Optical Mark Recognition

CSE 4128 : Image Processing and Computer Vision Laboratory

Submitted to:

Dr. Sk. Md. Masudul Ahsan
Professor
Department of CSE, KUET

Dipannita Biswas
Lecturer
Department of CSE, KUET

Submitted by:

Md. Zakaria Hossain
Roll: 1907056
Dept. of CSE, KUET

Objectives

The primary objectives of this project are as follows:

- Develop an automated system to process OMR sheets.
- Implement image processing algorithms to detect and extract marked responses from OMR sheets.
- Create a user-friendly interface for uploading, processing, and printing OMR sheet results.
- Ensure high accuracy in identifying and analyzing marked responses.
- Generate detailed reports and analytics based on processed OMR data.

Introduction

Optical Mark Recognition (OMR) is a process of capturing human-marked data from document forms such as surveys and tests. The OMR technology is widely used in various fields including education, surveys, and assessments due to its efficiency and accuracy in data collection. The primary aim of this project is to develop an automated system capable of scanning and processing OMR sheets, thereby reducing the time and effort involved in manual data entry and analysis.

This project leverages image processing techniques to detect and extract marked responses from scanned OMR sheets. By implementing advanced algorithms, the system will accurately identify marks even with varying scan qualities and minor imperfections on the sheets. Additionally, the project includes the development of a user-friendly interface to facilitate the upload, processing, and review of OMR sheet results. Through the implementation of this OMR system, we aim to achieve high accuracy and reliability in data processing, thus enhancing the efficiency of data handling in various applications. The generated reports and analytics from processed OMR data will provide valuable insights and assist in decision-making processes.

Tools Used

- **Python**: A high-level, interpreted programming language known for its ease of use and readability. Python's vast ecosystem of libraries and frameworks significantly accelerates the development of applications, especially in complex data analysis and GUI development.
- **OpenCV (Open Source Computer Vision Library)**: A highly optimized library focused on real-time image processing and computer vision applications. OpenCV is used extensively for its powerful image processing capabilities, such as image transformations, filtering, and object detection.
- **Tkinter**: The standard GUI toolkit for Python. Tkinter allows the easy and effective management of graphical user interface elements, making applications accessible to users.
- **PIL (Python Imaging Library)**: A library that adds image processing capabilities to Python. PIL allows for opening, manipulating, and saving many different image file formats, and it is essential for tasks like image resizing, format conversion, and basic editing.

- **openpyxl**: A library for reading and writing Excel files (xlsx/xlsm/xltx/xltm files). The `openpyxl.workbook` module provides tools to create, modify, and manage Excel workbooks programmatically, making it indispensable for applications that require spreadsheet manipulation and data analysis.

Theory

Optical Mark Recognition (OMR) is a technology used to detect and capture human-marked data from paper forms through image processing and pattern recognition techniques. The process begins by scanning the document and converting it to a digital grayscale image, which is then enhanced and segmented to isolate marked areas such as bubbles or checkboxes. Feature extraction and classification algorithms are used to analyze these areas, determining whether they are marked based on predefined criteria. Thresholding techniques help differentiate marks from the background, ensuring accurate data interpretation. Advanced OMR systems may employ machine learning to improve recognition accuracy and adapt to varying document formats and marking styles.

- **Gaussian filter**: A Gaussian filter is a type of smoothing filter used in image processing to reduce noise and detail by averaging pixel values with a Gaussian distribution.

$$G(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

- **Sobel filter**: The Sobel filter is used in image processing to detect edges by approximating the gradient of the image intensity function in both horizontal and vertical directions.

$$\text{Sobel Filter: Horizontal Gradient } (G_x): \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\text{Vertical Gradient } (G_y): \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- **Convolution**: Convolution is a mathematical operation used to combine two functions to produce a third function that expresses how the shape of one is modified by the other.

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau$$

- **Gradient magnitude and direction**: Gradient magnitude and direction are used to determine the strength and orientation of edges in an image by analyzing the changes in intensity at each pixel.

$$\text{Gradient Magnitude} = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

$$\text{Gradient Direction} = \tan^{-1}\left(\frac{\frac{\partial I}{\partial y}}{\frac{\partial I}{\partial x}}\right)$$

- **Non-maximum suppression**: Non-maximum suppression is a technique used in edge detection to thin out the edges by retaining only the local maxima in the gradient direction, ensuring that only the most significant edges are preserved.

$$\text{Non-maximum Suppression(NMS)} =$$
$$\begin{cases} \text{If } G(x,y) \text{ is a local maximum in the gradient direction, retain } G(x,y). \\ \text{Otherwise, set } G(x,y) \text{ to zero.} \end{cases}$$

- **Hysteresis thresholding**: Hysteresis thresholding is a technique in edge detection that uses two threshold values to identify strong and weak edges, ensuring that weak edges connected to strong edges are retained, while others are discarded.

$$\text{Hysteresis Thresholding} =$$
$$\begin{cases} \text{Strong Edge: } G(x,y) \geq \text{High Threshold} \\ \text{Weak Edge: Low Threshold} \leq G(x,y) < \text{High Threshold and connected to a Strong Edge} \\ \text{Suppressed: } G(x,y) < \text{Low Threshold} \end{cases}$$

- **Contour detection**: Contour detection using Breadth-First Search (BFS) involves traversing connected components of an image to detect and outline contours by exploring pixel neighborhoods.

  BFS Algorithm:

  1. Start at an initial boundary pixel.
  2. Use a queue to explore all connected pixels.
  3. Add neighboring pixels to the queue if they belong to the contour.
  4. Continue until all connected pixels are processed.

- **Area**: The area of a rectangular contour is calculated by multiplying its width and height, providing the total surface area covered by the rectangle.

$$\text{Area} = w \times h$$

- **Thresholding**: Thresholding is a technique used in image processing to segment an image by converting it to a binary image based on a specific threshold value that separates the foreground from the background.

$$\text{Binary Image}(x,y) = \begin{cases} 1 & \text{if } I(x,y) \geq T \\ 0 & \text{if } I(x,y) < T \end{cases}$$

Methodologies

1. **Convert to Grayscale Image**: This process involves transforming a colored image into a grayscale image where each pixel represents a shade of gray, simplifying the image and making it easier to process for further image analysis tasks.

2. **Smoothing the Image**: Smoothing techniques, such as Gaussian blur, are applied to reduce noise and minor variations in the image. This step is essential for improving the accuracy of subsequent image processing operations by minimizing irrelevant details.

3. **Edge Detection**: This technique identifies the boundaries within an image by detecting significant changes in intensity. Edge detection algorithms, like Canny edge detector, highlight the edges, making it easier to distinguish different objects within the image.

4. **Contour Detection**: Contour detection is used to identify and extract the outlines of objects within an image. Contours represent the shape of an object and are useful for object recognition and measurement tasks.

5. **Finding Biggest Contour**: Among the detected contours, the largest one is often of particular interest. This step involves identifying the contour with the maximum area, which typically corresponds to the primary object or region of interest in the image.

6. **Extracting Answer Area**: After finding the biggest contour, the corresponding region is extracted. This region often contains the answer section in OMR sheets, which needs to be further processed to evaluate the responses.

7. **Thresholding**: Thresholding converts a grayscale image into a binary image, where pixels are either black or white. This method is used to enhance contrast between the background and the objects of interest, making it easier to analyze marked answers on OMR sheets.

8. **Evaluating Answers**: This process involves analyzing the thresholded answer area to determine which options are marked. The evaluation criteria are based on predefined patterns or templates to accurately identify the selected answers.

9. **Showing Answers**: After evaluating the answers, the results are displayed to the user. This step provides a visual representation of the detected responses, allowing for quick verification and review.

10. **Finding Second Biggest Contour**: In some cases, the second largest contour is of interest, such as a grading area or a secondary region in the image. This step involves identifying and extracting this contour for further analysis.

11. **Extracting Grading Area**: Similar to extracting the answer area, this step focuses on isolating the region corresponding to grading or scoring. This area is processed to determine the overall score or grade based on the evaluated answers.

12. **Showing Final Grading**: The final grading, based on the extracted grading area and evaluated answers, is displayed to the user. This provides a clear and concise summary of the performance or results.

13. **Save into Excel File**: The evaluated answers and final grading are saved into an Excel file. This step ensures that the results are documented and can be easily accessed, analyzed, and shared for further processing or record-keeping.

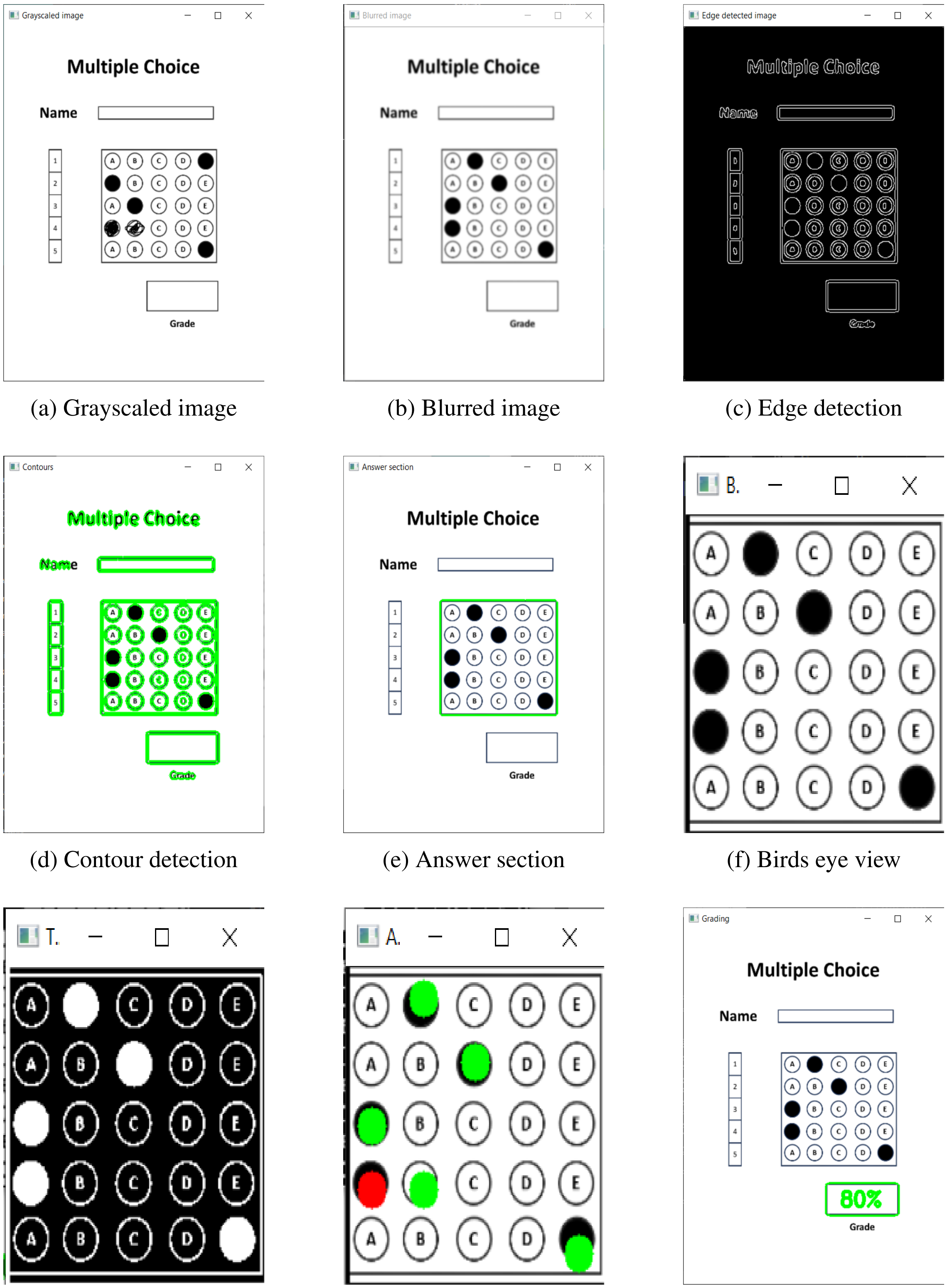
The overall processing is represented by the figure 1 below

Discussion

Optical Mark Recognition (OMR) projects are integral to automating the extraction and processing of data from scanned documents, revolutionizing fields such as education, surveys, and assessments. By employing advanced image processing techniques, OMR systems accurately interpret marked responses, enhancing efficiency and reducing human error in data collection and analysis. The primary objectives of such projects include automating data extraction from scanned forms, ensuring high accuracy in interpreting responses, and integrating seamlessly with existing database systems for streamlined data management. Methodologies typically involve image preprocessing to enhance scan quality, segmentation to isolate response areas, and classification algorithms to identify and evaluate marked choices. Applications span across educational institutions for grading exams and surveys, market research for gathering consumer feedback, and healthcare for managing patient information efficiently. Challenges include managing varying document layouts and ensuring scalability to handle large volumes of data effectively. Overall, OMR projects exemplify the convergence of technology and practical application, offering robust solutions for modern data processing needs across diverse industries.

Limitations

- **Document Layout Dependency**: OMR systems are sensitive to variations in document layouts and designs, which can affect the accuracy of data extraction and interpretation.
- **Handling of Handwritten Responses**: While adept at processing machine-marked responses, OMR systems may struggle with handwritten inputs, leading to potential errors in data capture.
- **Scalability Challenges**: Scaling OMR systems to handle large volumes of data efficiently can pose logistical and computational challenges, impacting processing speed and performance.
- **Complexity in Integration**: Integrating OMR technology with existing information systems or workflows may require significant customization and compatibility checks, potentially increasing implementation time and costs.
- **Accuracy Issues with Poor-quality Scans**: OMR systems rely heavily on clear and high-resolution scans for accurate data extraction. Poor-quality scans or document defects can compromise accuracy and reliability.
- **Limited Flexibility in Question Types**: OMR systems are primarily designed for multiple-choice and checkbox-based questions. They may not effectively handle complex question types or forms requiring free-form responses.



(a) Grayscaled image (b) Blurred image (c) Edge detection

(d) Contour detection (e) Answer section (f) Birds eye view

(g) Thresholding (h) Answers (i) Grading

Figure 1: Detailed image representation of the overall processing.

Conclusion

In conclusion, Optical Mark Recognition (OMR) technology offers significant advancements in automating data extraction and processing from physical documents. By leveraging sophisticated image processing techniques and machine learning algorithms, OMR systems streamline the capture and interpretation of marked responses, enhancing efficiency and reducing errors in various applications such as education, surveys, and assessments. Despite facing challenges like document layout dependency and scalability issues, OMR remains a powerful tool for modernizing data collection processes across diverse industries. Future advancements in image processing capabilities and integration with artificial intelligence promise to further improve the accuracy and flexibility of OMR systems, making them indispensable for organizations seeking efficient and reliable data management solutions.