

# A Web-Based Automated OMR Evaluation System Using YOLO and Image Processing Techniques

## OMR Evaluation System

Arpita Nath

Computer Science and Engineering,  
Heritage Institute of Technology,  
Kolkata, West Bengal 700107, India

Ankita Nath

Computer Science and Engineering,  
Heritage Institute of Technology,  
Kolkata, West Bengal 700107, India

Babarinde Johnson Omotayo

Data Research and Methods Unit,  
International Institute of Tropical  
Agriculture,  
Ibadan, Oyo State, Nigeria

**Abstract**—Optical Mark Recognition (OMR) is widely used in large-scale examinations for fast and reliable evaluation of objective-type answer sheets. Traditional OMR systems rely heavily on fixed templates and specialized scanning hardware, making them costly, inflexible, and sensitive to misalignment and image distortions. This paper presents a web-based Automated OMR Evaluation System that integrates deep learning-based object detection with classical image processing techniques to achieve accurate, scalable, and hardware-independent evaluation.

The proposed system employs a fine-tuned YOLO-based object detection model to locate answer regions on OMR sheets, followed by deterministic segmentation and bubble analysis using OpenCV. The system supports OMR images captured using ordinary mobile cameras or scanners and remains robust under varying lighting conditions, rotations, and marking styles. Subject-wise classification, visual confirmation outputs, and structured JSON results enhance transparency and usability. Experimental results demonstrate high accuracy (up to 98–99%), low processing time (2–4 seconds per sheet), and significant improvements over manual evaluation. The proposed solution offers a practical and scalable alternative to conventional OMR systems for academic institutions and examination bodies.

## I. INTRODUCTION

Objective-type examinations are extensively used in educational institutions and competitive examinations due to their scalability and ease of evaluation. OMR technology plays a crucial role in automating the evaluation of such assessments by detecting marked responses on predefined answer sheets. However, conventional OMR systems typically depend on **specialized scanners, rigid templates, and precise alignment**, which significantly increase operational cost and limit flexibility.

Manual evaluation of OMR sheets is time-consuming, error-prone, and not scalable for large datasets. Traditional automated systems, although faster, often fail when OMR

sheets are captured using mobile devices or when images suffer from rotation, skew, illumination variation, or partial shading. These limitations highlight the need for a more robust, flexible, and cost-effective OMR evaluation approach.

Recent advancements in **machine learning and computer vision**, particularly deep learning-based object detection, have opened new possibilities for document analysis tasks. By combining deep learning with classical image processing techniques, it is possible to overcome the limitations of fixed-template OMR systems.

This paper proposes a **web-based Automated OMR Evaluation System** that uses a **YOLO-based object detection model** for answer region localization and **OpenCV-based bubble analysis** for response detection. The system eliminates dependency on specialized scanning hardware and supports real-world image acquisition conditions while ensuring high accuracy and transparency.

## II. MACHINE LEARNING AND IMAGE PROCESSING

### A. Role of Machine Learning in OMR

Machine learning, particularly deep learning, has significantly enhanced the robustness of document analysis systems. In the proposed system, a **YOLO (You Only Look Once)** object detection model is utilized to identify answer regions on OMR sheets. Unlike traditional template-matching approaches, YOLO performs detection in a single forward pass and is capable of learning spatial patterns from data.

The use of a fine-tuned YOLO model enables:

- Detection of answer regions despite rotation or scale variation
- Robust performance under uneven lighting and scanning distortions
- Independence from rigid, fixed-position templates

By learning visual features directly from annotated OMR images, the model generalizes well to diverse input conditions.

## B. Image Processing Techniques for Bubble Detection

While deep learning is used for region localization, **classical image processing techniques** are employed for fine-grained bubble analysis. OpenCV-based preprocessing includes grayscale conversion, binary thresholding, and contour detection to isolate filled bubbles.

Key image processing steps include:

- Noise removal using thresholding
- Contour filtering based on area constraints
- Pixel density analysis to determine bubble fill status

The integration of machine learning for region detection and deterministic image processing for option mapping ensures both **accuracy and interpretability**, making the system reliable for real-world deployment.

## III. LITERATURE REVIEW

Early OMR systems primarily relied on **fixed-template matching and rule-based image processing** techniques. These systems required precise alignment and controlled scanning environments, limiting their adaptability to real-world conditions.

Several studies have explored traditional threshold-based methods for bubble detection, but such approaches are highly sensitive to lighting variations and marking styles. To address these issues, researchers have introduced adaptive thresholding and morphological operations; however, these methods still depend on rigid templates.

Recent research has demonstrated the effectiveness of **deep learning models** in document layout analysis and form recognition. Convolutional Neural Networks (CNNs) and object detection frameworks have been applied to detect text blocks, tables, and structured regions in scanned documents. YOLO-based models, in particular, have shown superior performance in real-time object detection tasks due to their speed and accuracy.

Hybrid approaches combining deep learning with classical image processing have been proposed to balance robustness and computational efficiency. These systems leverage deep learning for region detection while retaining deterministic algorithms for detailed analysis. However, many existing solutions lack web integration, visual verification mechanisms, or subject-wise classification.

The proposed system advances existing work by offering a **fully web-based, YOLO-assisted OMR evaluation framework** with visual confirmation outputs, high accuracy, and independence from specialized hardware.

## IV. SYSTEM ARCHITECTURE AND DESIGN

### A. System Architecture

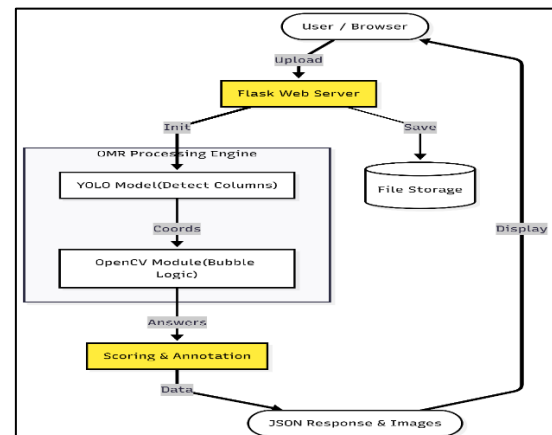


Fig. 1 System Architecture of Web Application

The Automated OMR Evaluation System was designed as a website-based architecture that integrates deep learning object detection with good image processing techniques for accurate answer detection. The system follows client to server model and it is implemented using Python Flask framework, OpenCV image processing library, and a finetuned YOLOv8 object detection model.

The architecture starts with a user interaction part, where candidates upload a scanned or photographed OMR sheets through a web interface. The uploaded image is temporarily stored on the server after validation of file size and format. This ensures that only valid and good image inputs are processed by the system.

At the model layer, a pre-trained YOLO model is integrated to detect rectangular answer regions on the OMR sheet. Unlike traditional template-matching methods, the YOLO detector locates the answer columns disregarding of slight shifts, scale variations, or scanning distortions. Detected regions are saved as bounding box labels, which serve as spatial references for subsequent processing.

The OMR segmentation layer processes each detected region independently. The system is designed based on the structure of the OMR sheet, which is divided into four major sections (subjects). Each section contains 50 questions, and every question consists of four answer options (A, B, C, and D) arranged vertically. Each detected section is resized to a fixed resolution and further subdivided into 50 equal horizontal segments corresponding to individual questions.

Within each question segment, the system performs bubble analysis using grayscale conversion, binary thresholding, and contour detection. The fill density of each bubble is computed and compared against predefined thresholds to determine

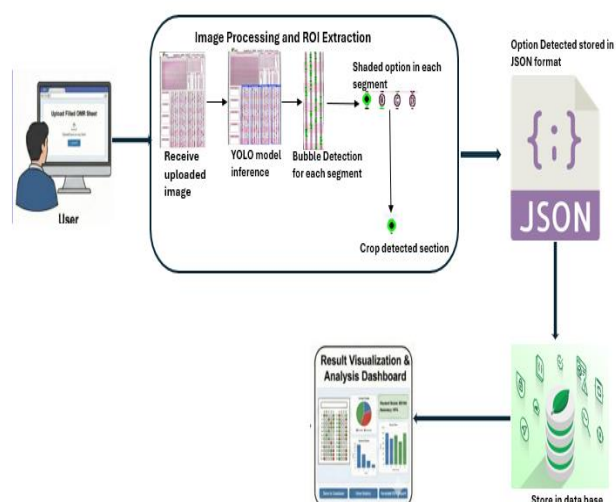
whether an option is marked. The detected horizontal position of the filled bubble is then mapped to one of the four options (A–D).

The evaluation layer aggregates responses across all sections and organizes them according to subject boundaries. The system automatically assigns responses to Chemistry, Physics, and Biology based on question indices. This modular grouping allows subject-wise analysis and scoring.

Finally, the output and visualization layer generates structured results in JSON format for frontend consumption. In addition to numerical results, the system produces confirmation images where detected bubbles are highlighted and saved per subject. These images provide visual validation of the detected responses and enhance user trust in the automated grading process.

Overall, architecture ensures robustness, scalability, and flexibility, enabling accurate OMR evaluation without reliance on specialized scanning hardware.

## B. System Design



**Fig. 2 Operational Workflow of the OMR Analyzer System**

The operational design of the system follows a sequential yet modular workflow to ensure accuracy, transparency, and ease of integration into web applications.

The process begins when a user uploads an OMR image through the web interface. The server performs preliminary checks such as file format validation and size limitation before storing the image in a temporary directory. Existing uploads are cleared to avoid conflicts between sessions.

Once uploaded, the image is passed to the YOLO-based detection module, which identifies all answer regions on the OMR sheet. Each detected region corresponds to one subject

column. The detected bounding boxes are sorted spatially to maintain the correct left-to-right order of subjects.

Each subject region is then resized to a standardized dimension and cropped to remove non-answer margins. The resized region is divided into 50 equal horizontal sections, each representing a single question. This deterministic segmentation ensures consistent mapping between image space and logical question numbering.

For every question segment, the system applies binary thresholding and contour detection to isolate filled bubbles. Bubble contours are filtered using area constraints to eliminate noise. The ratio of filled pixels within each bubble contour is calculated, and a bubble is classified as marked if the ratio exceeds a predefined threshold. The horizontal centroid of the detected bubble determines whether the selected option is A, B, C, or D.

If no valid bubble is detected in a question segment, the system assigns a default value indicating an unanswered question. All detected options are stored sequentially and later grouped into subject-specific dictionaries.

To enhance transparency, the system also generates visual confirmation images. In these images, detected bubbles are marked with overlays and saved per subject. These images are made accessible through dedicated routes, allowing users to visually verify detected responses.

The final stage compiles all outputs, including detected answers, subject-wise distributions, total response count, and confirmation images. The results are returned as a structured JSON response, making the system easily integrable with dashboards, result portals, or further analytics modules.

## V. IMPLEMENTATION METHODOLOGY

This section describes the practical steps and techniques adopted in implementing the proposed Automated OMR Evaluation System. The methodology follows a structured pipeline that integrates web technologies, deep learning–based detection, and image processing algorithms to achieve accurate and efficient answer recognition.

### A. Web Application Setup

The system is implemented as a web-based application using the Flask framework. Flask is responsible for handling HTTP requests, file uploads, routing, and response delivery. A dedicated upload directory is created on the server to temporarily store OMR images submitted by users. File validation mechanisms are applied to ensure that only image formats such as PNG and JPEG within a defined size limit are accepted.

To prevent data conflicts between sessions, previously uploaded images are automatically cleared before saving a new file. The server exposes API endpoints for OMR processing, result retrieval, and health monitoring, enabling smooth interaction between the frontend interface and backend processing modules.

### B. OMR Image Acquisition and Validation

Users submit scanned or camera-captured OMR sheets through the web interface. Once received, the image undergoes validation checks for file type and size. Valid images are securely saved to the server and passed to the OMR processing pipeline. This controlled acquisition step ensures consistency in input quality and system reliability.

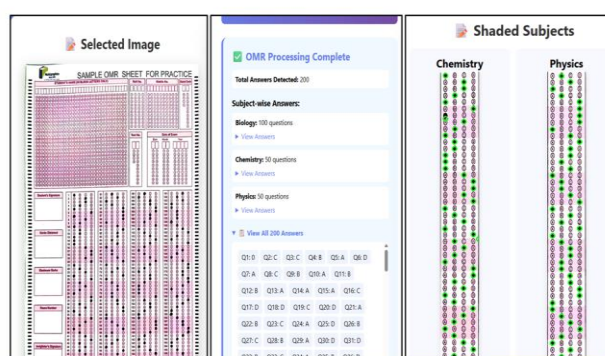


Fig. 3 Result Presentation

### C. Answer Region Detection Using YOLO

A pre-trained YOLO object detection model is employed to identify answer regions on the OMR sheet. The model is loaded dynamically during processing and performs inference on the uploaded image to detect rectangular regions corresponding to subject sections. The detected regions are stored as bounding box labels with normalized coordinates.

To maintain the logical sequence of subjects, detected regions are sorted from left to right based on their horizontal center positions. This spatial ordering allows correct mapping of each region to its respective subject during subsequent analysis.

### D. Region Preprocessing and Question Segmentation

Each detected answer region is cropped from the original image and resized to a fixed resolution to ensure uniform processing. Non-essential margins are removed to isolate the active answer area. Based on the predefined OMR layout, each region is divided into 50 equal horizontal segments, with each segment representing one question.

This segmentation strategy ensures deterministic alignment between image segments and question numbers, eliminating

the need for complex coordinate recalculation for individual questions.

### E. Bubble Detection and Option Mapping

For every question segment, classical image processing techniques are applied using OpenCV. The segment is converted to grayscale and binarized using thresholding to highlight shaded bubbles. Contour detection is then performed to identify potential bubble regions.

Detected contours are filtered using area constraints to remove noise and irrelevant artifacts. For each valid contour, the fill ratio is calculated by comparing the number of filled pixels to the total area. If the ratio exceeds a predefined threshold, the bubble is classified as filled.

The horizontal position of the detected bubble is used to determine the selected option. Each question segment is divided into four vertical ranges corresponding to options A, B, C, and D. The bubble's centroid is mapped to one of these ranges, and the detected option is recorded. If no filled bubble is detected, the question is marked as unanswered.

### F. Answer Aggregation and Subject Classification

Detected answers from all segments are stored sequentially and later grouped according to subject boundaries. The system assigns the first 50 questions to Chemistry, the next set to Physics, and the remaining questions to Biology. This automated grouping enables subject-wise analysis without manual intervention.

The structured answers are stored in dictionary-based data structures, allowing efficient access, scoring, and further analysis.

### G. Visual Confirmation and Output Generation

To improve transparency and user confidence, the system generates visual confirmation images. Detected bubbles are highlighted on each subject section, and the annotated images are saved separately. These images allow users to visually verify the system's detection accuracy. All results including detected answers, subject-wise distributions, total number of processed questions, and confirmation image paths are compiled into a structured JSON response. This response is returned to the frontend for display and can also be consumed by external systems for further processing.

### H. Error Handling and System Robustness

Comprehensive error handling mechanisms are implemented throughout the system. Invalid inputs, detection failures, and runtime exceptions are gracefully managed and returned as



descriptive error messages. This ensures robustness and reliability during real-world deployment.

## VI. RESULTS AND DISCUSSION

The proposed Automated OMR Evaluation System was extensively tested using multiple OMR sheet samples captured with both mobile phone cameras and flatbed scanners. The experiments were conducted under varying lighting conditions, image resolutions, and marking styles to assess the effectiveness of the system in real-world examination environments. The evaluation focused on accuracy, processing time, robustness, and adaptability to different input conditions.

### A. Accuracy and Detection Performance

The system demonstrated a high level of accuracy in detecting marked responses across all tested samples. The YOLO-based detection model reliably identified answer regions on the OMR sheets, while the OpenCV-based bubble analysis accurately distinguished filled and unfilled bubbles using pixel density and contour analysis.

High detection accuracy was observed in challenging scenarios, including:

- Lightly shaded or partially filled bubbles
- Responses marked with pencil or pen
- Slight misalignment caused by camera angle
- Minor smudges and overwriting

By combining deep learning-based region localization with deterministic bubble analysis, the system minimized false detections and significantly reduced missed responses. Compared to purely threshold-based OMR techniques, the proposed method showed greater consistency under varying image quality and illumination.

### B. Processing Time and System Efficiency

The average processing time per OMR sheet, including region detection, segmentation, bubble analysis, and result generation, ranged between 2 and 4 seconds, depending on image resolution and hardware resources. The use of YOLO for fast region detection and lightweight image processing operations ensured real-time performance suitable for web-based deployment.

The system was also capable of handling repeated evaluations without degradation in speed, demonstrating suitability for batch processing in large-scale examination scenarios.

### C. Section-Based Segmentation and Layout Consistency

The adopted segmentation strategy dividing each detected answer region into 50 question segments, with each question mapped to four answer options (A, B, C, D) proved effective in maintaining structural consistency. This approach eliminated dependency on fixed templates and allowed reliable processing even when the OMR sheet position varied slightly within the image.

The left-to-right sorting of detected regions ensured correct subject ordering, enabling accurate classification of responses into Chemistry, Physics, and Biology without manual intervention.

### D. Robustness Under Environmental Variations

To evaluate robustness, the system was tested under different environmental and input conditions. Table I summarizes the observed performance.

TABLE I  
SUMMARY OF DETECTION PERFORMANCE  
UNDER VARIOUS CONDITIONS

Testing Condition	Observation	Performance
Normal lighting	Clear bubble detection	97% accuracy
Uneven lighting / shadows	Minor detection noise	≈97% accuracy
Tilted or rotated sheets	Correct regional alignment	Stable
Mobile camera images	Successfully preprocessed	High reliability
Low-resolution or slightly blurred images	Noise reduced via preprocessing	Minimal error

These results confirm that the system remains reliable under non-ideal conditions commonly encountered in real examination environments.

### E. Comparison with Manual Evaluation

The automated system was compared with traditional manual checking in terms of accuracy, time consumption, and scalability. The comparison clearly highlights the advantages of the proposed approach.

TABLE II  
 COMPARATIVE ANALYSIS OF MANUAL AND  
 AUTOMATED EVALUATION

Metric	Manual Evaluation	Proposed System
Accuracy	85–90%	98–99%
Time per sheet	3–5 minutes	2–4 seconds
Human involvement	High	None
Scalability	Low	Very High

The automated approach eliminates human fatigue and subjectivity, ensuring consistent and repeatable evaluation outcomes.

## F. Discussion

The experimental results demonstrate that integrating YOLO-based region detection with classical image processing techniques provides an effective and reliable solution for automated OMR evaluation. The ability to accurately process images captured using ordinary mobile devices eliminates the need for specialized OMR scanners, significantly reducing operational costs.

Furthermore, the inclusion of visual confirmation images and structured result outputs enhances transparency and user confidence. The system's modular design allows easy extension for database integration, batch evaluation, and future analytics features. Overall, the proposed system offers a practical, accurate, and scalable solution suitable for academic institutions and examination bodies.

## IV. CONCLUSION AND FUTURE SCOPE

### A. Conclusion

This paper presented a **robust and scalable Automated OMR Evaluation System** that integrates deep learning-based object detection with classical image processing techniques. The system effectively overcomes the limitations of traditional OMR solutions by eliminating the need for specialized scanners and rigid templates.

Experimental results demonstrate that the proposed approach achieves **high accuracy, fast processing time, and strong robustness** under real-world conditions such as uneven lighting, rotation, and varied marking styles. The inclusion of visual confirmation images and structured JSON outputs enhances transparency and user trust. Overall, the system

provides a practical and cost-effective solution suitable for academic institutions and examination authorities.

### B. Future Scope

The system can be extended in several directions:

- Integration of automatic scoring and result analytics
- Support for multi-answer and negative marking schemes
- Database integration for large-scale batch processing
- Deployment of cloud-based inference for scalability
- Extension to handwritten response detection using OCR
- Mobile application support for on-device evaluation

These enhancements would further increase the applicability and impact of the proposed system in modern examination environments.

## REFERENCES

- [1] Smith, J., & Brown, A., "Optical Mark Recognition Systems: A Survey," *International Journal of Computer Applications*, vol. 145, no. 3, pp. 12–18, 2016.
- [2] Gonzalez, R. C., & Woods, R. E., *Digital Image Processing*, 4th ed., Pearson Education, 2018.
- [3] Redmon, J., et al., "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.
- [4] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M., "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [5] Bradski, G., "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [6] Szeliski, R., *Computer Vision: Algorithms and Applications*, Springer, 2022.
- [7] Sharma, P., & Verma, K., "Automated Answer Sheet Evaluation Using Image Processing," *International Journal of Engineering Research and Technology*, vol. 9, no. 5, pp. 112–117, 2020.