Automated OMR and Answer sheet grading using Open-CV

Si. No.	Name Of The Student	Branch	Email Id
1.	R.Arun	Cyber Security	artge565@gmail.Com
2.	M.Nithiyan	Cyber Security	mnithiyan555@gmail.Com
3.	B.Nandhakumar	Cyber Security	b.nandha2810@gmail.Com
4.	S.Sri Saran	Cyber Security	srisaransureshkumar@gmail.com

Guide Name: R.Rajalakshmi,
Designation : Assistant Professor ,
Mobile No : 7010654982
Email Id : rajalakshmisk@gmail.com

1. Problem Statement:

Develop an automated system using OpenCV to scan, process, and grade Optical Mark Recognition (OMR) answer sheets by detecting marked answers and comparing them to a predefined answer key.

5.Project Summary:

The project "Automated OMR and Answer Sheet Grading using OpenCV" focuses on developing a computer vision-based system to automate the process of evaluating Optical Mark Recognition (OMR) sheets and grading answer sheets.

6. Proposed Solution With Methodology:

Developing a document scanner for automated Optical Mark Recognition (OMR) and answer sheet grading using OpenCV involves leveraging image processing techniques to extract and analyze data from scanned documents. In automated Optical Mark Recognition (OMR) and answer sheet grading using OpenCV, sorting contours is a critical step to ensure the correct processing order. The contours, typically representing answer bubbles, must be sorted either row-wise or column-wise based on the design of the answer sheet. To achieve this, contours are first extracted using methods like cv2.findContours().To achieve a top-down view of answer sheets for automated Optical Mark Recognition (OMR) and grading using OpenCV, perspective transformation is a key technique. It ensures the answer sheet is properly aligned for accurate processing. The process begins with preprocessing, where the image is converted to grayscale and denoised using GaussianBlur or similar filters. Next, edge detection (e.g., using Canny Edge Detector) is performed to highlight the contours.

7. Plane Of Action Of Implementation:

- Detect the OMR sheet
- Apply perspective transform to get the top-down view of the sheet
- Extract out all the bubbles in the sheet

- Sort them in rows
- Determine the answer bubble for each row
- Match with the correct answer
- Do this for all questions (all rows)

8.Expected Outcome And Result:

1. The app assumes that the OMR document we are scanning is the main focus of the image.
2. All 4 edges of the OMR document are visible in the image.
3. The largest rectangle available in the image will be the OMR document.
4. Stored the correct answer keys in a dict in python.
5. Used Canny edge detection for detecting the edges in the document and Gussian blur for reducing high frequency noise.
6. OpenCV has the way to get the top-down view of the image. Used that mehodology to get the top-down view.
7. Used Otsu’s thresholding method for thresholding.
8. Determined the bubbles using the aspect ratio of approx one (1) for it's bounding rectangle.
9. Used bitwise operations and masking to find the filled in bubble using the amount of shaded pixels in the bubble