

# Android Based Automated Scoring of Multiple-Choice Test

Hendra Tjahyadi, Yoga G. Budijono, Samuel Lukas, and Dion Krisnadi

**Abstract**—In this paper design and implementation of an Android based automated scoring of multiple-choice test is reported. This application is designed as a more affordable alternative for more popular solutions which usually composed of a software and Optical Mark Recognition Scanner.

Two main parts are involved in the design process, namely the answer sheets design and the image processing procedure. Moreover, there are 4 main stages in the image processing procedure: (i) input taking stage, (ii) early processing stage, (iii) identifying stage, and (iv) evaluating stage. After those stages are implemented, tests are conducted to find the best distance and angle of the smart phone to the answer sheet in taking the answer sheet's images and to evaluate the accuracy in identifying right answers and gives valid scoring.

The experiment results show that the best distance and angle with 95% of identification success rate is between 20 to 50 centimetres with 45 to 90 degrees angle, and when the answer sheet's image is correctly identified, the identification of valid scoring success rate is 100%.

**Index Terms**—Multiple-choice test, Android, image processing, automated scoring.

## I. INTRODUCTION

Written test is an important tool for competence measurement and evaluation either in academic or industrial context. Multiple-choice usually uses as part of the written test where the people who sit on the test are asked to select a correct answer out of the choices in a list. One of the advantages of multiple-choice is in time efficiency for checking process and the possibility to be automated especially for a large number of examinees.

Several solutions for automated scoring of multiple-choice test are available commercially. Most of them are composed of a software and Optical Mark Recognition Scanner [1]-[3]. The devices are relatively expensive so that the accessibility is very limited.

On the other hand the availability of smartphones with integrated camera opens the possibility to create a cheaper solution for automated scoring test. Those two reasons

Manuscript received August 3, 2017; revised October 11, 2017. This work is funded by Indonesian Ministry of Research, Technology and Higher Education, Directorate General of Higher Education, Kopertis Wilayah III, No. 0419/K3/KM/ 2016 dated on 24 Mei 2017 and 188/LPPM-UPH/VI/2017.

Hendra Tjahyadi is with Department of Computer System, Faculty of Computer Science, Universitas Pelita Harapan, Indonesia (e-mail: hendra.tjahyadi@uph.edu).

Yoga G. Budijono is with Bank Central Asia, Indonesia (e-mail: yoga.gb@live.com).

Samuel Lukas and Dion Krisnadi are with Department of Information Technology, Faculty of Computer Science, Universitas Pelita Harapan, Indonesia (e-mail: samuel.lukas@uph.edu, dion.krisnadi@uph.edu).

become the motivation to implement an Android based automated scoring of multiple-choice test that is more accessible especially for education institutions.

Two main parts are involved for the implementation of the proposed application: (i) answer sheet design and (ii) image processing procedure. The purpose of the image processing procedure is to be able to identify right answers in the answer sheet and give valid scoring. The answer sheet is designed in such a way to make sure the efficiency in the image processing procedure and the accuracy of the identification of the right answers.

The discussion in this paper is composed of several sections as follows: in Section II the design methodology which consists of the answer sheet design and all the steps in the image processing procedure are discussed, followed by the implementation and experiments results discussion in Section III, and in Section IV this paper is concluded.

## II. DESIGN METHODOLOGY

### A. Answer Sheet Design

The answer sheet is designed based on the availability of suitable functions in OpenCV library to process images and also for the possibility to be machine-processed. By doing that it is expected that the identification process from input images will be simple with accurate results and suitable for further automation development.

Two main OpenCV functions that will be employed are contour function and Hough circle transformation function [4]. Contour function will be used in combination with perspective transformation to determine whether an image is skew or square, and to change a skew image into a square image to fit in the available square frame. Hough circle transformation function is used to find the position of darkened circles that corresponding to answers in the answer sheets.

The answer sheet designed is shown in Fig. 1. There will be no marker (A), (B) and (C) in the real answer sheet. Those markers are shown in the figure only for the explanation purpose. Marker (A) is to denote the rectangular frame in the answer sheet. The frame is bold printed to make contour function easier to find the four corners as input for the perspective transformation. Marker (B) is referring to three black circles on the left side and three black circles at the bottom of the answer sheet. In this paper, those black circles will be referred as key circles. Based on the key circles, the X and Y positions and the distance between circles in the answer sheet, referred as answer circles in this paper, can be calculated through Hough circle transformation. This is possible because all the circles in the answer sheet are designed to have uniform

distances, and black circles are used to indicate some initial and final positions of the answer sheet. Marker (C) denotes the small black squares at the four corners of answer sheet to limit the working area of image processing. The answer sheet is designed for forty questions with each question has five possible answers from A to E. Two other fields in the answer sheet are ExCode and MCode. ExCode is to indicate examinees' number, and MCode for question sets' number.

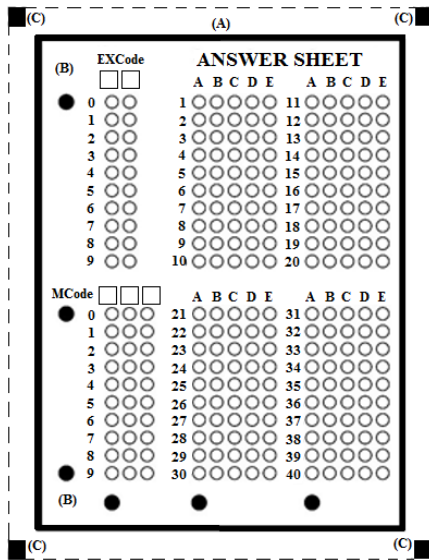


Fig. 1. Answer sheet designed.

### B. Image Processing Procedure

Based on the designed of answer sheet, the image processing procedure is determined. Four stages are involved in the image processing procedure: (i) input taking stage, (ii) early processing stage, (iii) identification stage, and (iv) evaluation stage.

#### Input taking stage

Input images that come either directly from camera or from internal memory will be troughed to a cropping process. This process is to determine a specific part that wants to be identified. After that process the image will be displayed through imageView in Android application for the next process.

#### Early processing stage

This process is to prepare the input image from the input taking stage to be ready for identification stage. There are four steps in this stage as shown in Fig. 2. Each step has different function as follows:

- 1) Format conversion to the required format. In this step two OpenCV functions are used namely "bitmapToMat()" and "CvtColor( RGB2GRAY)". Function "bitmapToMat()" is applied to convert the image from camera or internal memory that in bit data type into matrix data type that is supported by OpenCV [5]. The resulting matrix is composed of scalar green, red and blue colors. Some functions in OpenCV do not work with matrix that composed of color scalars, therefore "CvtColor( RGB2GRAY)" function is applied to convert that kind of matrix into grayscale matrix.
- 2) Image conversion into black and white image. The resulting grayscale matrix is then converted into a black and white image. Three functions are applied to have a

black and white image: (i) "GaussianBlur" to smooth the image and reduce noise in the image, (ii) "adaptiveThreshold()" to obtain black and white images, and (iii) "bitwise\_not" to produce negative images. This last function is required for lessen calculation because in the identification stage there will be many identification functions process the "not black" portion of the image.

- 3) Answer sheet detection. In this step three functions are used. "findContours()" to find forms in the image, "contourArea()" to calculate the area of the found contour, and "approxPolyDP()" to find contour with define number of corners. Using those functions answer sheet will be detected by searching a form with four corners with the largest area.
- 4) Application of transformation. After the answer sheet is detected, based on the four corners of the image, form transformation is applied. This transformation is applied to obtain a rectangular image from a skew image cause by different angle and distances of taking the image. To do that two functions, "getPerspectiveTransform()" and "warpPerspective" are employed. "getPerspectiveTransform()" is used to obtain a matrix that contains the comparison between the four corner of detected answer sheet and the four corners of the desired form. That matrix will be the input to "warpPerspective" function. Using those two functions the image of the answer sheet that may be skewed in form become rectangular similar to the designed of the answer sheet.

The output of the early processing stage is a rectangular image with a few noises as shown in Fig. 3 that will be more suitable for the next stage, the identification stage.

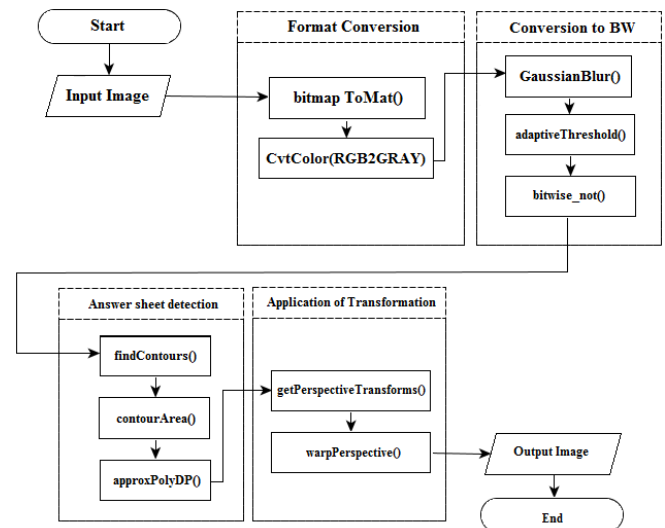


Fig. 2. Steps in the early processing stage.

#### Identification stage

In this stage the information about the positions of all answer circles and which circles are darkened are extracted. The first step to extract that information is by identifying the positions of the key circles which are at the left side and at the bottom of the answer sheet using "submat" function followed by HoughCircles transformation. The results of those two process are X positions from the key circles at the bottom and Y positions from the key circles at the left side

of the answer sheet. Based on the X and Y positions of the key circles the positions of the answer circles can be found using simple calculations.

Centre points of the two top key circles at the left side represent initial points of answer circles at Y axis and the center points of all the key circles at the bottom represent initial points of answer circles at X axis. The distances between two adjacent answer circles can be found as the distance between two bottom key circles at the left side divided by nine.

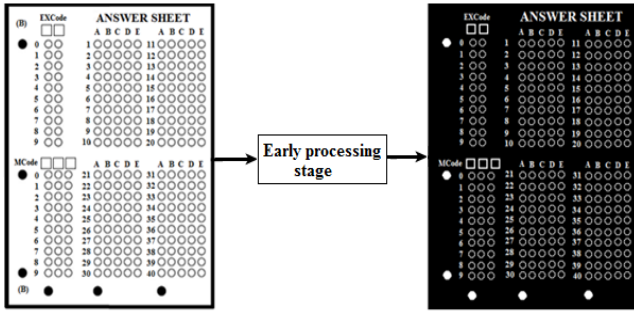


Fig. 3. Input and output of the early processing stage.

After all the positions of the answer circles are defined the next step is to determine whether the circles are darkened. The determination is by counting the number of non-zero pixel in a particular circle compared to the area of that particular circle using countNonZero function. If the result is more than 70% then the circle is consider to be darkened, and the position of the circle will be stored in an array. There will be 6 arrays, namely ExCode array, Mcode array, answers 1 – 10 array, answers 11 – 20 array, answers 21 – 30 array, and answers 31 – 40 array.

**Evaluation stage**

In this stage the arrays as the outputs from the identification stage will be evaluated to determine the category and validity of the identification. The evaluation process is shown in Fig. 4.

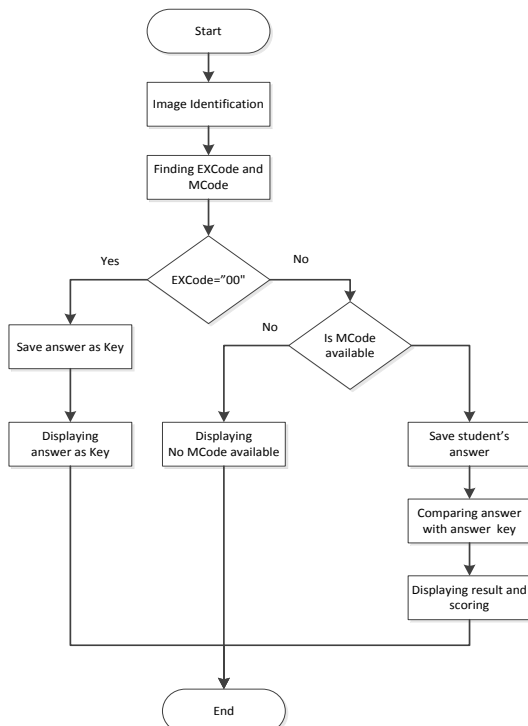


Fig. 4. Evaluation process.

Initially, application will check the ExCode in the answer sheet. If the ExCode is “00” it is mean that the answer sheet is belong to an examiner. In this case the application will check the MCode and then stores the answers for that particular MCode as answer key. If the ExCode is not “00” it is mean that the answer sheet is belong to examinees or students then the application will check the MCode and will seek answer key for that particular MCode. If the application finds the answer key then application will compare the answers with the answer key. As a result of comparison the answers and the score will be displayed, and all the results will be stored in a text file. If the application cannot find the answer key then the process will be stopped, and a message “No MCode available” will be displayed. In this application all the answer key will be lost if the program is closed or the system is restarted.

**III. IMPLEMENTATION AND EXPERIMENTS**

The proposed automated scoring test is implemented on Android application using Java with image processing library ofOpenCvdirectlybuild on the smartphone using Android Studio. The smartphone specification is shown in Table I.

TABLE I: SMARTPHONE SPECIFICATION

Android Version	4.1.2
CPU	Quad-core 1.4 GHz Cortex-A9
RAM	1 GB
Camera Resolution	8 MP, f/2.6
Display Resolution	720 × 1280 (306 ppi)

To test the performances of the proposed application two series of experiments are undertaken. The first series of experiments is to find out the best distance and the best angle of the smartphone to the answer sheet in the image taking process to have success identifications. The second series of experiments is to measure the success rate of the system to recognize the right answers compared to the answer key and gives valid scoring with the condition that the answer sheet is successfully identified.

**First series of experiments**

In this series of experiments images from a sample of answer sheet is taken with four different angles (20o, 45o, 75o, and 90o) and five different distances (20 cm, 30 cm, 40 cm, 50 cm, and 60 cm) to find the best results of identification. The image taking process of the answer sheet is illustrated as in Fig. 5.

For each distance and angle ten times image taking are performed. An answer sheet is consider to be successfully identified if the application is able to get answer sheet contour, circleskey and positions of answer circles accurately. The success rate of the identification for different angles and distances are shown in Table II.

From Table II, it can be seen that the best angle to have success identification is 45o and the best distances are in the range of 20 cm to 40 cm. In the range of angle between 45o to 90o and distances between 20 cm to 50 cm the average of success rate is 95%.

**Second series of experiments**

In this series of experiments, initially 2 answer sheets with ExCode “00”, one with MCode “409” and the other “137” are taken and stored as answer keys. Then 20 others

answer sheets, 10 corresponding with MCode “409” and the others 10 corresponding with MCode “137” are taken and compared for the evaluation. All the images are taken with 45° of angle and 20 cm of distance to make sure that all the images are successfully identified by the application. The results of the evaluation for answer sheets with MCode ”409” and with MCode “137” are shown in Table III and Table IV, respectively.

From Table III and Table IV, it can be seen that there are no false detections for all samples and the application give valid scoring results with the success rate of 100%.

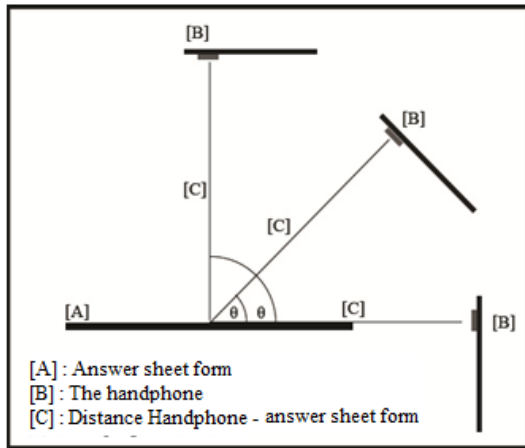


Fig. 5. Illustration of the image taking process.

TABLE II: SUCCESS RATE OF IDENTIFICATION FOR DIFFERENT ANGLES AND DISTANCES

Distance (cm)	Angle (degrees)			
	20	45	70	90
20	0%	100%	90%	100%
30	40%	100%	90%	100%
40	50%	100%	90%	90%
50	30%	80%	100%	100%
60	30%	50%	40%	70%

TABLE III: EVALUATION RESULTS OF 10 ANSWER SHEETS WITH MCODE “409”

EX Code	84	77	48	21	12	10	55	97	38	73
Correct	30	8	20	32	35	40	24	0	9	15
Error (%)	0	0	0	0	0	0	0	0	0	0

TABLE IV: EVALUATION RESULTS OF 10 ANSWER SHEETS WITH MCODE “137”

EX Code	22	66	17	10	77	88	15	50	99	11
Correct	40	19	36	1	34	37	33	29	19	20
Error (%)	0	0	0	0	0	0	0	0	0	0

#### IV. CONCLUSION AND FUTURE WORKS

An Android based application for automated scoring of multiple-choice test are designed and implemented. With a proposed answer sheet designed and image processing procedure the application is able to give valid scoring with 100% success rate. The angle and the distance of the smartphone to the answer sheets are some factors for the

success rate of the application. Although very high success rate is obtained further developments are required. A mechanism and machinery for automatic placing and image taking of answer sheet that will be able to guarantee the success rate of identification process is worth to explore. An algorithm that enable examinees to cancel one answer and choose another answer as final answer will be one of the future works.

#### REFERENCES

- [1] A. F. Mollah, N. Majumder, S. Basu, and M. Nasipuri, “Design of an optical character recognition system for camera-based handheld devices,” *International Journal of Computer Science*, vol. 8, no. 1, 2011.
- [2] H. Deng, F. Wang, and B. Liang, “A low-cost omr solution for educational applications,” in *Proc. ISPA '08. International Symposium on Parallel and Distributed Processing with Applications*, Dec. 2008, pp. 967-970.
- [3] K. Atal et al., “Cost effective optical mark reader,” *International Journal of Computer Science & Artificial Intelligence*, vol. 3, no. 2, 2013.
- [4] G. B. Gracia, et al., *Learning Image Processing with OpenCV*, Birmingham: Packt Publishing, 2015.
- [5] A. Kehler and G. Bradski, *Learning OpenCV*, Sebastopol: O’Reilly Media, 2008.



**Hendra Tjahyadi** was born in Bandung, Indonesia in 1969. He received bachelor degree from Electrical Engineering Department, Maranatha Christian University, Indonesia, and master degree from physics engineering, Institute Technology Bandung, Indonesia in 1993 and 1996, respectively. He was graduated as Ph.D in control engineering from Flinders University, Australia in 2006. He joined University Pelita Harapan in 2013. His research interests are adaptive control and signal processing. He is a senior lecturer in Faculty of Computer Science, Universitas Pelita Harapan, Karawaci, Indonesia.



**Yoga G. Budijanto** received Computer Science degree, majoring in information technology from Universitas Pelita Harapan, Indonesia in 2016. He is currently working as IT support in Bank Central Asia, Indonesia.



**Samuel Lukas** received his bachelor degree in electrical engineering from Trisakti University, Indonesia, Jakarta in 1985. He got the master degree in computer science from University of Tasmania, Tasmania, Australia in 1992 and doctorate degree in science education in 2005 from Curtin University, Australia. He is a senior lecturer in Faculty of Computer Science, Universitas Pelita Harapan, Karawaci, Indonesia. He wrote many papers and two books chapter. The latest book chapter was title Real-world Application of genetic Algorithms chapter 15. He is interested in artificial intelligence with research area in computer vision. Dr. Lukas is a member of Indonesian Higher Education Association for Information and Computer Studies.



**Dion Krisnadi** received computer science degree and mathematics degree in 2012 from Universitas Pelita Harapan, Indonesia. He got his master degree in information technology, 2014 from University of Indonesia, Indonesia. Since 2014, he worked for Universitas Pelita Harapan as a researcher in computer science studies His research interest is artificial intelligence. D. Krisnadi is a member of Indonesian Higher Education Association for Information and Computer Studies.