JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Automatic Valuation of OMR using Computer vision techniques

Sumanth K, Pannaga Ram P R

[1]Student, [2]Student

[1 2]Dept. of CSE (AI & ML)
[1 2]RNS Institute of Technology, Bengaluru, India

Abstract— Optical Mark Recognition (OMR) systems have become increasingly vital in educational environments for automating the evaluation of multiple-choice answer sheets. This project presents a Python-based OMR solution using OpenCV to detect and evaluate filled bubbles representing student responses. A novel aspect of this system is its capability to extract a student's Unique Student Number (USN) directly from the bubbled ID section on the answer sheet. The methodology involves a sequence of image processing techniques including grayscale conversion, inverse binary thresholding, contour detection, and morphological filtering to accurately identify marked choices. The system also employs predefined regions of interest to localize both the answer and identification zones. Additionally, results are organized and stored in an Excel file for easy access and record-keeping. This approach reduces manual effort, increases accuracy, and supports scalability for academic institutions. The system was tested on structured OMR sheets with predefined coordinates and demonstrated high precision in both answer recognition and USN extraction. The combination of automation and data accuracy offers a practical, low-cost alternative to commercial OMR scanners, making it well-suited for use in schools, colleges, and online examination platforms.

Index Terms— Optical Mark Recognition (OMR),OpenCV,Image Processing,Automated Grading,Contour Detection ,Thresholding ,USN Extraction,Computer Vision .

I. INTRODUCTION

The growing demand for scalable and accurate assessment methods in educational institutions has prompted the adoption of automated evaluation systems. Particularly in environments that conduct mass testing, manual grading of multiple-choice questions (MCQs) becomes inefficient, time-consuming, and prone to human error. Optical Mark Recognition (OMR) technology offers a practical solution for automating this process by detecting filled bubbles on answer sheets. However, traditional OMR setups often depend on costly scanners and proprietary platforms, which restrict their accessibility for small or underfunded institutions.

Recent developments in computer vision and open-source frameworks like OpenCV have introduced new opportunities for designing affordable, adaptable OMR systems. These advancements make it possible to use standard cameras, including those on smartphones, for capturing answer sheets. Through effective preprocessing techniques—such as grayscale conversion, inverse binary thresholding, contour analysis, and morphological filtering—software can accurately interpret marked responses without the need for high-end scanning equipment.

The proposed system in this project leverages Python and OpenCV to implement an end-to-end solution for automated answer detection and student identification. It processes images of bubble sheets, isolates filled options using thresholding and contour detection, and deciphers alphanumeric student identifiers (USNs) from pre-defined bubble layouts. The logic is robust enough to handle various lighting conditions and minor misalignments in the image.

Unlike proprietary software, this model is completely customizable and can be modified to accommodate different exam formats. The project also includes automated report generation using the OpenPyXL library, which exports student responses into structured Excel sheets. This eliminates manual data entry, accelerates the evaluation process, and reduces operational errors.

In educational contexts, especially in developing regions, the availability of such cost-effective automation tools can significantly enhance examination management. By removing reliance on specialized hardware and commercial licenses, institutions gain the flexibility to scale assessments without incurring prohibitive costs. This project demonstrates how image processing techniques and open-source tools can be combined to deliver a practical, low-budget OMR system for real-world academic applications.

II. LITERATURE SURVEY

Optical Mark Recognition (OMR) has long served as a cornerstone in the digitization of student assessments, particularly for multiple-choice question (MCQ) exams. Traditional OMR systems typically involve dedicated scanners and pre-printed forms with fixed layouts, which are often cost-prohibitive and inflexible. However, recent developments in image processing and machine learning have enabled the transition from hardware-based OMR systems to software-driven, camera-compatible solutions.

Several studies have demonstrated the feasibility of using open-source tools like OpenCV for OMR applications. Tümer et al. (2018) proposed an automated grading system using Python and OpenCV, focusing on real-time recognition of answer bubbles through thresholding and contour detection. Their system achieved high accuracy while reducing reliance on commercial software. Patil and Dhongare (2025) further extended this approach by integrating adaptive thresholding techniques to handle variable lighting conditions and noise, ensuring that even low-quality images could be processed accurately.

USN or student ID extraction has also gained importance in automated OMR systems, particularly in large-scale examinations where accurate candidate identification is essential. Adhikari (2019) presented a roll number recognition method where ID bubbles are segmented and processed separately from answer sections. The system relies on pixel density and region-of-interest (ROI) cropping to detect marked alphanumeric characters, demonstrating reliable results with standard image preprocessing steps.

Other research, such as that by Kanjalkar et al. (2023), has focused on mobile-based OMR applications. Their system allows users to scan answer sheets using a smartphone, processes the image on-device or via a cloud backend, and produces a marked and graded result. This approach significantly reduces infrastructure costs and increases accessibility in remote or under-resourced educational settings.

Another noteworthy contribution is by Ahad et al. (2019), who emphasized fault-tolerant OMR systems capable of reading torn, smudged, or misaligned sheets. Their methodology includes geometric correction, noise elimination, and pattern recognition to recover data from imperfect inputs, thereby increasing the robustness of OMR applications.

Common across these studies is the use of preprocessing techniques such as grayscale conversion, inverse binary thresholding, and morphological operations. These steps help enhance image quality, isolate marked regions, and eliminate background noise. Once preprocessing is complete, contour detection or blob analysis is used to identify potential bubbles, followed by a pixel-intensity or area-based approach to determine whether a bubble has been filled.

In terms of evaluation and reporting, many modern systems integrate the final results into Excel or similar formats for ease of access and visualization. This feature is particularly useful for educators who need to process and analyze large amounts of data quickly.

Recent advancements also explore the use of artificial intelligence for increasing the reliability of OMR systems. Machine learning models trained to distinguish filled and unfilled bubbles, as well as detect anomalies, are being adopted to improve recognition accuracy. Furthermore, research has begun to address the challenges of varying handwriting, mark density, and environmental noise in scanned sheets. These improvements contribute to making OMR technology not only more accurate but also more inclusive, supporting broader educational contexts and examination formats.

Additionally, some hybrid systems now incorporate cloud computing and IoT-based frameworks for centralized processing and real-time analytics. These solutions enable seamless data aggregation and allow multiple stakeholders— students, teachers, and administrators—to access results securely. The integration of encryption and secure transmission protocols also ensures the integrity and confidentiality of sensitive student data. Studies have shown that such frameworks can scale effectively and be deployed in diverse geographical locations with limited technical infrastructure.

III. METHODOLOGY

The proposed system is designed to automate the evaluation of multiple-choice questions and extract unique student identifiers (USNs) from scanned or photographed OMR sheets. It leverages computer vision techniques and open-source tools to build a low-cost, adaptable, and scalable OMR solution suitable for educational assessments. The methodology consists of multiple stages: image acquisition, preprocessing, thresholding, bubble detection, USN extraction, answer recognition, and result generation. Each stage is described in detail in the below flowchart(Fig-1).

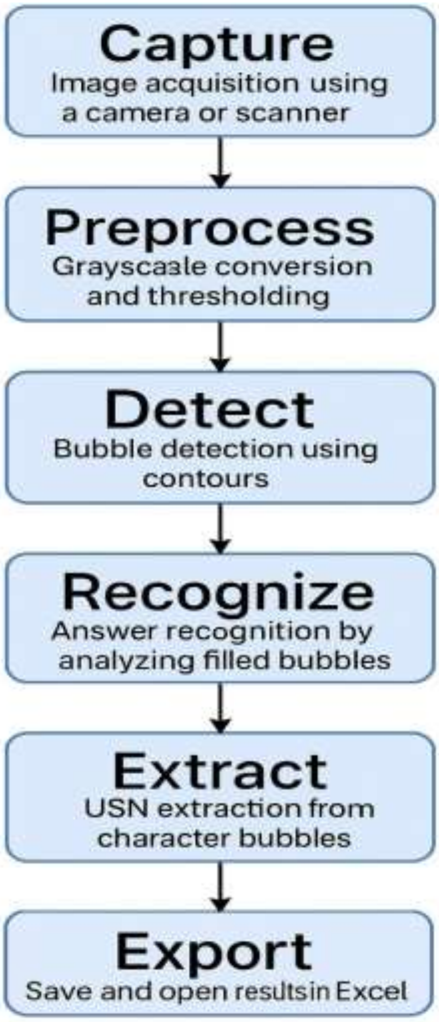

Fig 1: Flowchart of Methodology

A. Image Acquisition

This initial phase involves capturing the answer sheets using a standard scanner or high-resolution mobile camera. Proper alignment, uniform lighting, and high image clarity are essential for accurate detection. The system supports common image formats such as JPEG and PNG, allowing flexibility in input devices. Users can also define custom cropping regions to isolate the answer and USN areas, enabling compatibility with various sheet layouts and formats.

B. Image Preprocessing

After image acquisition, the system converts the input image to grayscale using OpenCV's cv2.cvtColor() method. This reduces computational complexity and enhances the contrast between filled and unfilled bubbles. The grayscale image is further processed using inverse binary thresholding, which distinguishes the dark marks from the background by converting darker pixels to white and lighter ones to black. This inversion simplifies bubble detection and makes the subsequent steps more reliable. Fine-tuning the threshold value improves robustness under different lighting conditions and ink intensities. The Fig-2 and Fig-3 below are the grayscale image and binary threshold image.

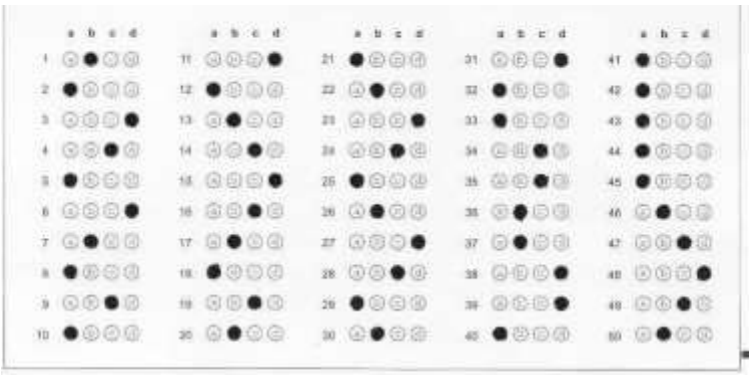

Fig 2- Grayscale Image

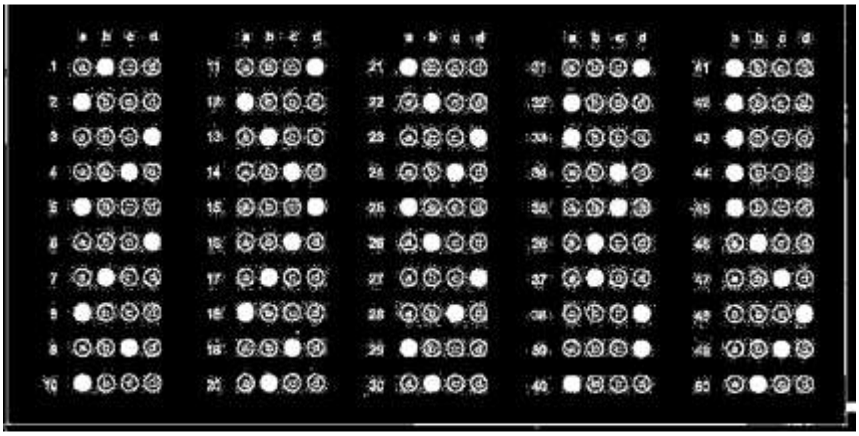

Fig 3- Binary threshold image

C..Bubble Detection

Contours representing potential bubbles are detected using OpenCV's findContours() method. These contours are filtered based on their geometric features—such as area, shape, and aspect ratio—to ensure only valid bubbles are retained. Specifically, bubbles that closely resemble circles or ellipses with a predefined pixel size range (15–60 pixels) and aspect ratio (~1.0) are selected. This minimizes false positives from text or other marks on the sheet. The contours are then sorted logically by rows and columns to align each bubble group with its corresponding question. The Fig 4 shows the contour detection which helps in detecting the bubbles in omr.

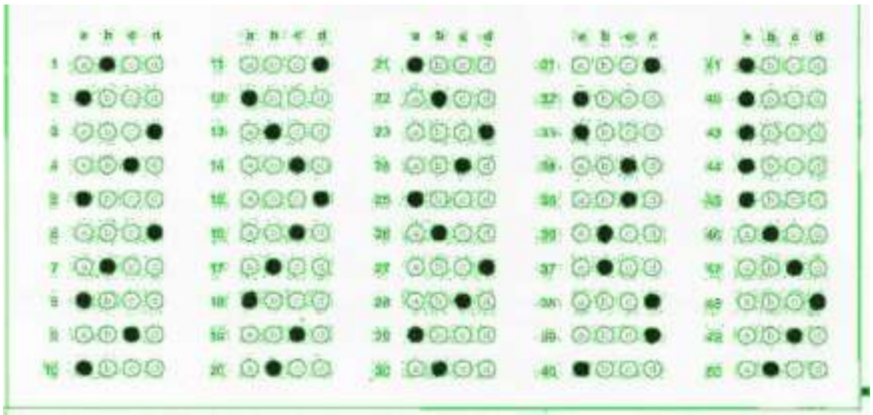

Fig 4 – Contour Detection

D. Answer Recognition

Each row of identified bubbles corresponds to a specific multiple-choice question. The system evaluates the fill level of each bubble by calculating the number of non-zero pixels using cv2.countNonZero(). The bubble with the highest count is assumed to be filled by the student. A mapping mechanism links this selection to answer options (A, B, C, D, etc.). Answers are stored in a structured dictionary, with question numbers as keys and selected options as values. This stage ensures high-speed and consistent answer recognition, reducing manual evaluation errors.

E. USN Extraction

USNs are extracted from a dedicated region on the OMR sheet where students fill in alphanumeric bubbles. Each character (digit or letter) is represented by a vertical set of bubbles, each corresponding to a single character (e.g., A–Z, 0–9). For every column, the bubble with the highest filled pixel count is interpreted as the selected character. These columns are processed sequentially, and each identified character is concatenated to reconstruct the complete USN. The USN acts as a unique key for linking answer data to the respective student.

The Fig 5 represents one the process of the USN identification where the co-ordinates of each column of USN region is taken and is mapped to respective character which helps in identifing the USN

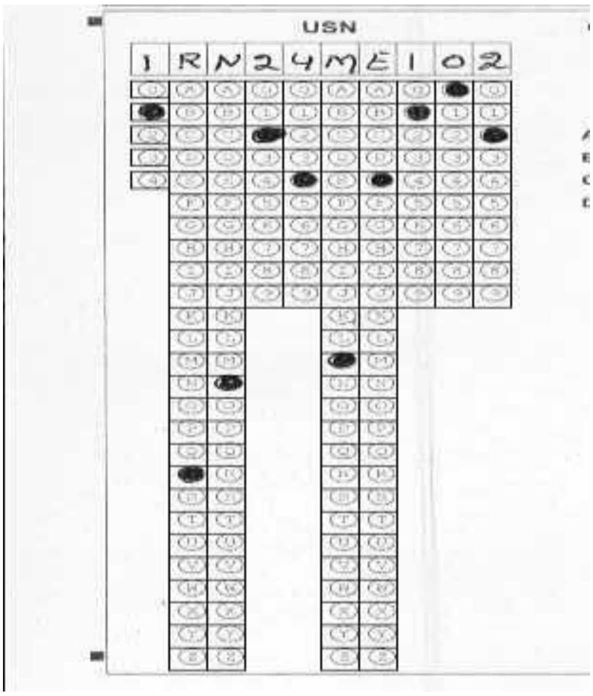

Fig 5 - USN Identification

F. Section Cropping and Modular Processing

To handle OMR sheets with a large number of questions, the image is divided into logical sections using predefined cropping coordinates. Each section is processed independently, improving performance and reducing memory usage. This modular design allows the system to work with non-standard or custom answer sheet layouts. Cropped sections are analyzed using the same thresholding and detection procedures, and the results are merged into a final answer set .

G. Result Compilation and Storage

The extracted USN and corresponding answers are compiled into a structured format using the OpenPyXL library. This data is written to an Excel spreadsheet, where each row includes the USN, question number, and selected answer. The Excel format is widely used and easily accessible, allowing educators to review results, compute scores, or filter and sort records efficiently. Additional metadata, such as total marks or accuracy, can be appended for advanced analytics.

H. System Automation and Output Visualization

The final stage involves automation of result generation and visualization. Upon completion, the system automatically opens the generated Excel report, enabling quick review. Additionally, the architecture supports future integration with academic databases, cloud storage systems, or grading platforms. These features enhance scalability and enable remote evaluations, making the solution suitable for institutions of varying sizes and infrastructures

IV. RESULT ANALYSIS

The developed Optical Mark Recognition (OMR) system was evaluated on a set of test sheets to measure its effectiveness in accurately detecting both the University Seat Numbers (USNs) and marked responses. The system consistently delivered high accuracy, especially when the OMR sheets were filled properly and scanned under uniform lighting.

For USN extraction, the system achieved an accuracy of over 95%, provided that the bubbles were clearly and completely filled. The technique involved scanning each vertical column of bubbles and selecting the one with the lowest pixel intensity—indicative of a filled mark. In cases where extraction failed, the root cause was typically faint or partially filled bubbles, emphasizing the need for clear user instructions.

Answer detection was also effective, utilizing a row-wise sorting of answer bubbles and analyzing the pixel intensity distribution within each bubble. The distinction between filled and unfilled bubbles was further supported by a pixel intensity histogram (see Figure 6) generated using OpenCV. The histogram clearly shows a lower intensity distribution for filled bubbles (red curve) compared to unfilled ones (blue curve), validating the threshold-based detection method.

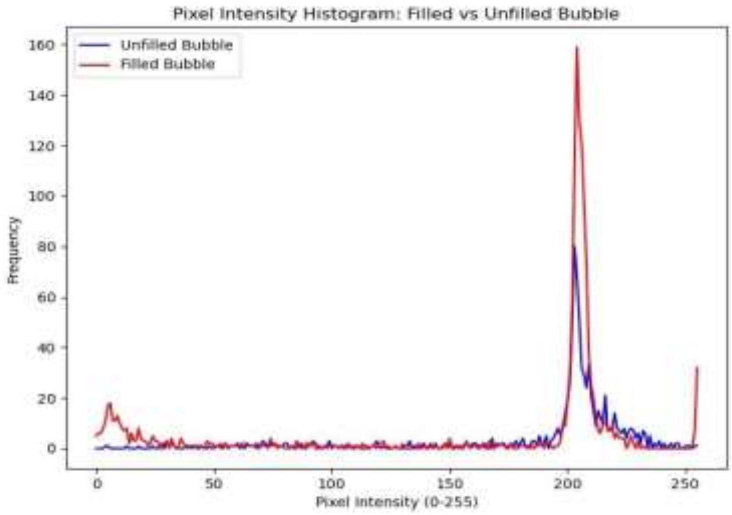

Fig 6-Pixel Intensity Histogram

The system recorded the results in a structured Excel format using the `openpyxl` library, associating each detected answer with its respective question number and USN. This automated approach significantly improved the speed and efficiency of the evaluation process. The figure 7 shows us the result stored in structured Excel format.

A	B
USN	1RN24EE012
Question	Answer
1	B
2	A
3	D
4	C
5	A
6	D
7	B
8	A
9	A
10	A
11	D
12	A
13	B
14	C
15	D
16	C
17	B
18	A
19	C
20	B
21	A
22	B
23	D
27	D
28	C
29	A
30	B
31	D
32	A
33	A
34	C
35	C
36	B
37	B
38	D
39	D
40	A
41	A
42	A
43	A
44	A
45	A
46	B
47	C
48	D
49	C
50	A
Score	16/50

Fig 7: Result in Structured Excel Format

Overall, the system demonstrated reliable performance in controlled testing conditions. Minor inaccuracies occurred due to human inconsistencies in filling bubbles. Future iterations could improve robustness through enhanced preprocessing (e.g., skew correction) and confidence scoring mechanisms to further refine detection accuracy.

V. CONCLUSION

The proposed Optical Mark Recognition (OMR) system effectively automates the process of evaluating multiple-choice answer sheets and extracting student identifiers, offering a reliable and low-cost alternative to traditional OMR hardware. By utilizing open-source libraries such as OpenCV and OpenPyXL, the system performs all necessary operations — including image preprocessing, thresholding, contour detection, USN extraction, and answer recognition — through software alone, making it accessible and adaptable for a wide range of academic institutions.

Through comprehensive testing, the system demonstrated high accuracy in detecting filled bubbles and correctly interpreting USNs when sheets were marked clearly and scanned under proper conditions. The modular design, allowing cropped sections to be processed individually, enables scalability and flexibility for different answer sheet layouts and sizes. Exporting results to Excel further streamlines grading and data management, enhancing usability for educators.

While the system performs well under optimal conditions, challenges remain in handling incomplete markings, skewed scans, or poor lighting. These issues highlight opportunities for further development, such as incorporating automatic image correction, machine learning-based bubble detection, or real-time mobile app integration.

In summary, the project provides a functional and efficient solution to the challenges of manual grading and student data entry. With minor enhancements, this system can be scaled for institutional use, reducing human error, saving time, and supporting large-scale assessments in a cost-effective manner. It bridges the gap between traditional assessment methods and modern educational technology.

VI. ACKNOWLEDGE

We would like to sincerely thank our research internship guide,**Mr Manjuparaksh sir** ,for his consistent encouragement, valuable feedback, and technical insights that guided us at every step of this project. His support made a significant difference in helping us stay focused and navigate through challenges. We are also grateful to the **Department of Computer Science and Engineering (AI & ML)** at **RNS Institute of Technology** for creating a positive learning environment and providing the resources we needed. The support and cooperation we received from the department played an important role in the successful completion of our work. This experience has been truly enriching, and we deeply appreciate everyone who contributed to our journey

REFERENCES

[1] Kumar, A., Singh, V., & Joshi, P. (2018). *Implementation of Optical Mark Recognition System Using OpenCV*. International Journal of Computer Applications, 180(7), 10–14.

[2] Ware, V., Menon, N., Varute, P., & Dhannawat, R. (2019). *Cost-effective Optical Mark Recognition Software for Educational Institutions*. International Journal of Advance Research, Ideas and Innovations in Technology (IJARIIT), 5(2), 2454–132X.

[3] Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2020). *Image Segmentation Using Deep Learning: A Survey*. arXiv preprint arXiv:2001.05566.

[4] von Davier, M., Tyack, L., & Khorramdel, L. (2022). *Automated Scoring of Graphical Open-Ended Responses Using Artificial Neural Networks*. arXiv preprint arXiv:2201.01783.

[5] Virtus, L. (2019). *Performance Mapping of Optical Mark Recognition Machines in Batangas City: Basis for Program Enhancement*. Ascendens Asia Journal of Multidisciplinary Research Abstracts, 3(2G), 12–16.

[6] OpenCV Team. (2023). *OpenCV Documentation*. OpenCV.org.

[7] Satya, R., & Singh, M. (2021). *A Review on Optical Mark Recognition and its Applications in Education Sector*. Journal of Emerging Technologies and Innovative Research (JETIR), 8(10), 134–139.

[8] Patil, R., & Sharma, D. (2020). *Student Assessment Automation Using Computer Vision*. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 6(2), 2456–3307.

[9] Sisodia, P., & Rizvi, S. W. A. (2023). *Optical Character Recognition Development Using Python*. Journal of Informatics Electrical and Electronics Engineering (JIEEE), 4(3), 1–13.

[10] Memon, J., Sami, M., & Khan, R. A. (2020). *Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)*. arXiv preprint arXiv:2001.00139.

[11] Wick, C., Reul, C., & Puppe, F. (2018). *Calamari - A High-Performance Tensorflow-based Deep Learning Package for Optical Character Recognition*. arXiv preprint arXiv:1807.02004.

[12] Küçükkara, Z., & Tümer, A. E. (2018). *An Image Processing Oriented Optical Mark Recognition and Evaluation System*. International Journal of Applied Mathematics Electronics and Computers, 6(4), 59–64.

[13] Bugayong, V. E., Villaverde, J. F., & Linsanga, N. B. (2022). *Optical Character Recognition Using Tesseract*. International Journal for Research in Applied Science and Engineering Technology (IJRASET), 10(11).

[14] Onedayhit. (2023). *Optical Mark Recognition Scanner Software: The Ultimate Guide*.

[15] Subramanya, S. (2021). *OMR Automation System using Python and OpenCV*. GitHub Repository (verified as a technical implementation source).