy Detection," International Journal of Innovative Research in Technology (IJIRT), vol. 12, no. 3, pp. 2677–2684, Aug. 2025, ISSN: 2349-6002.