Journal of Computer Science & Engineering
GSJ: Volume 11, Issue 4, April 2023, Online: ISSN 2320-9186
www.globalscientificjournal.com

A SCORING SYSTEM BASED ON AUTOMATIC ANSWER DETERMINATION AND IMAGE RECOGNITION FOR DIFFERENT EXAMINATION TECHNIQUES

Mohamed Elmi Ahmed
Currently pursuing a master's degree program in Computer Science at Gazi University
Ankara, Türkiye
mohamedelmi2019@gmail.com

Asst. Prof. Yilmaz Atay
Faculty member in the Department of Computer Engineering at Gazi University
Ankara, Türkiye
yilmazatay@gazi.edu.tr

ABSTRACT

The purpose of this study is to create a system that can evaluate and score multiple-choice questions (MCQs) and true/false (T/F) answer sheets using a real-time webcam. These types of questions are still widely used today by institutions, organizations, and educational institutions to evaluate students' performance. Large institutions use optical mark recognition (OMR) software, which is expensive and requires a special scanner and customized OMR sheets. Therefore, small schools could not pay for this expensive software to perform assessments. Thus, it is proposed a real-time webcam system that evaluates and scores test papers automati- cally, which can utilize anyone and save more time spend on manual scoring. First, it is given a brief overview of related studies. Second, it is discussed the proposed system in detail and with important steps. Third, it is explained how the image processing algorithms and methods are implemented in OpenCV. Finally, general evaluations about the study are made and a general per- spective about the future is mentioned.

Keywords

Automatic Answer Determination, Computer Vision, Examination Techniques, Grading, Image Recognition, Optical Mark Recognition, Scoring System.

Introduction

In today's technology, many applications in our lives have to do with computerized image processing and computer-aided recognition. At present, the correction of examination papers depends mainly on manual correction by teachers. Conducting multiple-choice question (MCQ) examinations is now one of the methods used most regularly to assess the performance of students all around the world. In education, the exam is important. For teachers, correcting students' exams means providing feedback on their teaching. For students, correcting their exams on time is also important for their studies. However, it is usually time-consuming for teachers to grade exams carefully and accurately. Therefore, the development of an automatic exam grading method has attracted much attention.

Millions of students take standardized exams every year, in which they must answer numerous questions by covering up rounds on OMR sheets. Existing solutions for scoring these OMR sheets are costly and need the purchase of a specific scanner. As a result, lecturers, small organizations, and institutes cannot afford to adopt this software. They depend on hand scoring of answer papers. It takes an average of 5 minutes to score a student's answers on a standardized test. The main purpose of our project is to automate the correction and grading of answer sheets in real-time, saving money as well as time and manpower, we combine the (MCQs) and (T/F) in one answer sheet, using a webcam and a computer. So there is no additional cost associated with this com- puter-based solution. It covers the majority of the requirements of the institutions and organizations while also resolving the aforementioned issue.

The following is how the rest of the article is structured: Part II provides a brief overview of relevant studies. Part III goes through the proposed project. Section IV details the proposed system's algorithms and approaches. Finally, Part V summarizes the work and an outlook for the future.

Literature Review

The OMR scan evaluation technique was first used in the 1950s when a machine with a set of scanning brushes found graphite particles on paper [1]. Tien Dzung Nguyen [2] introduces another camera-based, dependable, and effective multiple-choice test-scoring technique. They showed the use of less expensive non-transoptic response sheet paper to achieve precision. As a result, they assigned the answer fields using skew fitting and normalization measures. The amount of dark pixels is used to make the choice.

AL -Marakeby [3] proposes a low-cost and fast solution for an optical marker recognition system that operates on a multi-core processor system. A digital camera captures the response board. The sheet's edges are found first, followed by the speech bubbles. Rotation, skew, perspective projection, and specularity are all solved by using bold lines and edges as markings to detect the orientation of the bubbles. The adaptive binarization technique is used to solve the illumination variation problem. An edge-tracking algorithm is a fast tool for detecting lines and edges.

Azman Talib et al. [4] took a different technique with involved matching. It was divided into two parts: detection and training. During the training phase, the web camera collects an image of the OMR sheet, which is subsequently processed using smoothing filtering algorithms. Following that, a rectangular ROI (Region of Interest) is manually chosen around a collection of response blocks, utilizing the question number as the template. The picture of the template is now placed on the OMR sheet and compared in the recognition phase. Lastly, the intensity values of the candidate image and the template are compared to assess whether ornot the candidate's answer fits the template.

Sanguansat [5] proposed automatic data entry using Optical Mark Recognition (OMR). They use normal image scanners in OMR software applications. In this work, they proposed a solution for creating questionnaires consisting of closed questions only. It also includes a proper report of the obtained output in spreadsheet software.

Proposed Work

The proposed system will be discussed in detail and basic information about its operation will be shown. The proposed system is easy, effective, and inexpensive compared to existing systems. We have designed the answer sheet in MS word, it is easy for teachers to design the exam format using MS word, but the algorithm can work with any other design tool. The format of the assessment papers, the main part of the assessment sheet contains seven alternatives of (MCQs) and (T/F) together such as A, B, C, D, E, T, and F, and there are corresponding boxes where students can mark their correct answers. The evaluation sheet comprises a set of 20 questions, with each question offering seven alternative choices. Additionally, there is a designated space provided for the display of the grades shown in Fig 3.

Capturing images: After starting the webcam, turn it on. Place the answer sheet in front of the webcam, examine it in real-time, and display the scores to decrease the waiting times. Remember that the answer sheet should be set appropriately in front of the webcam, have sufficient lighting, and not cast shadows on the sheet during this assignment, since this might interfere with the filtering process.

Content Filtration: After we receive the answer sheet in real-time, the next thing is to extract the portion of the image that we want, as we do not require the entire image. On the picture, we create rectangles and extract the image within the rectangles. The retrieved picture is then utilized as input for image processing.

Image processing: After providing the picture as input to this component of the algorithm, the following step is to do some type of preprocessing on it before applying the image processing method to it. The next stage is to give a real answer to our algorithm, which will then evaluate if the response matches the true response and, if so, will show how many answers are right and how many are wrong, as well as the percentage gained. If the answer is right, it is marked with a green circle; if it is wrong, it is marked with a red circle.

Saving the result: At this point, after checking the answer sheet, the teacher or user will press the letter S to save the final answer sheet with the corresponding scores and store it in a folder.

Algorithms and Techniques

The previous section explained that our main algorithm is utilized to extract the answer location from an image of the answer sheet. This section aims to provide an in-depth description of our algorithm. The Canny Edge Detection Algorithm is a precise method for detecting the edges of images. It was originally developed by John F. Canny in 1986 [6, 7] and is one of the most widely-used image processing devices. The Canny Edge Detector is often employed to identify the edges of images and is considered to be a standard edge-detecting method. Canny viewed the edge-detection problem as an optimization problem in signal processing, and he developed it to optimize an objective function.

Phase 1: The first step is to use the webcam to capture an image of the answer sheet. You require the VideoCapture function for this function of OpenCV, which directly checks the sheet in real time and displays the scores, which reduces the implementation time.

Phase 2: The entire image is not required. Because just the answer sheet part is required, the image must be cropped. We will need OpenCV and key events for this.

Phase 3: Once the picture has been filtered, it should be converted to grayscale format so that the user's markings can be seen. Subsequently, the picture is heavily blurred to eliminate high-frequency noise, which results in picture distortion.

Phase 4: Once converting to grayscale format, the following step is to extract the page from the provided image. This is a multi-step process that creates a picture with all edges (width and high).

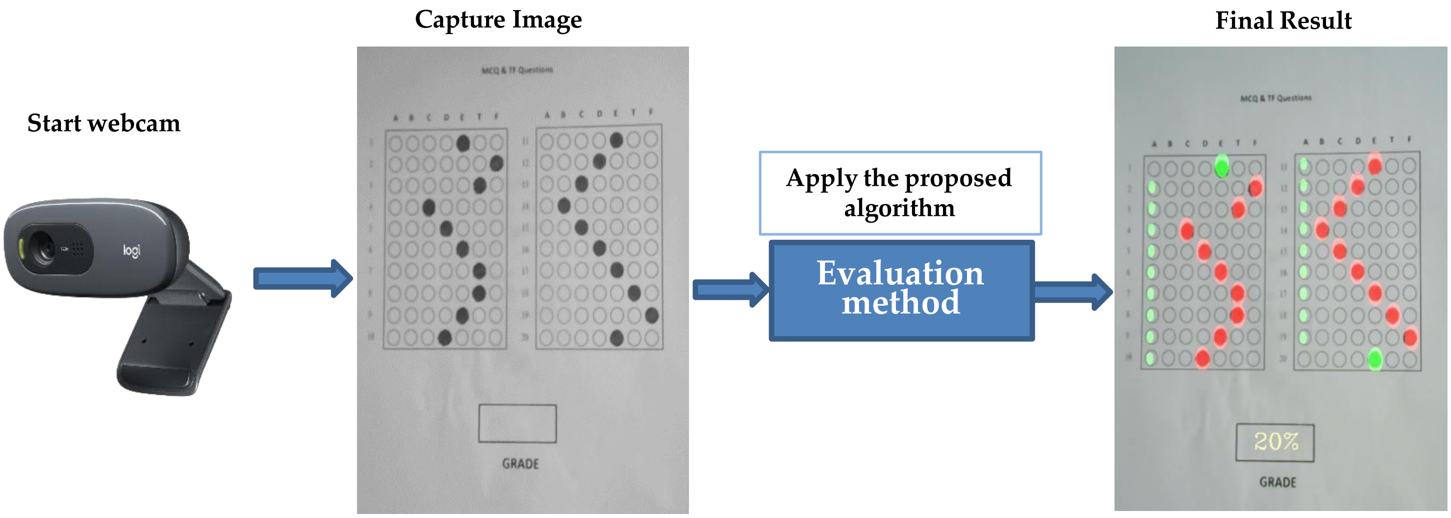

Fig 1. Webcam Fig 2. Answer sheet Fig 3. Final Result

Phase 5: Once we have obtained the shape, use the transformation to examine the previous image from top to bottom and from a bird's eye perspective. This is carried out so that we can get four edge points on our exam sheet using the open cv library "cv.find- Contours()" to find these four points. Before proceeding with this step, Canny Edge Detection must be used. This stage involves locating a white structure against a black background, which is only achievable with the preceding phase. Use the argument "cv. CHAIN _APPROX_ SIMPLE" to retrieve simple vertices rather than a continuous line creating a rectangle (pager structure). Only corner points will be extracted, and duplicate points will be removed.

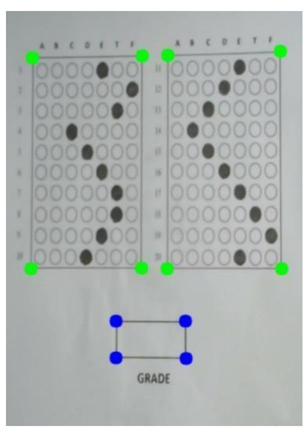

Fig 4. Finding all contours

Phase 6: Once we have the contours, arrange them so that the larger contours are at the top of the array and the smaller contours are at the bottom. The underlying rationale is that the four retrieved edge contours are greater than the contour of the item in the picture. Here the picture of the answer paper should be the main focus within the two big rectangles and the small one in the picture.

Phase 7: We use the method "four-point transform()" to get a birds-eye perspective from above. This function is used to perform geometric transformations on a picture to extract only the side image.

Phase 8: Later, we will perform binarization on the image, which means we will convert the image to black and white up to 256 levels of gray. This is used as a pre-processing step before applying OCR or OMR logic. This is possible using the Otsu transformation. A black-and-white picture is formed, as seen below. This is accomplished by using threshold() from open cv and supplying the otsu parameter.

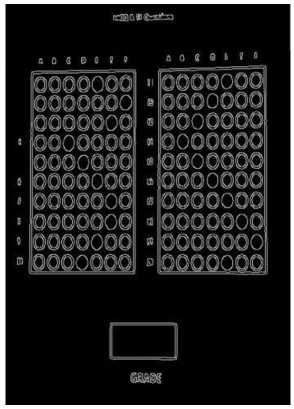

Fig 5. Grayscale image

Phase 9: The following phase is to extract the marked options from the response paper. We apply transformation and binarization once more to do this. There are several methods for finding marked in a picture. This is kept in an array, where each index repre- sents one of the circle's seven options/contours, and each row is stored in each index of the array, resulting in a nested array.

Phase 10: It is runned a function to check whether the marked answers are correct by comparing them to an array of correct answers. We repeat through the contour array previously created and compare each option to our original answer array to determine its correctness. However, we still need to identify which option is marked within the circle contour. We use thresh() parameter of OpenCV to distinguish a marked white circle from the other black circles with white borders. Starting with the first index in the contour array contour [0] [0], we check if it is a white circle. If not, we move to the next option until we reach the end of the array. We repeat this process for all indexes in the circle contour array-contour [1][0]. If the current contour matches the threshold contour of the circle, the counter of the correct array is incremented. The correct answer is then marked with a big green circle, while the wrong answer is marked with a big red circle. In cases where the user marked the wrong option, the system marked a big red and also point to the correct as a small green. This can be achieved using the drawContour() function of OpenCV.

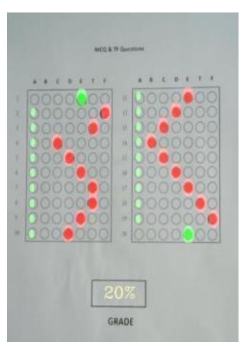

Fig 6. The obtained final score

Phase 11: When you approach the last contour of the last row, use the fundamental formula to compute and show the percentages.

Conclusion

This proposed system for automatic grading of (MCQs) and true/false (T/F) tests provides high accuracy and execution speed. Manual grading of a sheet is taking more time, is tedious, and is difficult. The suggested approach saves the user a signific ant amount of time, is inexpensive and is simple to implement. The goal of this project was to create an OMR grading system in OpenCV using the Canny Edge Detection Algorithm. The experimental input paper was printed on an A4 sheet. Current systems require an OMR scanner, a regular scanner, or a mobile app to scan the filled sheet and then use it as input for the system. However, our suggested system is a real-time scoring system, which means that instead of capturing screenshots, when the sheet is held in front of a webcam, then immediately checks the sheet in real-time and shows the results, minimizing implementation time and removing the human interface. Then, the final result image with the corrected answer and grade was saved in a folder after the user pressed the letter S. In future work, the proposed system should improve the algorithm that can adjust to bad lighti ng conditions, simplify the setup process, improve accessibility, and integrate the system into an LMS for better usability.

References

[1] Rakesh S, Kailash Aftal et al "Cost Effective Optical Mark Reader," International Journal of Computer Science and Artificial Intelligence, vol. 3, no. 2, pp. 44-49, 2013.
[2] Tien Dzung Nguyen, Quyet Hoang Manh, Phuong Bui Minh, Long Nguyen Thanh, Thang Manh Hoang, "Efficient and Reliable Camera based MultipleChoice Test Grading System," IEEE International Conference on Advanced Technologies for Communications (ATC), 2011.
[3] Mahmoud Afifi, Khaled F. Hussain, "The Achievement of Higher Flexibility in Multiple Choicebased Tests Using Image Classification Techniques", arXiv.org,Cornell University Library , 2017
[4] A. AL-Marakeby "Multi-Core Processors for Camera based OMR", International Journal of Computer Applications (0975 – 8887) Volume 68– No.13, 2013.
[5] A. Rajasekaran and Senthilkumar. P, "Image Denoising Using Median Filter with Edge Detection Using Canny Operator," International Journal of Science and Research,vol. 3, no. 2, 2014
[6] Parinya Sanguansat, "Robust and Low-cost Optical Mark Recognition for Automated Data Entry," IEEE Transactions on, vol. 9, no. 1, pp. 62-66, 2015.
[7] G. Bradski and A. Kaehler, Learning OpenCV. USA: O'Reilly Media, Inc., 2008.