INDIAN INSTITUTE
OF
TECHNOLOGY
INDORE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER VISION PROJECT
CS 419/619

Automatic Grading of Scanned Multiple Choice Answer Sheets

Authors:

Drishti Sharma ms2204101002
Ashutosh Dhamaniya ms2204101005

Under the guidance of :

Dr. Surya Prakash
Associate Professor
Department of Computer Science and Engineering
IIT Indore

December 2022

Contents

1 Introduction 4

2 Pre-requisite 4

2.1 OMR Sheet . . . 4

2.2 Contour . . . 4

2.2.1 Working of CHAIN_APPROX_SIMPLE method . . . 5

2.2.2 Working of **cv2.threshold()** method . . . 5

3 Methodology 5

3.1 Techniques Used . . . 5

4 Dataset 6

4.1 Samples of Dataset . . . 6

5 Results 7

6 Conclusion 7

References 8

List of Figures

1 Example of OMR sheet. 4
2 Flow of the project 6
3 Image samples from the Dataset 6

1 Introduction

This project aims to detect the marked answer from the filled OMR, i.e., Optical Mark Recognition Sheets. In this project, we will apply some techniques first to detect the options encircled in the OMR sheet. After detecting encircled choices, we will match them with the correct answers and calculate the score.

2 Pre-requisite

2.1 OMR Sheet

The OMR stands for Optical Mark Recognition. These are designed to simply the typical pattern of multiple choice examination, form-filling methods, and surveys.

The basic idea is to automate the system of evaluation of answer sheets,form etc.

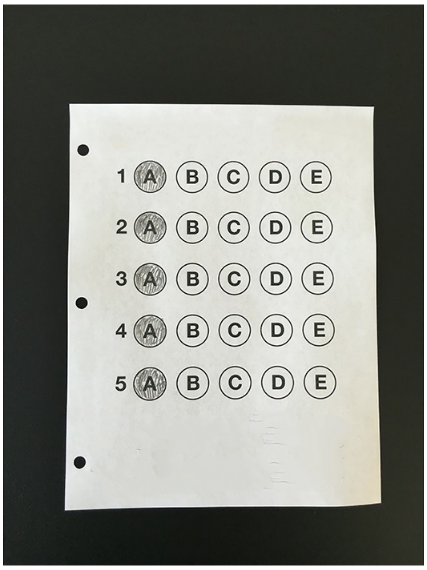

Figure 1: Example of OMR sheet.

2.2 Contour

Contours can be defined as the curves joining all the continuous points along the boundary. With the help of contours, we can identify the borders of object and locate them on images easily[1].

In computer vision, we use the function $cv.findContours()$ to get the desired parameters from an image. The function works well on binary image.

2.2.1 Working of CHAIN_APPROX_SIMPLE method

- Initially loop over region we get from $find\_Contours()$.
- CHAIN_APPROX_SIMPLE[2] will compress the horizontal, vertical and diagonal aspect of the contour region.
- We will get the endpoints necessary to represent the region that results in saving memory.

2.2.2 Working of cv2.threshold() method

- This method requires four arguments sampleImage, minValue, maxValue and thresholdingTechnique.
- With human intervention, we specify a threshold T.
- All pixel intensities below T are set to 255. And all pixel intensities greater than T are set to 0.
- If applying cv2.THRESH_BINARY_INV, the inverse assignment of intensity will happen.

3 Methodology

To achieve the objective of this project, we are using techniques like Gaussian Blurring, Canny Edge Detection etc. The following are described below:

3.1 Techniques Used

1. **Gaussian Blurring:** This technique is used to remove the unnecessary noise and artifacts from the image. Here we are using blurring to avoid any unwanted error in detecting the optical mark[3].
2. **Canny Edge Detection:** It is used to detect the edges of object present in the image. Here the output of the canny edge detection is the boundary of rectangle and the circles[4].
3. **Four_point Method:** This method allows us to detect objects with the help of four parametric values or points. The points used in this project are $min(x), max(x), min(y), max(y)$, where x, and y are the coordinates detected via contours.

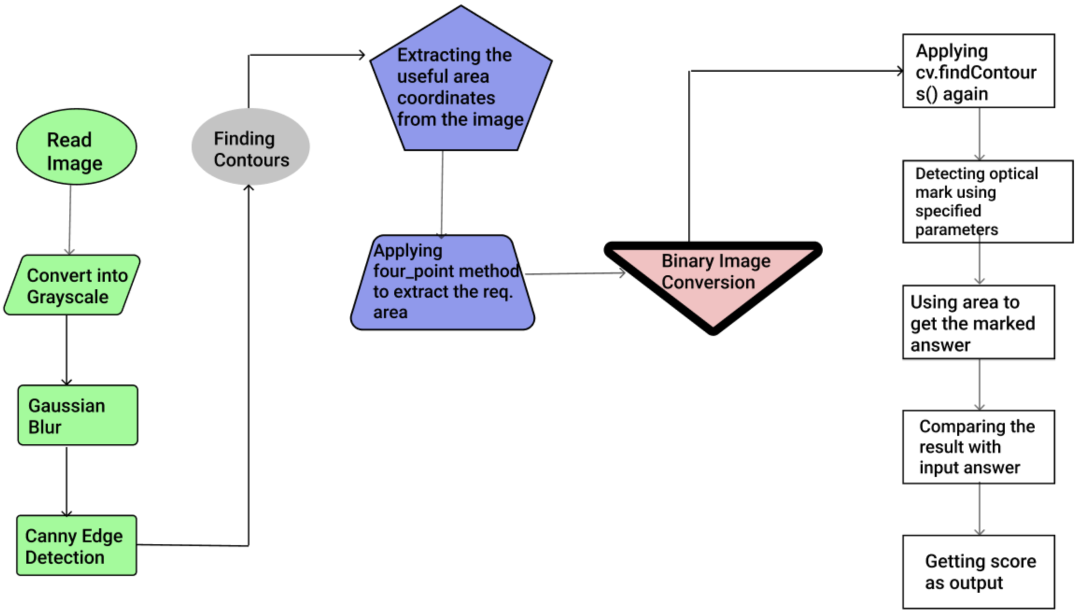
Read Image
Convert into Grayscale
Gaussian Blur
Canny Edge Detection
Finding Contours
Extracting the useful area coordinates from the image
Applying four_point method to extract the req. area
Binary Image Conversion
Applying cv.findContours() again
Detecting optical mark using specified parameters
Using area to get the marked answer
Comparing the result with input answer
Getting score as output

Figure 2: Flow of the project

4 Dataset

For this project, we have collected the images of marked OMR sheets by group of students. For testing purpose we have used six different images following different marking.

4.1 Samples of Dataset

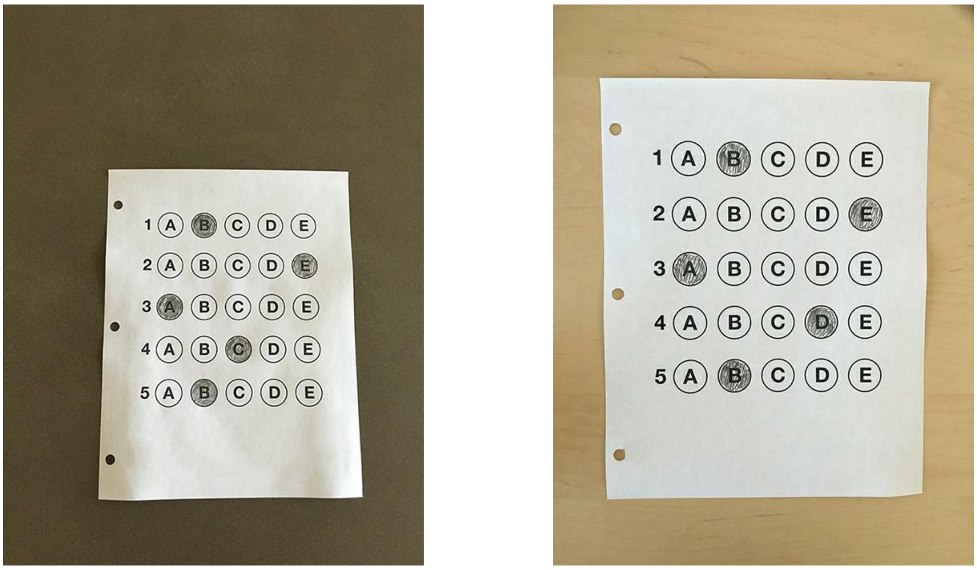

Figure 3: Image samples from the Dataset

5 Results

The score can be calculated by multiplying the ratio of the number of correct marked answers and the total number of questions by 100.

$$\text{Score} = (\tfrac{\text{number of correct answers marked}}{\text{total number of questions}}) * 100$$

6 Conclusion

The Automatic Grading Of Scanned Documents can perform the checking with an accuracy of approx 80 percent. The canny edge detection and contours make it possible to avoid noise and help in detecting the optical mark row-wise.
The model can be improved by allowing it to take the input of correct answers via scanning of answer key document and further testing it with OMR containing columns using distance as a factor.

References

[1] S. Suzuki *et al.*, “Topological structural analysis of digitized binary images by border following,” *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.

[2] N. Deepika and V. S. Variyar, “Obstacle classification and detection for vision based navigation for autonomous driving,” in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI).* IEEE, 2017, pp. 2092–2097.

[3] E. S. Gedraite and M. Hadad, “Investigation on the effect of a gaussian blur in image filtering and segmentation,” in *Proceedings ELMAR-2011.* IEEE, 2011.

[4] M. A. Ansari, D. Kurchaniya, and M. Dixit, “A comprehensive analysis of image edge detection techniques,” *International Journal of Multimedia and Ubiquitous Engineering*, 2017.