	Pimpri Chinchwad Education Trust's **Pimpri Chinchwad College of Engineering** **Formative Assessment 1**	
Department: Computer Engineering **Academic Year:** 2024 - 2025		
Year and Div: Final Year B. Tech C	**Maximum Marks:**	
Subject: Computer Vision	**Date: 17/09/2024**	
Name: Mahima Nair	**PRN: 121B1B161**	
Title:Project Report on AutoGrade: Optical Mark Recognition (OMR) MCQ Automated Grading System		

AutoGrade: Optical Mark Recognition (OMR) MCQ Automated Grading System

---

1. Introduction

Optical Mark Recognition (OMR) is a process used to detect marked answers from forms or exam papers where users fill in their answers. This project titled, “AutoGrade: OMR MCQ Automated Grading System," automates the grading process for multiple-choice question (MCQ) exams. The system takes an image of the OMR answer sheet as input and provides a grade based on the answers marked by the candidate, comparing it to a pre-defined answer key.

This system is particularly useful for reducing human effort, increasing accuracy, and speeding up the grading process, especially for large-scale exams where manual grading is inefficient and prone to errors.

---

2. Objectives

The primary objectives of this project are as follows:

1. To develop an automated system for grading OMR-based MCQ exams using computer vision techniques.
2. To implement contour and corner detection to identify the answer sheet and individual answer boxes.
3. To extract the marked answers from the answer sheet.
4. To compare the extracted answers with the provided answer key to calculate the final score.
5. To display the results, including the final score and individual answer correctness, on the processed image.

---

3. System Requirements

- A computer with a minimum of 4 GB RAM and a 64-bit processor.
- **Python 3.x**: The primary programming language used.
- **OpenCV**: A library for computer vision tasks.

- **NumPy**: For matrix operations and numerical computations.

---

4. Methodology

The methodology followed in this project includes several key stages, starting from image acquisition to grading.

4.1 Preprocessing

1. **Image Acquisition**:
    - A set of pre-scanned images are loaded from a dataset repository found on Kaggle.
2. **Image Resizing**:
    - The captured image is resized to a fixed dimension (700 x 700 pixels) to ensure uniformity in processing.
3. **Grayscale Conversion**:
    - The image is converted to grayscale to simplify further processing.
4. **Noise Removal**:
    - A Gaussian Blur is applied to reduce noise and make the edges of the answer boxes more detectable.
5. **Edge Detection**:
    - Canny edge detection is applied to highlight the contours of the shapes (answer boxes, boundaries) in the image.

4.2 Contour Detection

Contours are the outlines of shapes or objects in an image. We use the cv2.findContours() method to detect all contours present in the image. Contours are filtered to extract only rectangular shapes, which are potential candidates for the answer sheet and grading area.

4.3 Corner Detection

The biggest rectangle (which corresponds to the answer sheet) is identified by extracting its corner points. The second largest rectangle is used to display the grading box. Corner detection is critical for performing perspective transformation and ensuring the answer sheet is properly aligned before extracting marked answers.

4.4 Perspective Warping

Once the corners are detected, perspective warping is applied to align the answer sheet, removing any distortion due to skew or rotation in the captured image. This transforms the sheet into a flat, properly oriented image where each box is clearly defined.

4.5 Feature Extraction and Answer Detection

1. **Answer Box Splitting**:
    - The answer sheet is divided into multiple smaller boxes using the splitBoxes() function. Each box corresponds to a choice in the MCQ.
2. **Marked Answer Detection**:
    - The number of non-zero pixels in each answer box is counted. The answer with the highest count of non-zero pixels is identified as the marked answer for that question.
3. **Thresholding**:
    - Binary thresholding is applied to enhance the visibility of marked areas.

4.6 Grading Logic

1. **User's Answer Extraction**:
    - For each question, the index of the box with the highest pixel count is recorded as the selected answer by the user.
2. **Answer Comparison**:
    - The detected answers are compared with a predefined correct answer key. A list is created where each correct answer is marked as 1 and incorrect as 0.
3. **Score Calculation**:
    - The final score is computed as the percentage of correct answers.
4. **Grade Display**:
    - The final score is displayed both numerically (e.g., "80%") and graphically on the processed image.

---

5. Results and Discussions

The system successfully detected the answers marked on the OMR sheet and compared them with the predefined correct answers. The final score was displayed both on the console and directly on the answer sheet image.

Example:

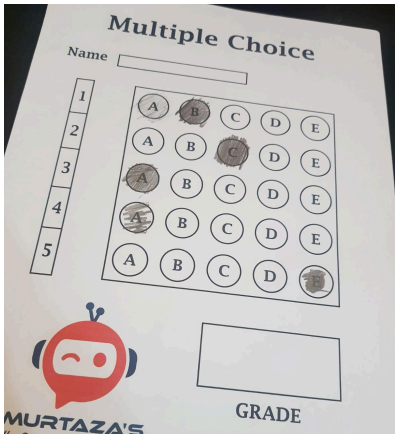


Fig 1: Input image 1

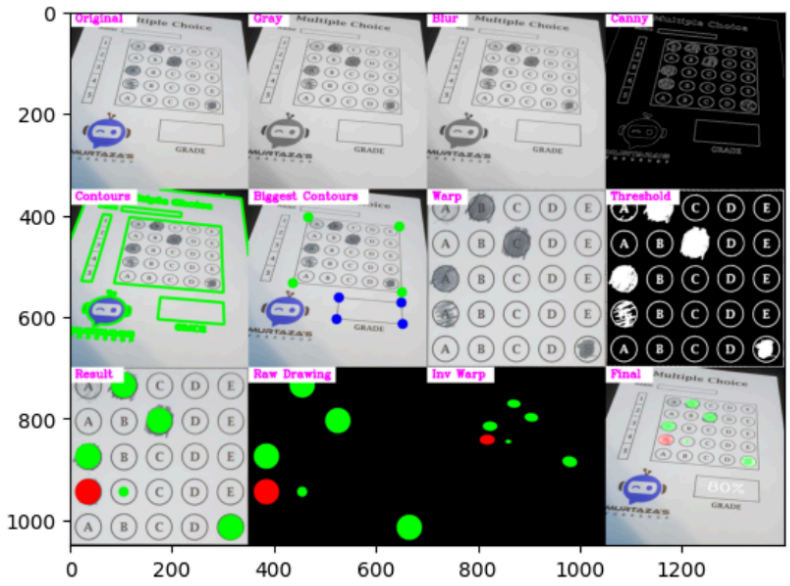


Fig 2: Processed image 1

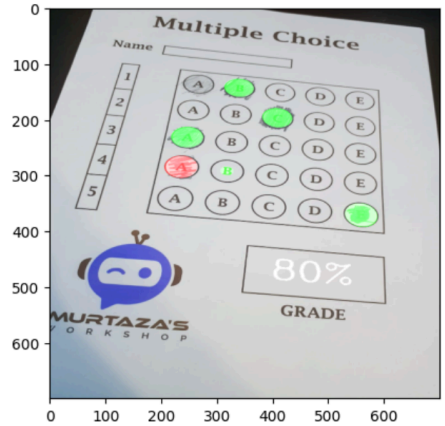
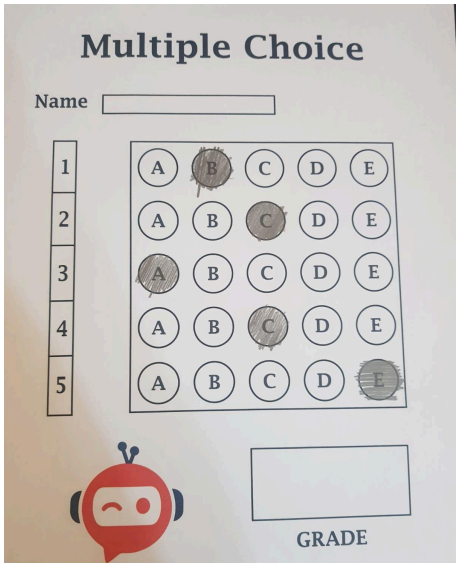


Fig 3: Final result of image 1

- **Correct Answers**: 4 out of 5
- **Score**: 80%
- **Final Image Display**: The system displayed the correct and incorrect answers on the warped image, with a clear indication of the final score.



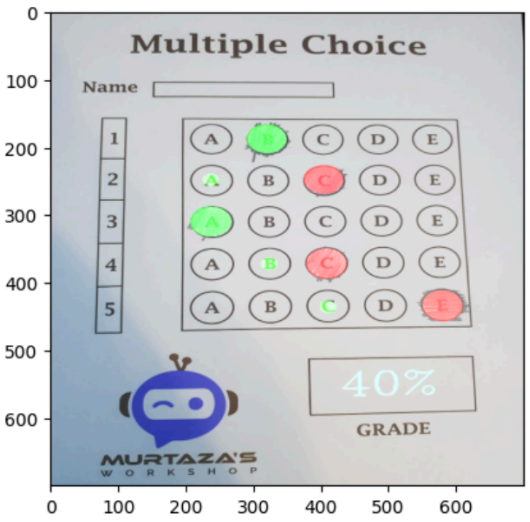


Fig 4: Input image 2

Fig 5: Final Result of image 2

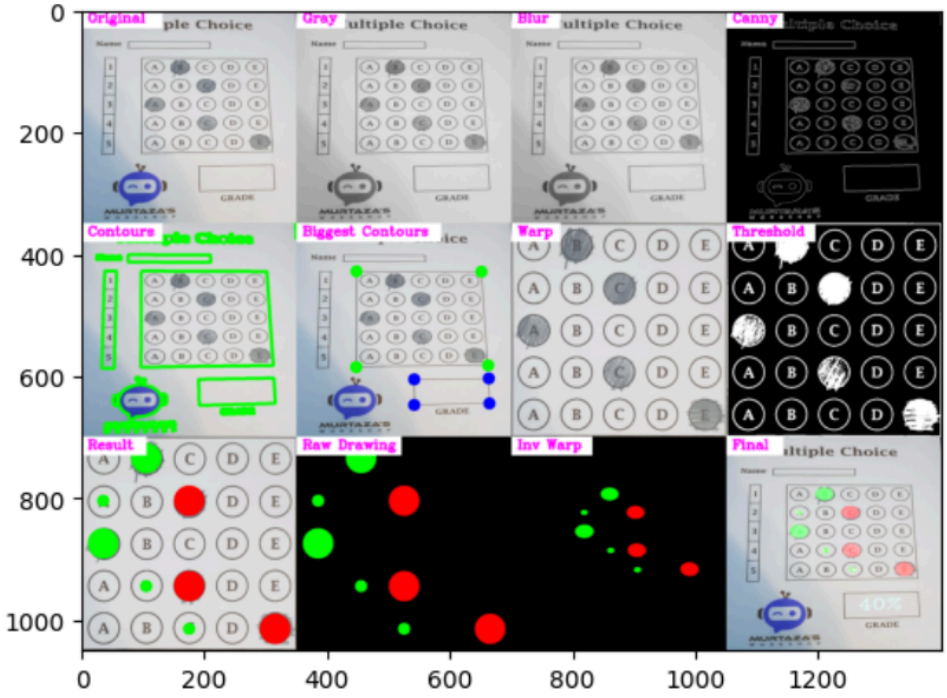


Fig 6: Processed image 2

- **Correct Answers**: 2 out of 5
- **Score**: 40%
- **Final Image Display**: The system displayed the correct and incorrect answers on the warped image, with a clear indication of the final score.

.

---

6. Conclusion

This project demonstrates the successful development of an automated OMR grading system using computer vision techniques. The system effectively detects the answers marked by the user, compares them with the correct answer key, and displays the final score. This solution is fast, efficient, and reduces the scope of human errors in the grading process.

---

7. Future Scope

The project can be further extended by incorporating the following features:

- **Support for Multiple Answer Sheets**: Adding the ability to process multiple answer sheets in batch mode.
- **Improved Skew Correction**: Implementing advanced skew correction techniques for images taken at extreme angles.
- **Handwritten Answer Recognition**: Integrating Optical Character Recognition (OCR) for handling questions that involve handwritten responses.
- **Mobile Integration**: Developing a mobile application for easy scanning and grading using a smartphone camera.
