IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 6 August 2024, accepted 22 August 2024, date of publication 29 August 2024, date of current version 10 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3451972

RESEARCH ARTICLE

An Enhanced Fault Tolerance Algorithm for Optical Mark Recognition Using Smartphone Cameras

QAMAR HAFEEZ[1], WAQAR ASLAM[1], ROMANA AZIZ[2], AND GHADAH ALDEHIM[2]

[1]Department of Information Security, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan

[2]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

Corresponding author: Waqar Aslam (waqar.aslam@iub.edu.pk)

This work was supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R387), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

ABSTRACT Optical Mark Recognition (OMR) systems have been studied since the 1970s. Due to its simplicity of use and efficiency in bulk operations, OMR technology has been gaining popularity over time. They are used as an automated data input technique for surveys and multiple-choice question papers in educational institutions for automatic evaluation and grading of student inputs. The requirement of the conventional OMR systems comprises specialized OMR machines or optical scanners with automatic document feeding capability. These machines and scanners are fixed-location devices and cannot be moved easily. Their energy requirements are high, while they also require human efforts to operate. These machines are expensive and, hence pose budget constraints for small educational institutions. Due to being mechanical, their maintenance and operating cost is high. To overcome these limitations, alternate devices are smartphone cameras, which though handy adversely lack the capability of scanning documents in a controlled environment. An uncontrolled environment leads to inputs that existing OMR algorithms do not recognize at large, while the accuracy rate and precision stay low to an undesirable extent. Due to this shortcoming, the usage of smartphone cameras is still not feasible. In this experimental study, we have proposed an OMR algorithm specifically for inputs taken from smartphones equipped with decent cameras and running Android or iOS operating systems. Thus effectively, we have ported the OMR technology to smartphones, offering more flexibility, easiness, and mobility of its usage in daily life. The key issue that transpired in our experiments is the bad illumination in different lighting conditions. Our results are very promising and comparable to those obtained from the usage of optical scanners.

INDEX TERMS Optical mark recognition, information extraction, grade marking system, smartphone camera.

I. INTRODUCTION

Optical Mark Recognition (OMR) is an automatic "mark sensing" computing technique for reading input from users. It facilitates conducting surveys, interviews, examination systems, and questionnaires. Users first mark answers on a paper, which are optically sensed by counting the number of black pixels of the expected input area. The input area for each answer is usually a circle. Due to automation, the system is efficient and effective, thus enabling fast progress of an organization. OMR systems leverage speed and accuracy for relevant tasks to a greater extent than the staff performing the same tasks [1]. In the implementation of the OMR system, generally, the first step is to create a template of an OMR document. Multiple copies of the OMR document are paper-printed. The OMR documents are distributed among users participating in the examination or questionnaire. Users fill in the circles of their choices on the OMR document. The

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Sharif.

OMR documents are then collected from users and sent for scanning, which converts physical OMR documents in paper form to digital images in the form of computer memory. Once the image is in the computer memory, image processing algorithms can be applied to extract the information of interest. The main task of the image processing algorithm in this system is to clean noise, remove unwanted artifacts and extract information from the OMR document.The accuracy of the image processing algorithms used in an OMR system is very important. A small mishandling by the image processing algorithms can produce false results that an organization does not accept. Hence, a fault tolerant OMR system is required so that it can be trusted and adapted widely by organizations.

The decade of the 1950s was the first era when scientists tried to implement mark recognition technology. In those days, graphite pencils were used by the users for filling the circles. Specialized sensing brushes were used to sense the graphite material on the paper. The document is placed below those sensing brushes so that each brush touches the circle in a row. In this way, the only brushes touching the graphite material sent signals to the system. As time passed, improvement in the form of optical mark recognition technology was designed in the 1970s. In this era, the technology was shifted from sensing brushes to optical sensors. In this mechanism, the light was projected on the OMR documents [2]. In the OMR systems, when light was projected on circles, only partial light was allowed to pass through the filled circle whereas light with greater intensity passed from the empty circles. Optical sensors were placed at the bottom of the circles. The positions of sensors are aligned so that one sensor exists directly below each circle. These sensors detected light intensity threshold values, which were turned on them. If a sensor received less intensity of light than its threshold value, it sent a signal of a filled circle back to the system [3]. There were some limitations to this type of system. One of the major limitations was the need to change the hardware whenever there was a change in the template/layout of the OMR document. One may need to add or reduce the number of optical sensors used in the OMR document, such as the position of the sensors. Hence changing the design of the OMR document resulted in costly changes. Due to the limited ability of mass-scale production, the cost of those OMR systems was high and proportional to the cost of expensive sensors. Similarly, the running and maintenance costs are also very high accordingly.

After optical sensor-based systems, optical mark recognition scanners were designed. They could convert the physical OMR documents to digital OMR documents, which were fed to image processing algorithms for extracting required information [4]. This technology was cost-effective and hence got adopted widely worldwide as is the case today. These OMR systems overcome the limitations of the older technologies, especially due to hardware limitations. Changing the design of the OMR document was a time-consuming task. Now the system has evolved enough so that OMR documents can be changed without any change of hardware or software. Also, the running and maintenance costs of the OMR systems are much less.

In the current OMR systems, the digital OMR documents are passed through multiple passes to remove unwanted artifacts and noise. Next, they are analyzed by the OMR algorithms to track user choices. OMR documents can be created using any designing software such as Pages (Mac OS X), Microsoft Word (Windows) LibreOffice (Linux), or any other word processing software. OMR documents can also be created by using image manipulation programs like GNU Gimp, Photoshop by Adobe, etc. This method of OMR system implementation is dynamic and the process of extracting information does not need any change in hardware or software. It is flexible enough to accommodate any design of an OMR document. However, optical scanners introduce challenges that have to be addressed. For instance, the introduction of skew errors in the OMR document by optical scanners is inherent to the OMR systems. Even printers can introduce skew angles. Similarly, another challenge arises due to the translation (skidding) of the printer roller [5]. Both scanners and printers introduce unwanted artifacts in the OMR documents, thus leading to inaccurate results. Such issues need primary consideration by the image processing algorithms.

A digital image consists of pixels arranged on the X and Y axes. The algorithm used to detect OMR information by processing the pixels on the 2D plane is also called the pixel projection method, introduced in 1999. This method was used to identify the position of the circle on OMR documents [4]. The earlier version of the pixel projection method was not dynamically handled by the implementation software (discussed in Subsection II-A). This software needs to be based on algorithms that can dynamically adjust to the changing OMR documents so that the majority of OMR designs can be processed [6]. Improvement in the skew detection and correction mechanisms was achieved using horizontal and vertical projection [7]. In another variant, instead of the fiducial markers, bold vertical line borders, called ''pattern finders'', were added to the OMR documents [7], [8], [9], [10], [11], [12]. Another substitute for the fiducial markers was the addition of pattern boxes on three sides of the OMR documents [9]. Yet another method was to use flag points on horizontal and vertical axes on the OMR document at the time of OMR document design. Those flag points were placed at regular distances on the OMR document and used to calculate the reference point just like the function of fiducial markers. Similarly, four filled squares were used [10]. Extraction of the OMR information using multimode operations was also used in OMR systems [4]. In this method, there were two modes added to process the OMR document. Learning mode was the first mode, which was used to learn where the horizontal and the vertical lines exist on the OMR document and calculate their crossing points so that areas within the Region of Interest (ROI) could be determined. Then the second mode called the operation mode, calculated the number of black pixels in circles.

In a novel approach, the image is extracted from the OMR document using a base image, which contains all the correct answers [13]. Both the base image and the marked OMR document are scanned by the optical scanner. Both images are aligned at the same angle using image processing algorithms. The images are then cropped and multiplied, resulting in a new image with only correctly marked circles. Multiplication operation is used to skip the incorrect answers. By using multi-core processing systems to parallelize tasks, the performance of the OMR system was boosted [14]. Preprocessing tasks on the OMR system included circle detection, border detection, and feature extraction. One approach was to use the thick border on the OMR document. The purpose of adding a thick border was to detect the skew angle on the OMR document [15]. After each hundred pixels, vertical scan lines are dropped until the thick border is found. The algorithm looks for the first block of black pixels and when found, it stops and records the length of the scan line dropped from the top. In this way, the algorithm can calculates the skew angle and corrects this skew angle. A framework written in Python was created for the OMR system [16]. The framework was called the Gamera Framework. In another approach, during the process of OMR, flag points are added to the OMR document at the OMR design time. Improvements in the alignment of the OMR document, scale correction, and skew correction of the OMR document have also been made [17]. In their method, the regions of interest on the OMR document are searched using a neighborhood rule and pattern recognition technique. To improve the accuracy of the scanner-based OMR system, training and classification methods have been used [18]. One open source Java-based project called "FormScanner" is also available [19].

Another approach in the OMR system is to calculate regions of interest using OpenCV [20]. Calculated regions of interest are then cropped. In the end, the cropped regions are passed to the algorithm that is responsible for extracting information from the OMR document. In the United States, balloting was done with the help of OMR systems during their elections [21]. Another two-phase system was introduced, based on the training phase and the recognition phase. These two phases were used with Modified Multi Connect Architecture [22].

To increase the processing speed of the OMR system, Field Programmable Gate Array (FPGA) was introduced in the OMR system [23]. The maximum speed achieved was up to 50,000 OMR documents per hour. They have used a high-resolution CCD linear image sensor to achieve high speed. To fix the skew angle error introduced in the handwritten documents, the Radon Transform method was used [24]. Another study using the comparison of OMR documents was done at the stage of prepossessing of OMR documents, while in the next step, AND operation was performed between the original OMR document and the captured OMR document to identify the marked positions [25].

With the advancements of PC webcams, experiments are done for their use in OMR systems. Using the Hough transform algorithm, initial work is done [26]. The Octave script was used for the creation of a framework for the OMR system [27]. In the field of OMR, the Canny edge detection method was introduced. The purpose of canny edge detection was to detect lines as well as the position of circles [28]. The usage of artificial vision improved the accuracy by correcting the deformed circles [29]. The Skewness of the paper was addressed using marks on four corners of the OMR document. A study was done in the field of OMR using a webcam [30]. In this approach, a shape-based matching training phase was added to process of the OMR system. The issues coming at the stage of scanning of OMR documents are discussed [31]. Another OMR system using a conveyor belt was implemented using an ordinary webcam and a microcontroller [32]. A system based on a dynamic template mechanism was also introduced that does need a template design [33]. A real-time OMR system was introduced which was based on a web camera [34]. An important phase was the identification of the OMR document that was added during the process of OMR [35]. An important work on the detection of tampered OMR documents is discussed in their work [36]. An in-depth literature review on the OMR systems compares their performances [37].

As technology evolved, phone cameras have become powerful and rich enough to capture detailed images. Another major development in the field of OMR systems was to perform the OMR process using smartphone cameras instead of optical scanners [14], [38], [39]. The performance of phone cameras as a substitute for optical scanning in OMR systems is affected by new challenges. One such study is reported in the literature [40]. In such phone camera-based tests, a bold vertical line was used for skew detection and correction of the OMR document [41]. Perspective correction of images results in the improved accuracy of the OMR system [42], [43]. They also used black squares as fiducial markers to find the four corners of the region of interest in the OMR document.

Smartphones have been widely used by an ever-increasing population since 2010. Many people already have smartphones, making it possible to perform OMR by phone cameras, thus allowing a cheaper and more affordable solution for the low-budget class of people. Other benefits include their mobility, ubiquitousness, and high energy efficiency as compared to optical scanners and desktop PCs. These advantages motivated us to design a reliable OMR algorithm that is robust enough to meet the relevant challenges of OMR documents captured by smartphone cameras.

A robust OMR system should consider issues that arise due to noise and introduced artifacts, resulting in increased accuracy [36]. These reported issues stay even when phone cameras are used as a substitute to optical scanners. Additionally, phone camera-specific issues appear, thus there is a need to propose an algorithm for a smartphone camera case. For this purpose, three algorithms are tagged as candidates for selection of an appropriate one that can be tuned for requirements of our current case of a smartphone camera. In our previous work [36], these algorithms are named as Algorithm 1, Algorithm 2, and Algorithm 3. For preciseness, we denote these three algorithms based on the approach used therein [36]:

- Algorithm OMR_PP (Algorithm 1 [36] that uses pixel projection, i.e. predetermined position of circles);
- Algorithm OMR_HT (Algorithm 2 [36] that uses Hough Transformation);
- Algorithm OMR_FT (Algorithm 3 [36] which is our previous proposal and introduces fault tolerance in the OMR system).

From now onwards, these algorithms will be referred to as OMR_PP, OMR_HT and OMR_FT.

In Algorithm OMR_FT, we deal with the detection of skew angle even if the OMR document misses information like fiducial markers or its damaged variants. Damages to the fiducial markers are due to torn, folded, or badly printed documents, and also due to artifacts induced during their printing and scanning. This allows us to define four research questions:

- RQ1: How to handle skew errors when fiducial markers are undetectable?
- RQ2: How to detect marks on OMR documents with distortion, or translation for improving the accuracy?
- RQ3: How to detect user input when the circles are deshaped either due to any artifact on the OMR document, or either due to overfilling by the users.
- RQ4: How to extract the user input if other areas on the OMR document are damaged.

Smartphones are becoming more affordable, power efficient, lightweight, and easy to use. These features motivated us to make a reliable smartphone camera based OMR system with acceptable and improved accuracy. Conventional scanner based OMR_HT and OMR_FT are the state-of-the-art solutions found in the literature but they suffer in poor accuracy when OMR fiducial markers are damaged, OMR documents are torned, OMR documents have a translation of pixels, or the circles are deshaped by the users. In the experiments section, these solutions are compared with our proposed algorithm.

This paper is organized as follows. Related work is presented in Section II, which features two seminal approaches in Subsections II-A and II-B. The process of OMR systems is explained in Section IV. This section also includes our classification of errors in Subsection IV-G. The proposed methodology is discussed in Section V. The experimental details, results, and discussions are given in Section VI. The conclusion of the work is drawn in Section VII.

II. RELATED WORK

The majority of the work found in the literature is based on two seminal approaches, which we discuss next. Minor modifications of these approaches have been adopted to serve specific purposes. These variations set the basis for successful works.

A. OMR SYSTEMS BASED ON PREDETERMINED CIRCLE POSITIONS

Predetermined positions of the circles are the predominantly basis of early approaches [44], [45], [46]. At the time of creating an OMR document, the positions of all circles with respect to the anchor points / fiducial markers are stored in a JSON/XML file or a database. Mostly, the left top anchor is selected by the OMR designers for the relative positioning of the circles. Flag points can also be added in place of anchor points [44]. The search of the four fiducial markers is done with the help of pattern-matching techniques. The top left anchor point is considered to be the starting point of the OMR calculation and is considered as the origin $P(0, 0)$. The locations of the anchor points have a significant impact on the readability of the OMR markings on the OMR document. Failure or inaccuracy in finding the position of the anchor point fails to read the marks on that OMR document. Hence, the accuracy of the fiducial markers has a key role in the accuracy of the OMR system. For determining the response of the user on circles, relative positions stored in the JSON/XML file or the database are retrieved and the cursor scanning the pixels is moved to that position. The process now traverses $n \times n$ pixels counting the black ones in that circle, where n is the diameter of the circles in pixels. If the number of black pixels in the circle's area exceeds the minimal threshold value, it is considered to be marked by the user. The point to be noted here is that if the position of the fiducial marker is not accurate, the cursor will not move on top of the circle exactly, and will scan the wrong area, whereas the user has marked in a different place on that OMR document.

This method of OMR is suitable for OMR images acquired by the scanners but still lacks accuracy. The reason why this method is not suitable for smartphone cameras is that the OMR pictures captured by it are not precise enough. Every photograph taken may differ largely from other images. There is also variability of light conditions and angles at which the user holds the phone each time acquiring a picture. Whereas, when using a scanner, we don't have to face the issue of perspective correction and bad illumination. When using a phone camera, we need to handle these two additional issues as compared to the optical scanners. For the OMR images acquired by the phone camera, we need a better approach to achieve accuracy in the OMR system. A list of related works with minor variations is listed in Table 1.

B. OMR SYSTEMS BASED ON DETECTION OF CIRCLE POSITIONS

In this method, Hough Transformation is used to determine the circle positions. Positions of all possible circles are

TABLE 1. Existing approaches based on pixel projection method.

Year	Method
2015 [9]	A position detection pattern box as fiducial marker.
2017 [8]	A bold border is used.
2008 [44], 2013 [10]	A tracing pattern line is used.
2017 [41], 2008 [44], 2018 [12]	Black box to trace each row is used.
2017 [27]	Circles as fiducial markers are used.
2015 [46]	Black border with checkit logo is used.
2005 [23], 2011 [16], 2017 [40]	Black box as fiducial markers is used.
2019 [47]	Vertical line in the center is added.
2009 [11]	Stars as fiducial markers are used.
2015 [20]	Image processing to find ROI.
2015 [25]	Template matching approach is used.
2019 [48]	Angle brackets as fiducial markers are used.

determined in the binary image using pattern matching [33], [49], [50], [51], [52], [53], [54]. Circle positions are not permanently stored, but rather kept in memory till the process is complete. If the number of black pixels within each determined circle exceeds the minimum threshold value, the circle is considered marked. The major difference of this method from the previous methods is that there is no need for fiducial markers, nor do the circle positions need to be stored in an XML/JSON file or a database. Hough Transformation allows runtime calculation of circle positions.

Non-dependency on fiducial markers allows relief from the relevant errors - also the printer wheel translation errors are eliminated. Nevertheless, this method brings the challenge of the inability of the Hough Transform algorithm to calculate circle positions if any circle shape is distorted or the circle is missing. Thus the accuracy of the OMR system is reduced. A list of related work with minor variations is listed in Table 2.

TABLE 2. Existing approaches based on detection of circles.

Year	Method
2016 [53], 2019 [17], 2018 [28], 2021 [29], 2019 [52], 2019 [51]	Hough Transform to detect circles.
2013 [14]	Bold lines with small peaks.
2011 [39], 2012 [6], 2018 [54], 2020 [55], 2019 [33], 2023 [49]	Hough Transform / Canny Edge to detect ROI.
2018 [56]	Bubble template matching is used.
2015 [30], 2013 [26]	Shape based matching is used.
2010 [42]	Box finding algorithm is used.
2021 [50]	Location of darkened circles is detected.

C. OTHER APPROACHES

Few other approaches have given promising results. The canny edge detection algorithm is applied to the OMR document to find the position of the region of interest [28], [29], [49]. The bubble pattern matching method focuses on minimizing the effect of issues that emerge during the processing of OMR documents [56]. It is based on determining the regions of interest [57]. In another method, the images are classified before giving them differential treatment [18]. There has been an experimental study of using unsupervised learning of Machine Learning Pixel to cluster pixel-based images [15]. Light sensors are also used to detect user-filled marks on the OMR documents [2]. OMR data extraction is also done using tensor flow neural networks [58].

III. GAP ANALYSIS

In Section II, we have discussed two commonly used methods of OMR. One is the pixel projection method. The pixel projection method fails if one of the fiducial markers is damaged or missed during the handling of an OMR document. In that case, most of the OMR systems fail to calculate the skew angle of the OMR document. The OMR system also fails to scan the correct area on the OMR document. Translation introduced during the printing and scanning of the OMR document is another cause of reduced accuracy. The second method is based on the algorithms that finds the positions of all circles on the OMR document. Later these positions of circles are scanned for user input. The circle detection algorithm may fail to identify a circle on an OMR document if the circle is missed during the printing or scanning process. A circle may be deshaped when filled by the user. A circle may also be not detectable by the algorithm if it is deshaped due to translation introduced during the scanning or printing phase. A circle may also be misinterpreted due to artifacts in its region. Both optical scanners and smartphone cameras have inherent limitations when used as input methods for OMR.

Our previous work [36] addresses the above challenges for the OMR documents captured by optical scanners. Our current proposed algorithm focuses on OMR documents that are captured by smartphone cameras, duly addressing the mentioned issues, hence improves the accuracy. It utilizes a circle detection method to determine the position of the circle on an OMR document. Our approach employs a heuristic technique to identify potential circle locations, thus optimizing accuracy in circle detection. The algorithm identifies the position of circles within the potential areas of the OMR document. If the circle detection method fails to locate a circle in the circle regions, the relative position is determined based on the adjacent circles. The skew angle is corrected with the help of identified circles by the circle detection algorithm and hence we get rid of unnecessary fiducial markers on the corners of the OMR document. Details of our proposed algorithm are given in Algorithm 1 (OMR_FTSC).

IV. THE PROCESS OF OMR

This section explains the complete process of an OMR system, discussing the steps involved.

A. CREATING THE OMR DESIGN AND PRINTING THE OMR DOCUMENTS

The first step is the creation of an OMR document using any word processing or image processing application. OMR documents can be printed by any printing press or a laser printer or an inkjet printer. As reported in the literature, there is a basic requirement of adding special symbols or patterns called fiducial markers, on the four corners of the OMR document. Fiducial markers can be angle brackets, solid squares, thick circles, or any shape of a special pattern, which can be easily recognized by computer algorithms. Another approach reported in the literature is printing a dotted border or thick border in place of fiducial markers. In our case, we have used the circles having a diameter of 30 pixels, but this size can be changed according to the requirements. A few key aspects of such OMR systems are listed here.

i. When printing at a large scale, strict quality control checkpoints need to be ensured to get accuracy from the OMR system.
ii. A printer is a mechanical device, and as we know the nature of mechanical devices is more likely to have wear and tear as compared to non-mechanical devices. Pixel-perfect printing of the OMR documents is required to get good accuracy from the OMR system when implemented with most of the methods found in the literature.
iii. A printer's drum may get dirty or can have minor damage, due to which noise is induced in the printed OMR document. This noise is highly undesirable and results in low accuracy of the OMR system.
iv. Missing printing in some areas of the OMR document is also noticed. It fails to achieve good accuracy from the OMR system.

B. USER INPUT ON THE OMR DOCUMENTS

In this step, the circles on the printed OMR document are filled by the user. The possible cases of mistakes while filling the circles on the OMR document are:

i. More than one circle is filled, whereas only one filled circle is expected by the OMR algorithm.
ii. Partially filled circles.
iii. Circles that are not filled carefully appear in non-circular shapes, thus the OMR algorithm does not recognize them.
iv. None of the circles is filled, whereas one filled circle is expected.

C. SCANNING AND PREPROCESSING OF IMAGES

After the OMR documents have undergone input, they are scanned by optical scanners, which are predominantly the devices used for the purpose. As an alternate scanning device, smartphones are handy. In this work, we calibrate the performance of our already reported state-of-the-art algorithm [36] fed with smartphone camera OMR images. The results allow us to propose an optimal algorithm for such smartphone camera images. To overcome the limited memory issue of a medium-rated smartphone, the acquired images can be stored in network storage. Even a locally created MySQL database may be feasible. Further, only the case of smartphone camera images is focused.

There is a need to preprocess the acquired images before actually doing the OMR calculations. When images are acquired by a smartphone camera, images likely have perspective errors. Their correction is mandatory when it is beyond the acceptable value [25]. The perspective correction is beyond the scope of our work, so we use one existing technique. After perspective error correction, the OMR document is converted to a binary image (black and white) using a threshold value i.e., 127. Conversion to binary image also fixes the issue of bad illumination to some extent, if the value of threshold is selected carefully. Using an adjustable threshold value is a smart way to accomplish image binarization [59], [60]. This will support the OMR document's handling of various lighting circumstances. When data is supplied by optical scanners, information can be extracted from OMR documents with a preset threshold value. However, we must determine the threshold value when photos are taken under various lighting circumstances. The value of the threshold depends on the brightness of the OMR document. Also, the complexity of image processing is reduced to a great extent. The algorithm performance on binary images is much higher as compared to that on color-rich images, where each pixel is represented by three bytes (red, green, and blue). Deviation correction and noise removal to improve the quality of the OMR document are also done at this stage.

If an OMR document is worn out, or if the OMR document is tilted beyond an acceptable angle, then the current state-of-the-art methods are not able to calculate and correct the skew angle. Without skew angle correction, the OMR system can not extract responses from the OMR documents.

D. IDENTIFICATION AND CORRECTION OF SKEW ANGLE ERRORS

Scanning of the fiducial markers (angle brackets or circles) is generally done by the current methods so that the OMR document's skew angle may be calculated. The fiducial marker positions detected by the algorithm are treated as reference positions in the OMR document. Also, these fiducial markers calculate the skew angle and allow correction of the skew angle.

E. PROCESSING OF THE FILLED CIRCLES

During this step, the filled circles are detected to determine the consumer's responses. The method used in this step is crucial as it directly impacts the performance of the OMR system. Inaccurate calculation of the filled circle position results in scanning the wrong area, thus compromising the accuracy. To this end, two commonly used methods are discussed in the literature, which we have already included in Subsections II-A and II-B.

F. SCORE CALCULATION

After the user response is determined, this step is used to group the circles and determine the marked answers from the user. A marked answer can be from a group of two or more circles. For instance, a true/false type question has two circles, while a typical MCQ type question may require four or five circles. As an analogy, it is similar to radio buttons used in an HTML form. In case, an OMR system is used to evaluate the performance of users, the marked answers are compared with the correct answers stored in the database, and the score is determined accordingly, without any manual effort.

TABLE 3. A classification of commonly occurring errors in OMR system.

Method	Errors
Pixel Projection Method	• One of the fiducial markers is missing. • Translation introduced in the OMR document.
Hough Transform Method	• A circle is missed during printing or scanning. • De-shaped circles due to overfill by the user or some other reason. • Artifacts introduced in the OMR document.

G. CLASSIFICATION OF ERRORS

We have classified error types that arise when using the methods discussed in Subsection II-A and Subsection II-B, and given in Table 3.

V. THE PROPOSED ALGORITHM

The two methods used in OMR systems, the Pixel Projection Method and the Hough Transform Method set the extent of errors that the system faces. Both methods have their pros and cons. The first method has an extra dependency on finding the positions of anchor points or fiducial markers on an OMR document. If fiducial markers are damaged, the accuracy of the OMR system is directly compromised. Another issue with this method is the introduction of translation in the OMR document, which has a huge impact on the accuracy of the OMR system. The second method overcomes the issues of the first method, but it introduces a new issue of missing circles in OMR documents. The circles can go undetected due to missed printing, de-shaped circles, or due to introduced artifacts. Due to their issues, these two methods are not able to achieve high accuracy in general, while results have further deteriorated in the case of images acquired by smartphone cameras. However, the accuracy is still acceptable when images are acquired by optical scanners. As a common observation, improvement is nevertheless still needed for OMR systems used for performance evaluation of subjects. Thus, the need of a new algorithm that could address the mentioned issues was established and fulfilled in our previous work [36]. The performance of the previously proposed algorithm is promising. This proposed algorithm is currently denoted as Algorithms OMR_FT (please see Section Introduction). Next, we focus on the derivation of yet another algorithm that takes images acquired by smartphone cameras.

There are multiple benefits of using a smartphone camera to capture images of OMR documents. These include smartphone portability and mobility due to being lightweight and small in size, no attachment to wires, low energy consumption, cost-effectiveness, and high availability. Thus the idea of using a smartphone camera for OMR is feasible and offers significant energy savings for an institution requiring extensive scanning of OMR documents regularly. It is worth mentioning here that a smartphone camera being an electronic device, does not undergo wear and tear, which is normally a device using mechanical parts faces. Optical scanners are highly dependent on mechanical mechanisms, due to which their OMR accuracy is directly affected. The cheaper price advantage of smartphones over optical scanners is also a fact.

For bench marking, the performance of the three algorithms, Algorithms OMR_PP, OMR_HT, and OMR_FT [36] is evaluated on images captured by a smartphone camera. Given that the error rate of the first two algorithms is up to 15% when using an optical scanner, the error rate increases significantly in the case of a smartphone camera. This is due to the artifacts introduced by smartphone cameras. Still, the third algorithm exhibits noticeable fault tolerance. The obtained results steered us to identify challenges that have to be overcome if acceptable performance is to be derived. Thus, in the current work, we underpin the case of images captured by smartphone cameras and propose yet another algorithm.

To proceed, the primary concern is to underpin and address the challenges of using smartphone cameras for OMR documents. The first and foremost challenge is the high chance of perspective errors in the captured images, which is because most phones are not held at the correct angle while capturing images. This perspective error affects the accuracy of the used OMR algorithm. During our experiments, we captured the images of OMR documents very carefully to minimize the perspective errors. The second challenge is regarding the light conditions that vary easily during various tests. Hence, the smartphone camera images are taken in various light conditions to check the impact of light on the performance under various light profiles. The initial results guided us to tune Algorithm OMR_FT to propose our new algorithm, Algorithm OMR_FTSC (Fault

Algorithm 1 OMR_FTSC
// Fault Tolerant Optical Mark Recognition Using a Smartphone Camera

Input: OMR image scanned by smartphone camera
Output: Circles indicating user input

```
1  convertImageToBinary();
   // Conversion of a color or a
      greyscale image to a binary image
      described in Algorithm 2.
2  applyGussianBlurToImage();
   // Removal of noise to smooth out
      circles for detection.
3  templateCircles ← readPostionFromJSON();
      // Reading relative position of
      each circle from the stored JSON
      file or a database.
4  circles ← FindCircles();
   // Finding all circles in the image
      by applying Hough Transform
      Algorithm.
5  Lines ← DrawLinesOnXAxis();
   // Drawing lines by joining centers
      of circles along x axis.
6  FixSkewAngle(Lines);
    circles ← FindCircles();
   // Finding all circles in the image
      by applying Hough Transform
      Algorithm after fixing skew that
      can potentially change the
      position of circles.
7  Regions ← RegionBoundaries();
   // Finding circles on the boundaries
      and returning their position.
8  forall currentRegion of Regions do
       // Looping through all regions.
9      currentRegionCircles ←
        FCRC(templateCircles);
10     forall currentCircle of currentRegionCircles do
           // Looping through all the
              circles for the current
              region that is input from
              the template file.
11         probabilityRegion ←
            FindProbabilityRegion(currentCircle);
12         if circleExists(probabilityRegion) = TRUE
            then
13             CalculateByHT();
14         end
15         else
16             CalculateByPP();
17         end
18     end
19 end
```

Algorithm 2 convertImageToBinary()
// Binarization of Image Using Adaptive Threshold

Input: Grayscale image I of size $W \times H$, sub-image size $N \times N$
Output: Binarized image B of size $W \times H$

```
1  subImages ← getSubImages(); // Divide
      image into subimages of size
      N × N
2  index ← 0;
3  forall subImage of subImages do
       // Looping through all subImages.
4      index ← index + 1;
5      adaptiveThresholdValue ←
          getAvgGrayscale(subImage)
          // Calculate average grayscale
          value from all pixels in the
          subImage
6      binaryImage ←
        applyBinarization(adaptiveThresholdValue)
        subImages[index] ← binaryImage;
7  end
8  binarizedImage ←
     combineSubImages(subImages);
9  return binarizedImage;
```

Tolerant Optical Mark Recognition using a Smartphone Camera). Algorithm OMR_FTSC is optimized for our tagged case of images captured by smartphone cameras and gives accuracy improvement over the Algorithm OMR_FT. Algorithm OMR_FTSC is explained as follows.

Lines 1-3 outline the preprocessing stage of the Algorithm OMR_FTSC, during which the quality of the OMR document is improved for further analysis. Algorithm 2 provides a detailed description of the function invoked in Line 1 of Algorithm OMR_FTSC, specifically the convertImageToBinary() function. Lines 4-6 outline the skew angle correction in the Algorithm OMR_FTSC. It ensures appropriate processing and interpretation of the OMR documents by correcting any angular misalignment in the documents. Lines 7-10 show the analysis of the OMR document. This stage contains computational operations that are intended for regions of interest in the OMR documents. Finally lines 11-20 outline the main functionality of Algorithm OMR_FTSC regarding the extraction of user input from the OMR document.

VI. EXPERIMENTS & DISCUSSION

The experimental setup is shown in Fig. 4. We tested the performance of our algorithm using two smartphones to acquire images, Samsung A23 and Infinix Zero X. Two smartphones are used to cross-validate our results across different hardware platforms. Nevertheless, both smartphones are lower mid-range devices that are priced

TABLE 4. Capabilities comparison of our proposed method with existing featured works.

Type of issue	**Pixel Projection Method (OMR_PP)** [9] [8] [44] [10] [41] [44] [12] [27] [46] [23] [16] [40] [47] [11] [20] [25] [48]	**Hough Transform Method (OMR_HT)** [53] [17] [28] [29] [52] [51] [14] [39] [6] [54] [55] [33] [49] [56] [30] [26] [42] [50]	**Proposed Algorithm (OMR_FTSC)**
One of the fiducial marker or tracing line is damaged.	✗ Unable to correct skew angle error.	✓ Can extract user input.	✓ Handle skew angle up to 30 degrees.
Translation introduced in the OMR document.	✗ Unable to extract user input.	✓ Can extract user input.	✓ Handle translation up to 100 pixels.
A circle is missed during printing or scanning.	✓ Can extract user input.	✗ Unable to extract user input.	✓ Handle upto one missed circle per question.
De-shaped circles due to overfill by the user or some other reason.	✓ Can extract user input.	✗ Unable to extract user input.	✓ Handle any number of deshaped circles.
Artifacts introduced on areas other than regions of interest.	✗ Unable to correct skew angle error.	✗ Detect artifacts as OMR circles.	✓ Handle any number of artifacts.

TABLE 5. A performance comparison of our proposal with conventional approaches based on experimental results.

Algorithm	**Image capture method**	**Number of documents**	**Number of errors**	**Software error percentage**
OMR_PP	Smartphone camera	300	135	$\frac{129}{300} \times 100 = 43\%$
OMR_HT	Smartphone camera	300	142	$\frac{142}{300} \times 100 = 47\%$
Proposed OMR_FTSC	Smartphone camera	300	10	$\frac{15}{300} \times 100 = 3.33\%$

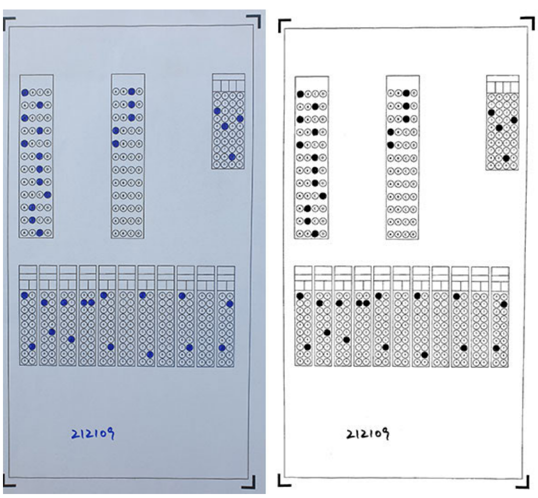

FIGURE 1. Image captured in normal outdoor light condition (a) original (b) binary.

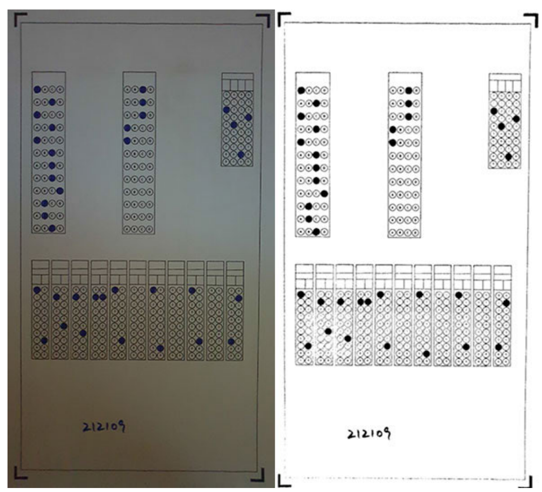

FIGURE 2. Image captured in low light condition (a) original (b) binary.

not more than $250 each in the United States. Selection of these smartphones allows assessment of a cost-effective solution, also due to their high availability everywhere - thus aiming to propose a low-cost OMR system. We experimented under various lighting conditions. Fig. 1 was taken outside under ambient light. Comparably, Fig. 2 was captured in a very dimly lit room or without any light at all. Fig. 3 was captured in direct sunlight with some shade. The image was captured on 24 December 2023 at 3:00 pm in Multan, Pakistan (30.1864° N, 71.4886° E). The OMR document's shadow effect was successfully eliminated during the binarization process. Three hundred OMR documents were printed and filled by a group of three hundred users to ensure variability of filling patterns. After acquiring the images of these OMR documents using the mentioned phones, they were sent to the PC for further processing by the four OMR algorithms, OMR_PP, OMR_HT, OMR_FT, and

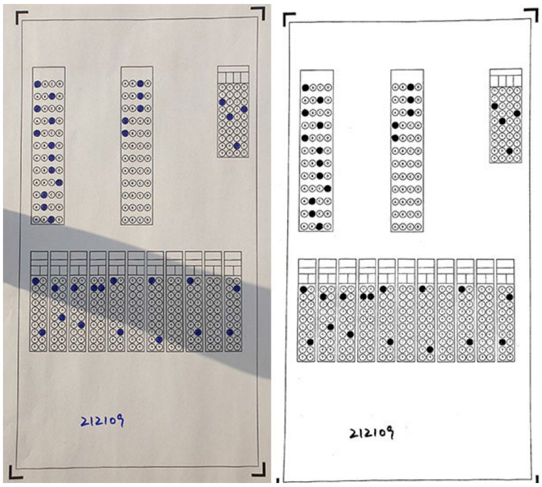

FIGURE 3. Image captured in direct sunlight with shadow (a) original (b) binary.

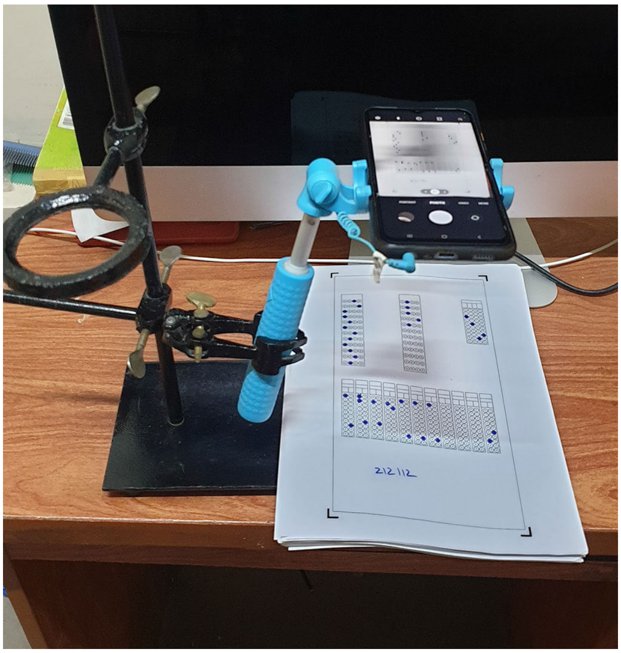

FIGURE 4. Experimental setup.

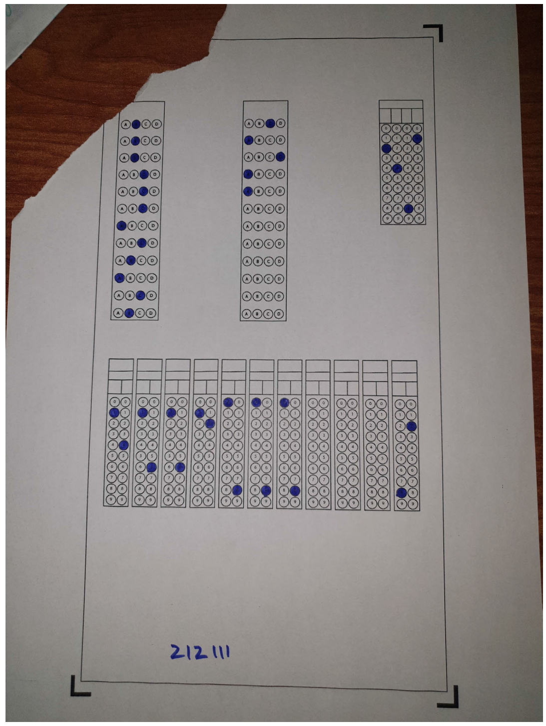

FIGURE 5. OMR document with missing fiducial marker (Original).

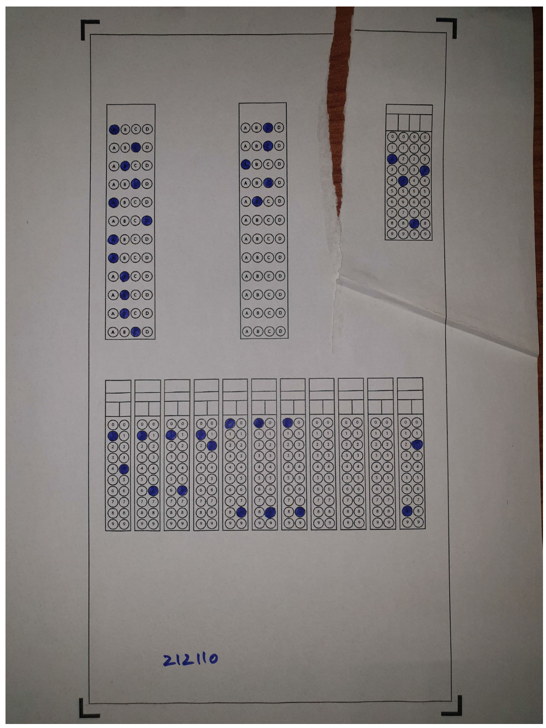

FIGURE 6. Case study where unwanted artifact introduced on OMR document (Original).

OMR_FTSC. The process was partially automated by writing a few command-line PHP scripts to read circles from the images using the Open CV image library. The PC has a Core i3 3rd generation CPU coupled with 8 GB RAM. Other than using a PC, it is also feasible to use the selected smartphones. Since our focus is not on the execution performance of the computing device used, but rather only on the accuracy performance of Algorithm OMR_FTSC, so we are restricted to using PC only. Even any low-end smartphone not capable of executing the algorithms can transmit the images to a cloud service that can execute them.

The capabilities of Algorithm OMR_FTSC, and those of Algorithm OMR_PP and Algorithm OMR_HT are given in Table 4. Our proposal recognizes the circles on the OMR documents. For each recognized circle, the radius depends upon the resolution of the image acquired, thus the radius threshold is adjusted accordingly during the preprocessing step. Another improvement is achieved by converting the

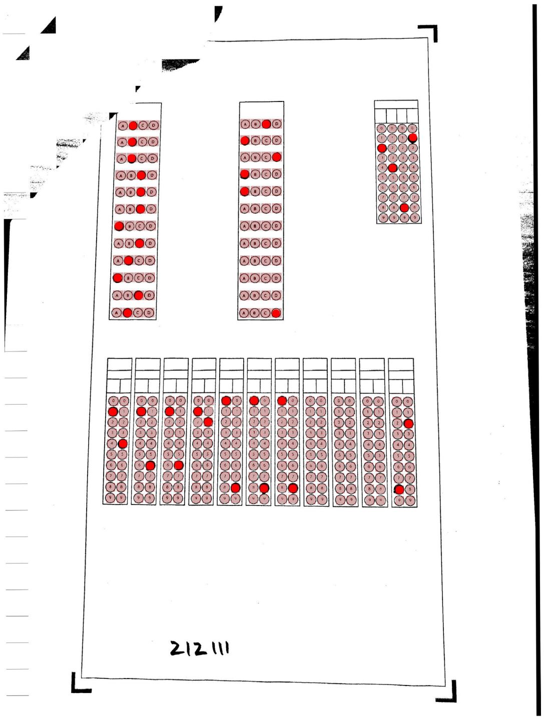

FIGURE 7. OMR document with missing fiducial marker (Processed).

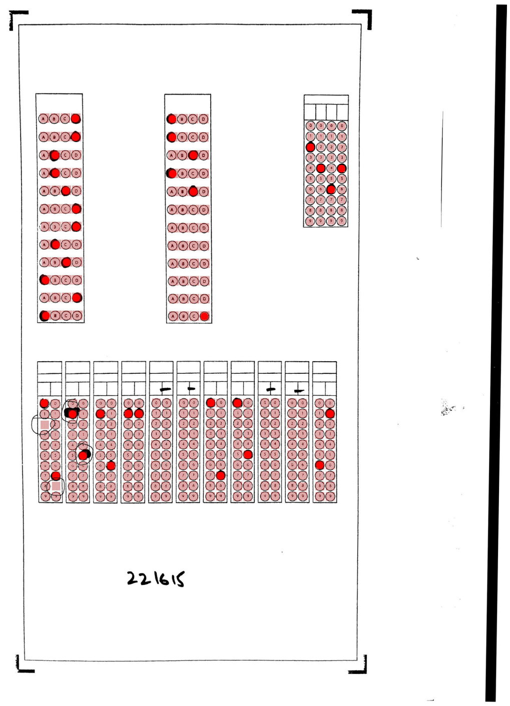

FIGURE 9. Case study with missing circle and deshaped circle.

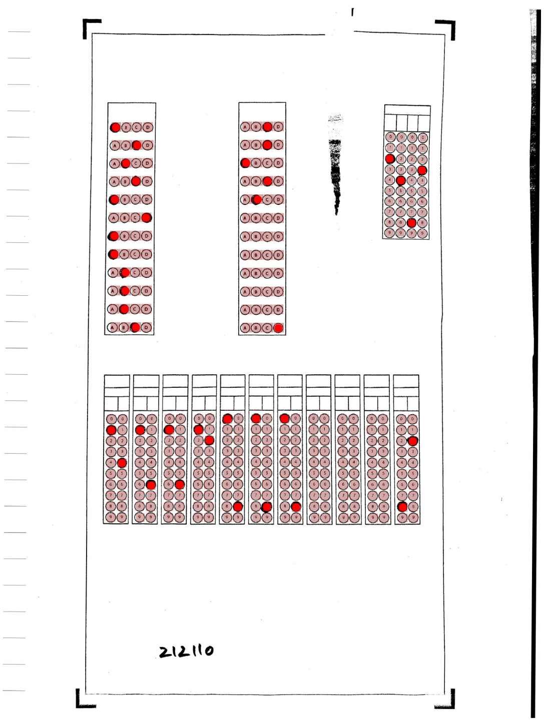

FIGURE 8. Case study where unwanted artifact introduced on OMR document (Processed).

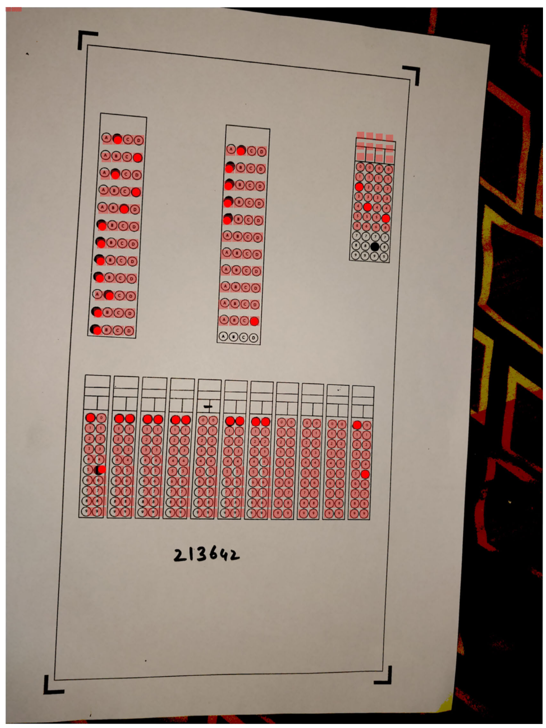

FIGURE 10. OMR document failed to read circles in paper code region due to tilt and perspective error.

images to binary and then by applying Gaussian blur. Gaussian blur removes the unwanted noise and makes the circles smooth enough to be detected by the Hough Transform algorithm.

The capabilities of Algorithm OMR_FTSC are due to the fundamental reason that it is based on Algorithm OMR_FT [36], which is an improvement over conventional approaches. The latter already addresses the following

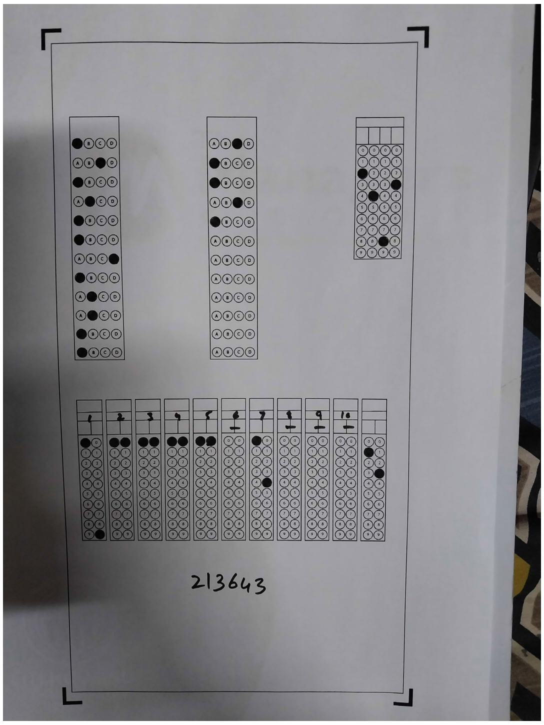

FIGURE 11. OMR document failed due to shadow in the captured image.

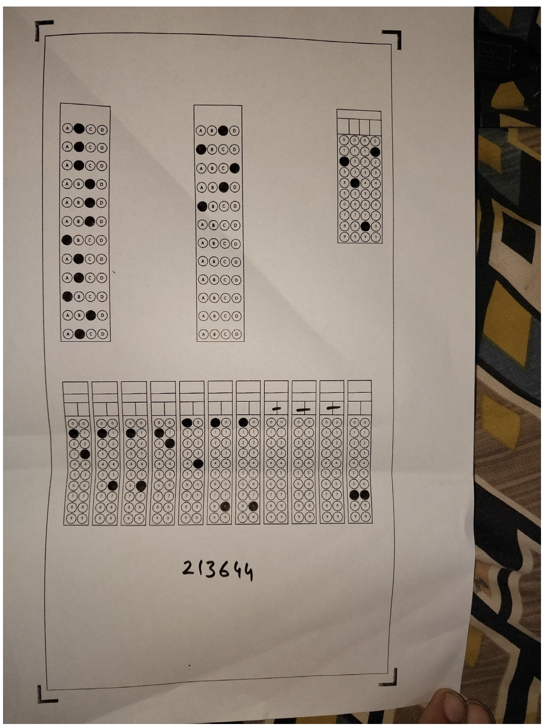

FIGURE 12. A de-shaped OMR document due to mishandling during the process.

issues effectively by readability of OMR documents when the following issues prevent conventional approaches from having a reasonable performance:

i. OMR documents have fiducial markers missing;
ii. OMR documents have unwanted artifacts;
iii. OMR documents have translation of pixels;
iv. OMR documents have deshaped circles;
v. OMR documents are partly torned or folded during scanning.

The performance of Algorithm OMR_FTSC that emerged out of experimental results is excellent (summarized in Table 5). Our proposed algorithm performs very well for images acquired both from smartphone cameras and from optical scanners. We achieved an accuracy of about 97% when images were captured carefully by placing the phone camera on top of the OMR documents. On the other hand, the performance of OMR_PP algorithm and OMR_HT algorithm remained low. The same dataset of OMR images was used for both algorithms. The main causes of low performance are perspective issues and excessive translation in the OMR documents. Nevertheless, we made an effort to minimize the perspective error while capturing the images. In general, the accuracy of the OMR algorithms when used with a phone camera is slightly lower than that with the scanner, but it can be increased by improving the perspective error correction during preprocessing of the images.

Its highly pertinent to point out here that Algorithm OMR_FT [36] is also tested on Smartphone camera images. It is not able to read any image captured by a smartphone camera. We associate this 0% success rate due to the following reasons:

i. Smartphone cameras have varying resolutions, due to which it is not able to adjust itself dynamically;
ii. There is not much attention paid to capturing OMR document from an appropriate distance. Algorithm OMR_FT was not designed to handle such situations;
iii. There is no control exercised over varying lighting conditions when Smartphone cameras are used, which is yet another aspect Algorithm OMR_FT lacks in capability;
iv. Shadows on OMR documents appear while capturing images with Smartphone cameras, whereas optical scanners do not have this issue.

For our experiments, the application is made debug-enabled for the OMR process. For that purpose, the scanned images are altered by drawing dotted lines on the marked circle areas that are sought by the application. If the position of the dotted lines is exactly on top of the circle, it reflects that the OMR algorithm is scanning the area perfectly. Otherwise, the dotted lines show that the OMR algorithm is scanning at the wrong place in the OMR document. Fig. 10 shows the wrong scanning area at the paper code region. This means due to the tilt and perspective error, the OMR algorithm fails to identify the region for paper code and hence produces wrong results. Similarly, we have artificially added artifacts on the OMR document to perform experiments to our proposed system. It can be seen clearly in Fig. 6 that the OMR document is readable unless the actual OMR area is not damaged due to artifacts on the OMR document. It can also be noticed that translation was introduced in the paper code region due to the tear and folding of the paper. It can be seen that the algorithm can handle such issues, as Fig. 8 shows the situation after processing. This type of error is produced when the OMR document is captured without being careful of the tilt angle. One of the fiducial markers and borderline is damaged in Fig. 5 captured by phone camera. The situation after processing of the OMR document is shown in Fig. 7. Another case is shown in Fig. 9 wherein two circles are missing and two circles are deshaped, but still there is no effect on the performance of the proposed OMR system. All the fiducial markers are removed intentionally to see how our algorithm behaves in that case. Another case of a shadow introduced in the image is shown in Fig. 11, wherein the OMR system fails. The sample image is taken by turning off the flashlight of the smartphone camera. One of the de-shaped samples of the OMR document is shown in Fig. 12. Our algorithm reads marks successfully from the image. The general observation is that the algorithm can read marks when the circles are deshaped. Our OMR algorithm does not detect such cases, resulting in reduced accuracy.

VII. CONCLUSION

Alongside other futuristic technologies, smartphones are also enablers of the fifth industrial revolution (5IR). They offer an infrastructure towards pervasiveness and ubiquitousness. Our proposed algorithm OMR_FTSC aims for an OMR system that uses images acquired by smartphone cameras, thus realizing mobility, easiness, efficiency, and cost-effective solution. Our current proposal is based on our previous proposal that already beats in performance of the existing benchmark algorithms.

We have not corrected perspective errors in our experiments, instead, we have tried to capture images with great care so that the acquired images will have minimum perspective error.

In the future, many challenges need to be addressed. For instance, perspective errors beyond small angles are not considered in the current work. There is a need to design the OMR system flexible enough such that it can process images that are taken from a wider range of angles. Thus, coping with situations of high perspective errors. More precisely, the idea is to cater for errors of type shown in Fig. 10. This would be achieved by introducing preprocessing for perspective errors. The second challenge is to cater to the issues that arise due to the threshold values greater than the shadow (please refer to errors shown in Fig. 11). If we can adaptively reduce the threshold, our algorithm can read marks perfectly, but the threshold reduction renders images unrecognizable by our existing algorithm. This occurs because reducing the threshold value results in a loss of image content. By making a smart and dynamic threshold value that can be applied using an automated analysis of image, this situation can be handled. In other words, the OMR system needs to be fault-tolerant from yet another aspect. Such an improvement would allow the subject to capture images without worrying about the placement issues of phone cameras on top of the OMR documents. The third challenge is the OMR algorithm angle-robust. The fourth challenge is to study the issues when images are captured by Webcams, which are quite similar to phone cameras. It would be interesting to test our proposed algorithm for webcam scenarios.

ACKNOWLEDGMENT

The authors are thankful to Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R387), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

REFERENCES

[1] J. L. Pérez-Benedito, E. Q. Aragón, J. A. Alriols, and L. Medic, "Optical mark recognition in Student continuous assessment," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 9, no. 4, pp. 133–138, Nov. 2014, doi: 10.1109/RITA.2014.2363005.

[2] S. Kumar, "A study on optical mark readers," *Int. Interdiscipl. Res. J.*, vol. 3, no. 11, pp. 40–44, 2015.

[3] D. D. Poor, "Image capture and storage techniques in association with optical mark reading," U.S. Patent 5 452 379, Sep. 19, 1995.

[4] K. Chinnasarn and Y. Rangsanseri, "Image-processing-oriented optical mark reader," in *Proc. SPIE*,, vol. 3808, pp. 702–708, Oct. 1999, doi: 10.1117/12.365883.

[5] N. C. R. Fisher, N. Brown. (2000). *Image Tranlate*. Accessed: Nov. 14, 2023. [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/translte.htm

[6] A. M. Abdu and M. M. Mokji, "A novel approach to a dynamic template generation algorithm for multiple-choice forms," in *Proc. IEEE Int. Conf. Control Syst., Comput. Eng.*, Nov. 2012, pp. 216–221.

[7] A. Papandreou, B. Gatos, S. J. Perantonis, and I. Gerardis, "Efficient skew detection of printed document images based on novel combination of enhanced profiles," *Int. J. Document Anal. Recognit. (IJDAR)*, vol. 17, no. 4, pp. 433–454, Dec. 2014, doi: 10.1007/s10032-014-0228-5.

[8] D. Chai, "Automated marking of printed multiple choice answer sheets," in *Proc. IEEE Int. Conf. Teaching, Assessment, Learn. for Eng. (TALE)*, Dec. 2016, pp. 145–149.

[9] P. Sanguansat, "Robust and low-cost optical mark recognition for automated data entry," in *Proc. 12th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON)*, Jun. 2015, pp. 1–5.

[10] S. Rakesh, K. Atal, and A. Arora, "Cost effective optical mark reader," *Int. J. Comput. Sci. Artif. Intell.*, vol. 3, no. 2, pp. 44–49, Jun. 2013.

[11] C. Saengtongsrikamon, P. Meesad, and S. Sodsee, "Scanner-based optical mark recognition," *Inf. Technol. J.*, vol. 5, pp. 69–73, 2009.

[12] I. A. Belag, Y. Gültepe, and T. M. Elmalti, "An image processing based optical mark recognition with the help of scanner," *Int. J. Eng. Innov. Res.*, vol. 7, no. 2, pp. 108–112, 2018.

[13] A. Awny Abbas, "An automatic system to grade multiple choice questions paper based exams," *J. Univ. Anbar Pure Sci.*, vol. 3, no. 1, pp. 174–181, Apr. 2009.

[14] A. Al-Marakeby, "Multi core processors for camera based OMR," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 1–5, Apr. 2013.

[15] E. O. Gyamfi and Y. M. Missah, "Pixel-based unsupervised classification approach for information detection on optical markup recognition sheet," *Adv. Sci., Technol. Eng. Syst. J.*, vol. 2, no. 4, pp. 121–132, Aug. 2017.

[16] A. Spadaccini and V. Rizzo, "A multiple-choice test recognition system based on the gamera framework," 2011, *arXiv:1105.3834*.

[17] E. M. De Elias, P. M. Tasinaffo, and R. Hirata, "Alignment, scale and skew correction for optical mark recognition documents based," in *Proc. 15th Workshop Comput. Vis.*, 2019, pp. 26–31.

[18] M. Afifi and K. F. Hussain, "The achievement of higher flexibility in multiple-choice-based tests using image classification techniques," *Int. J. Document Anal. Recognit. (IJDAR)*, vol. 22, no. 2, pp. 127–142, Jun. 2019.

[19] C. Young, G. Lo, K. Young, and A. Borsetta, "FormScanner: Open-source solution for grading multiple-choice exams," *Phys. Teacher*, vol. 54, no. 1, pp. 34–35, Jan. 2016.

[20] S. B. Gaikwad, "Image processing based omr sheet scanning," *Int. J. Adv. Res. Electron. Commun. Eng. (IJARECE)*, vol. 4, no. 3, pp. 519–522, 2015.

[21] E. H. Barney Smith, D. Lopresti, and G. Nagy, "Ballot mark detection," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.

[22] R. H. Hasan and E. I. A. Kareem, "An image processing oriented optical mark reader based on modify multi-connect architecture MMCA," *Int. J. Modern Trends Eng. Res.*, pp. 414–424, 2015.

[23] S. Hussmann and P. W. Deng, "A high-speed optical mark reader hardware implementation at low cost using programmable logic," *Real-Time Imag.*, vol. 11, no. 1, pp. 19–30, Feb. 2005.

[24] Y. R. Hadi, F. Karim, and A. Mohammad Hadi, "A novel approach of skew estimation and correction in Persian manuscript text using radon transform," in *Proc. IEEE Symp. Comput. Informat. (ISCI)*, Mar. 2012, pp. 198–202.

[25] N. Karunanayake, "OMR sheet evaluation by web camera using template matching approach," *Int. J. Res. Emerg. Sci. Technol.*, vol. 2, no. 8, pp. 40–44, 2015.

[26] J. A. Fisteus, A. Pardo, and N. F. García, "Grading multiple choice exams with low-cost and portable computer-vision techniques," *J. Sci. Educ. Technol.*, vol. 22, no. 4, pp. 560–571, Aug. 2013.

[27] J. A. Catalan, "A framework for automated multiple-choice exam scoring with digital image and assorted processing using readily available software," in *Proc. DLSU Res. Congr.*, 2017, pp. 1–5.

[28] R. K. V. Srinivas, "OMR evaluation using image processing," *Int. J. Innov. Advancement Comput. Sci.*, vol. 7, no. 4, pp. 572–576, 2018.

[29] C. Y. C. Buleje, Y. T. P. Atencio, and E. E. C. Tinoco, "System with optical mark recognition based on artificial vision for the processing of multiple selection tests in school competitions," in *Proc. 46th Latin Amer. Comput. Conf. (CLEI)*, Oct. 2020, pp. 172–177.

[30] N. Ahmad, "OMR form inspection by web camera using shape-based matching approach," *Int. J. Res. Eng. Sci. (IJRES)*, vol. 3, no. 3, pp. 1–7, 2015.

[31] M. J. Ab Aziz, F. D. Ahmad, A. A. Abdul Ghani, and R. Mahmod, "Automated marking system for short answer examination (AMS-SAE)," in *Proc. IEEE Symp. Ind. Electron. Appl.*, vol. 1, Oct. 2009, pp. 47–51.

[32] N. Kakade and D. R. C. Jaiswal, "OMR sheet evaluation using image processing," *J. Emerg. Technol. Innov. Res. (JETIR)*, vol. 4, no. 12, pp. 640–643, 2017. [Online]. Available: www.jetir.org

[33] N. C. Dayananda Kumar, K. V. Suresh, and R. Dinesh, "Automated parameter-less optical mark recognition," in *Data Analytics and Learning*, D. S. Guru, B. H. Shekar, and Y. H. S. Kumar, Eds., Singapore: Springer, 2019, pp. 185–195.

[34] H. Atasoy, E. Yildirim, Y. Kutlu, and K. Tohma, "Webcam based real-time robust optical mark recognition," in *Neural Information Processing*, S. Arik, T. Huang, W. K. Lai, and Q. Liu, Eds., Cham, Switzerland: Springer, 2015, pp. 449–456.

[35] M. Alomran and D. Chai, "Automated scoring system for multiple choice test with quick feedback," *Int. J. Inf. Educ. Technol.*, vol. 8, no. 8, pp. 538–545, 2018.

[36] Q. Hafeez, W. Aslam, M. Ikramullah Lali, S. Ahmad, M. Alqahtani, and M. Shafiq, "Fault tolerant optical mark recognition," *Comput., Mater. Continua*, vol. 74, no. 2, pp. 3829–3847, 2023. [Online]. Available: http://www.techscience.com/cmc/v74n2/50190

[37] E. M. de Elias, P. M. Tasinaffo, and R. Hirata, "Optical mark recognition: Advances, difficulties, and limitations," *Social Netw. Comput. Sci.*, vol. 2, no. 5, p. 367, Sep. 2021, doi: 10.1007/s42979-021-00760-z.

[38] D. Patel and S. Zaid, "Efficient system for evaluation of OMR sheet," *Ijariie*, vol. 3, no. 3, pp. 1349–1362, 2017.

[39] T. D. Nguyen, Q. H. Manh, P. B. Minh, L. N. Thanh, and T. M. Hoang, "Efficient and reliable camera based multiple-choice test grading system," in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, Aug. 2011, pp. 268–271.

[40] H. Tjahyadi, Y. G. Budijono, S. Lukas, and D. Krisnadi, "Android based automated scoring of multiple-choice test," *Int. J. Mach. Learn. Comput.*, vol. 7, no. 5, pp. 110–113, Oct. 2017.

[41] D. Krisnadi, A. R. Mitra, R. I. Desanti, W. D. Ciputra, and H. Hery, "A multiple-choice test recognition system based on Android and RBFNN," *DEStech Trans. Comput. Sci. Eng.*, pp. 501–505, 2017, doi: 10.12783/DTCSE/CMSAM2017/16423.

[42] F. D. A. Zampirolli, J. A. Q. Gonzalez, and R. P. D. O. Neves, "Automatic correction of multiple-choice tests using digital cameras and image processing," *Universidade Federal do ABC*, 2010.

[43] R. T. China, F. D. A. Zampirolli, R. P. D. O. Neves, and J. A. Quilici-Gonzalez, "An application for automatic multiple-choice test grading on Android," *Revista Brasileira de Iniciačão Científica*, vol. 3, no. 2, pp. 4–25, 2016.

[44] H. Deng, F. Wang, and B. Liang, "A low-cost OMR solution for educational applications," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl.*, Dec. 2008, pp. 967–970.

[45] A. Gupta and S. Avasthi, "Image based low cost method to the OMR process for surveys and research," *Int. J. Sci. Eng. Appl. Sci. (IJSEAS)*, vol. 2, no. 7, pp. 91–95, 2016. [Online]. Available: https://www.researchgate.net/publication/335169247

[46] R. Patel, S. Sanghavi, D. Gupta, and M. S. Raval, "CheckIt—A low cost mobile OMR system," in *Proc. IEEE Region 10 Conf.*, Nov. 2015, pp. 1–5.

[47] K. Rachchh and E. S. Gopi, "Inclusion of vertical bar in the omr sheet for image-based robust and fast omr evaluation technique using mobile phone camera," in *Proc. 2nd Int. Conf. Data Eng. Commun. Technol.*, A. J. Kulkarni, S. C. Satapathy, T. Kang, and A. H. Kashan, Eds. Singapore: Springer, 2019, pp. 39–46.

[48] M. P. Calado, A. A. Ramos, and P. Jonas, "An application to generate, correct and grade multiple-choice tests," in *Proc. 6th Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2019, pp. 1548–1552.

[49] G. Himabindu, A. Reeta, A. S. Manikanta, and S. Manogna, "Evaluation of optical mark recognition (OMR) sheet using computer vision," *Int. Res. J. Modernization Eng. Technol. Sci.*, vol. 5, no. 4, pp. 5–9, 2023.

[50] T. Jingyi, Y. K. Hooi, and O. K. Bin, "Image processing for enhanced OMR answer matching precision," in *Proc. Int. Conf. Comput. Inf. Sci. (ICCOINS)*, Jul. 2021, pp. 322–327.

[51] P. Raundale, T. Sharma, S. Jadhav, and R. Margaye, "Optical mark recognition using open CV," *Int. J. Comput. Appl.*, vol. 178, no. 37, pp. 9–12, Aug. 2019.

[52] V. Ware, N. M. Menon, P. Varute, and R. A. Dhannawat, "Cost effective optical mark recognition software for educational institutions," *Int. J. Advance Res., Ideas Innov. Technol.*, vol. 5, pp. 1874–1877, 2019.

[53] G. Bayar, "The use of Hough transform to develop an intelligent grading system for the multiple choice exam papers," *Karaelmas Sci. Eng. J.*, vol. 6, no. 1, pp. 100–104, 2016.

[54] Z. Küçükkara and A. E. Tümer, "An image processing oriented optical mark recognition and evaluation system," *Int. J. Appl. Math., Electron. Comput.*, vol. 6, no. 4, pp. 59–64, Dec. 2018.

[55] S. Kumar and D. Anu Rathee, "Implementation of OMR technology with the help of ordinary WebCam," *Int. J. Modern Trends Sci. Technol.*, vol. 6, no. 12, pp. 77–81, Dec. 2020.

[56] S. C. Loke, K. A. Kasmiran, and S. A. Haron, "A new method of mark detection for software-based optical mark recognition," *PLoS ONE*, vol. 13, no. 11, Nov. 2018, Art. no. e0206420.

[57] G. M. R. I. Rasiq, A. Al Sefat, and M. M. F. Hasnain, "Mobile based MCQ answer sheet analysis and evaluation application," in *Proc. 8th Int. Conf. Syst. Model. Advancement Res. Trends (SMART)*, Nov. 2019, pp. 144–147.

[58] Y. Ju, X. Wang, and X. Chen, "Research on OMR recognition based on convolutional neural network tensorflow platform," in *Proc. 11th Int. Conf. Measuring Technol. Mechatronics Autom. (ICMTMA)*, Apr. 2019, pp. 688–691.

[59] J. Sauvola, T. Seppanen, S. Haapakoski, and M. Pietikainen, "Adaptive document binarization," in *Proc. 4th Int. Conf. Document Anal. Recognit.*, vol. 1, 1997, pp. 147–152.

[60] T. R. Singh, S. Roy, O. I. Singh, T. Sinam, and K. M. Singh, "A new local adaptive thresholding technique in binarization," 2012, *arXiv:1201.5227*.

QAMAR HAFEEZ received the B.S. and M.S. degrees in computer science from Bahauddin Zakariya University, Multan, Pakistan, and the Ph.D. degree in computer science from The Islamia University of Bahawalpur, Pakistan, in 2023. He is an accomplished professional with a strong foundation in computer science. His research interests include image processing, computer networks, the Internet of Things, and network security.

WAQAR ASLAM received the M.Sc. degree in computer science from Quaid-i-Azam University, Islamabad, Pakistan, and the Ph.D. degree in computer science from Eindhoven University of Technology, The Netherlands, funded by the Overseas Scholarship from HEC, Pakistan. He is currently a Professor of computer science and IT with The Islamia University of Bahawalpur, Pakistan. His research interests include performance modeling and QoS of wireless/computer networks, performance modeling of (distributed) software architectures, radio resource allocation, the Internet of Things, fog computing, effort/time/cost estimation and task allocation in (distributed) agile software development, social network data analysis, DNA/chaos-based information security, machine learning, and AI.

ROMANA AZIZ received the bachelor's degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, and the master's and Ph.D. degrees in computation from The University of Manchester (UMIST), U.K. She is currently an Associate Professor with the College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Saudi Arabia. She previously with the University of Twente, The Netherlands, and Prince Sultan University, Saudi Arabia. Her research interests include decision technologies and analytics, IT/IS service quality, logistics, and supply chain management. She is a fellow of the Higher Education Academy, U.K.

GHADAH ALDEHIM received the B.S. degree in computer science from KSU, and the M.S. degree in information systems and the Ph.D. degree in data mining from the University of East Anglia, U.K. She is currently an Assistant Professor with the College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Saudi Arabia. Her research interests include data science, machine learning, data mining, and big data.