

# Automated Paper-based Multiple Choice Scoring Framework using Fast Object Detection Algorithm

Pham Doan Tinh, Ta Quang Minh  
School of Electrical and Electronic Engineering  
Hanoi University of Science and Technology  
Hanoi, Vietnam

**Abstract**—Optical mark reader (OMR) technology is an important research topic in artificial intelligence, with a wide range of applications such as text processing, document recognition, surveying, statistics, and process automation. Researchers have proposed many methods employing either traditional image processing and statistics or complex machine learning models. This paper presents a feasible solution for the OMR problem. It uses a fast object detection model to detect markers effectively and then segment the answer sheet into smaller regions for the mark reader model to recognize the user's selections accurately. The experimental results on actual answer sheets from college exams show that the error is less than 0.5 percent, and the processing speed can achieve up to 50 answer sheets per minute on standard core i5 personal computers.

**Keywords**—Optical mark reader; multiple choice exam; automatic scoring; segmentation; fast object detection

## I. INTRODUCTION

In the era of digitalization and automation, the education sector has attracted significant attention due to its potential to revolutionize traditional educational methods by incorporating cutting-edge technologies to improve the quality of education and academic management. In teaching, evaluating learning progress and the assessment of the learners is very important. Automating this process by applying technology such as an Optical Mark Reader (OMR) attracts the attention of many researchers and organizations. OMR technology has become essential for the automatic multiple choice scoring system, especially in large-scale competitions.

OMR is now widely used for exams or surveys with multiple choice answers [1]. According to Zhang et al. [2], this is the most common type of exercise used in education. This technology focuses on rapidly detecting data extracted from filled-in forms created with a pencil or pen. OMR technology involves the use of Multiple Choice Questions (MCQs) in exams, which allows quick results for students, serves as a tool for teachers and educational institutions to apply in their exams, reduces the need for manual labor, and improves performance. OMR initially appeared as a dedicated hardware solution [3, 4, 5, 6] or using paid resources [7, 8, 9, 10]. These approaches have often been studied before. But then, software solutions [11, 12, 13, 14, 15] appeared along with the development of technology, gradually replacing specialized hardware devices. OMR approaches can be divided into two main categories: Using conventional image processing [4, 16, 17, 18], and using artificial intelligent machine learning [11, 19, 20]. In conventional image processing approaches, first, they adjust the orientation of the input image [5, 11, 20], then apply

the segmentation techniques to search for areas that need identification [5, 14]. After that, they detect whether the answer area is circled based on the grayscale level [21, 22] or the number of pixels in the area [18, 23, 24]. These approaches are easy to build and have a short implementation and runtime. However, they may need to fully capture the complex attributes and variations of each specific test, leading to low accuracy.

The limitation of conventional methods has led to growing interest in deep learning methods, especially the convolutional neural networks (CNN), which have demonstrated superior image processing and recognition capabilities. Deep learning offers the potential for many research fields such as image and signal processing [25, 26]. It provides more accurate and robust OMR systems capable of handling diverse types of tests. In addition to the processing algorithms used in the pure image processing approach, this method builds a neural network suitable for the problem. The classification techniques [13, 19, 23, 27] are commonly used. This technique can accurately and quickly identify an answer box to identify whether an answer is selected.

In addition, the input images may come from many sources like cameras, webcams [2, 28] or from smartphones [16, 19, 20, 29]; this factor also dramatically affects construction costs and model implementation time. Models using many image formats and sources will save time and effort and reach more users.

Several methods of deploying the system into software [4, 12] on desktop or mobile devices have built a relatively complete system. The benefit of this is that it can be used flexibly in many places and has high practical applications. These systems often require users to print or create exam papers using predefined software [9]. However, this must ensure excellent and stable performance because it is difficult to maintain, modify, and add features.

According to Sumit Tiwari and colleagues [30], manipulation of OMR board data is shared and affects exams nowadays. This form of data tampering has not been taken into account by existing systems. This article aims to use an algorithm to encode the characteristics of the answers and information students have highlighted in the answer sheet. Then, create a QR code and use that QR code to evaluate whether the exam paper is fake or not. A novel method that achieves successful research results can be applied in practice.

This article addresses the above research limitations by proposing a deep learning method based on the YOLO (You Only Look Once) algorithm to score multiple choice tests

accurately. We use YOLOv8 because it stands out as the fastest model with lower parameters compared to the other versions [31]. This study uses a data set of real-life multiple choice test sets and training and testing processes to create a powerful and effective model. The contributions of the research include flexible use of input images, low implementation costs, and high accuracy requirements, which is important to propose a fast, easy-to-use method with the use of an optimal resource.

The rest of the article is presented as follows: Section II offers the proposed system architecture and detailed algorithm implementation. Section III presents the test and the results evaluations. The final section concludes the article with the future direction of the system given in Section IV.

## II. METHOD

### A. Answer-sheet design

The answer sheets given to students in each exam are designed as shown in Fig. 1. The above form was redesigned from the answer sheet in Vietnam's national high school exam.

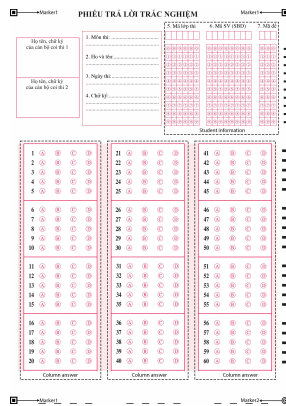


Fig. 1. Sample answer sheet.

The answer sheet has the student's registration number, exam code, and exam-class sections to get information about students and exam questions, making the process of statistics and data processing easier when the data set is large. In addition, the answer section has a maximum of 60 questions, and you can optionally score specific questions, which is suitable for multiple-choice exams.

### B. Overall System Implementation

In Fig. 2, we build a general system diagram for the system analyzed in the previous section. This diagram can be used to understand how the system's components interact with each other, as well as provide an overview of the system's architecture and functionality.

We propose to divide the method into phases: Segment and preprocess data phase, Labeling phase, Training phase, and Online Recognition phase. These stages are presented in detail in the following sections.

### C. Segmentation and Pre-processing Phases

Before entering recognition, to achieve a balanced accuracy and performance, YOLO recommends that the model's input

image be sized 640x640. The model will even resize the input image to have the most significant side size set to 640 and maintain the original aspect ratio. Because of this, if the image is not segmented into small parts, small details will be lost when the input image is trained. Therefore, we improved the accuracy by focusing on the desired portion of the original input image (segmenting the portion and keep the original resolution).

We can find the constant lines that surround the blocks and shapes to segment the input image into student's information section and the question answer choice section as showed in Fig. 3:

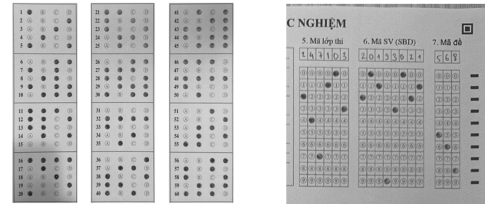


Fig. 3. Segmented image.

Segmented image components after cropping will be resized prior to recognition. Here, we will resize to have the most significant edge size set to 640 and keep the same proportions.

The training answer sheets are divided into two sets: training and validation with 85% and 15%, respectively.

### D. Labeling phase

After having preprocessed data, we build a labeling process for the model. Labeling is defining bounding boxes around class types in the image and placing captions for each box. The model can be trained to detect and classify classes in the following training phase by accurately locating the courses in the photo. Here, we label each cropped image with the Labelling software.

1) *Marker*: In reality, the input images can be skewed, rotated, etc. The coordinates of the markers that help us specifically handle these problems will be presented in the next part. We placed three markers in three corners: Top left, top right, and bottom left of the exam paper with the same shape and labeled them as "marker1". The marker in the lower right corner using other shape type and labeled as 'marker2' shown in Fig. 4.

2) *Question-Answer Section*: In this section, each question will have four answer options; each question can have many correct answers, so we will have  $2^4$  cases where the answer is selected. Therefore, we use 16 labels, encoded in bits 0 and 1, shown in Table I and labeled as in Fig. 5.

3) *Student Information Section*: Student information includes the exam class code, student registration number, and exam code. These fields are identified by integer numbers from 0 to 9. Therefore, we use 10 labels, shown in Table II and labeled as in Fig. 6.

### E. Training phase

During the training process, there are several main steps as the following description:

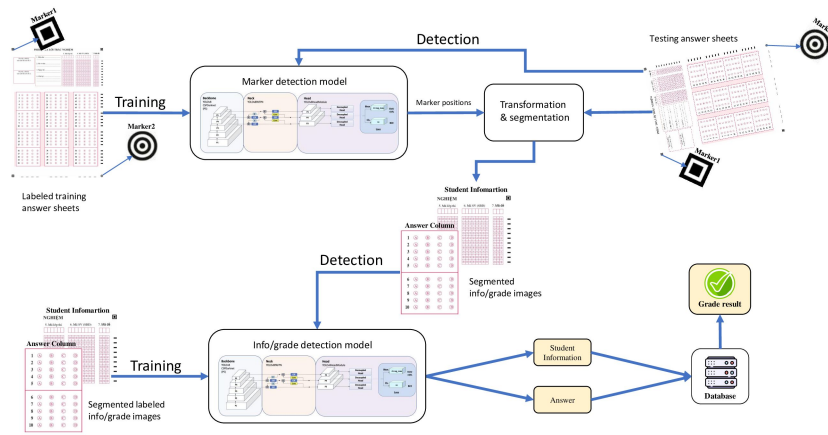


Fig. 2. System overview diagram.

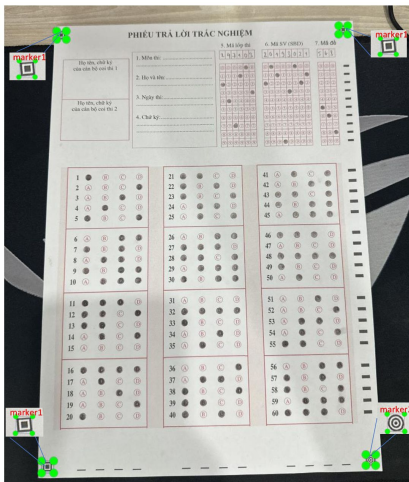


Fig. 4. Marker-labeled image.

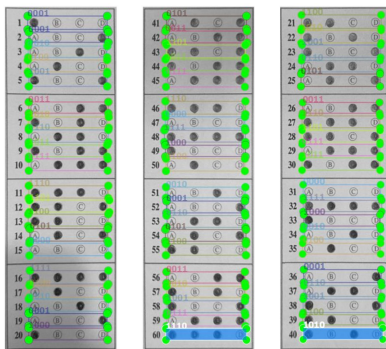


Fig. 5. Labeling answers.

TABLE I. LABEL ANSWER

Label	Value	Label	Value
1000	A	0101	B and D
0100	B	0011	C and D
0010	C	1110	A, B and C
0001	D	1101	A, B and D
1100	A and B	1011	A, C and D
1010	A and C	0111	B, C and D
1001	A and D	1111	A, B, C and D
0110	B and C	0000	Not selected

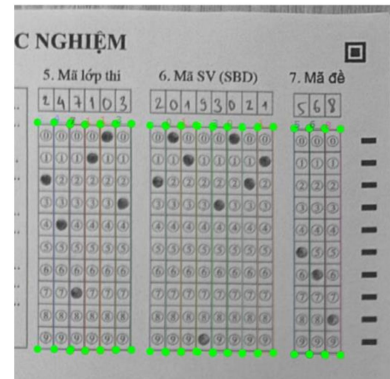


Fig. 6. Labeling student information,

model. Metrics such as loss, mAP are monitored to assess the model. These parameters are explicitly described in Table III.

TABLE III. CUSTOM TRAINING MODEL

Parameter	Value
Model	YOLO
Image size	640 × 640
Number of Epochs Trained	150
Batch Size	16
Number of classes	29

a) *How to use the training set:* The training set accounts for 85% of the total data collected. This data set includes labeled images that correspond to each class.

b) *Select parameters:* YOLOv8's training configuration contains parameters such as Number of classes, Image size, Number of epochs, and Batch size. The training process is monitored to evaluate the progress and effectiveness of the

TABLE II. LABEL INFO

Label	Value
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
unchoice	Not selected

c) *Model Optimization*: After completing model training, the next step is to test and fine-tune the model. This step is done to ensure that the model can perform well in new tests and is accurate in different types of questions.

The performance and accuracy of the model are evaluated through commonly used parameters in Machine Learning in general and object recognition problems in general: Precision, recall, mean precision (AP) and mean average precision (mAP) [32]:

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \quad (1)$$

In the Formula 1:

- $TP$  : Number of cases correctly predicted as Positive.
- $FP$  : Number of cases predicted to be Positive but actually Negative.
- $FN$  : Number of cases predicted to be Negative but actually Positive.

From Precision and Recall, we calculate the average accuracy of the object detection model:

$$AP = \sum_{i=0}^{i=n-1} [Recalls(i) - Recalls(i + 1)] \times Precisions(i) \quad (2)$$

Thence inferred:

$$mAP = \frac{1}{h} \sum_{i=1}^{i=n} AP_k \quad (3)$$

In the Formulas 2 and 3:

- $h$  is the number of classes
- $Recalls(i)$  and  $Precisions(i)$  are the value of the  $i^{th}$  element of the Recalls and Precisions array
- $AP_k$  is the AP value of the  $i^{th}$  class

#### F. Online Recognition Phase

The input image is taken directly from the camera or smartphone, so it is impossible to avoid the cases where the input image has different angles and distances from the camera to the answer sheet or cases where the input image is blurry,

incorrect and misaligned. This stage's purpose is to process the image to extract the part of the image that only contains multiple-choice answer sheets. The test paper must be aligned in the most appropriate direction, brightness, and color to be included in the identification model.

First, predict the input image, the target to identify four markers, and we get position marker. After the YOLO model recognized four markers (3 square markers and one circle marker), we got the coordinates of the four markers on the original exam paper. Note that the order of the detected angles is unconventional. Because of the above reason, we need to rearrange the four corners in the correct order. Based on the Position Marker (PM), we determine the direction of the image by placing three markers1 in positions: top-left, top-right, bottom-left, and marker 2 in the bottom-right position. Therefore, to retrieve the part of the image that only contains multiple-choice answer sheets, we rotate the image so that marker 2 is always in the bottom-right position.

Call the top left point  $P_1$ , the top right point  $P_2$ , the bottom right point  $P_3$ , the bottom left point  $P_4$ . Suppose that each point is defined by the coordinates  $(x,y)$ :

$$PM = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\} \quad (4)$$

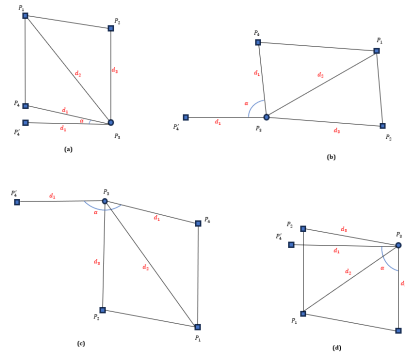


Fig. 7. Illustration of the input image.

Fig. 7 illustrates the first image of the process. After obtaining the position of marker 2 through identification, we consider this position to be the new bottom-right position. Then rotate the remaining corners according to this marker2 position.  $P'_4$  is the new bottom-left position. In fact,  $P'_4$  can be in many places around  $P_3$ . We need to determine the rotation angle  $\alpha$ , wherever  $P'_4$  is.

First, determine the coordinates  $P_4$  among the 3 marker1 coordinates. Because  $P_4$  is considered a bottom-left point, based on the distance, we determine  $P_4$  is the point with the shortest distance to  $P_3$ , specifically  $d_1 < d_2 < d_3$ :  $P_4 = \{(x_i, y_i) \in PM | d_{P_4} = \min\{d_1, d_2, d_3\}\}$ . In the next step, determine the coordinates  $P'_4, P'_4$  at the new bottom-left position, with a distance equal to  $P_4$  to  $P_3$ , so the coordinates of  $P'_4$  are always equal to:

$$\begin{cases} x_{P'_4} = x_{P_1} - d_1 \\ y_{P'_4} = y_{P_3} \end{cases} \quad (5)$$



From the coordinates  $P'_4$ , applying the trigonometric formula for triangle  $P'_4P_4P_3$ , we calculate angle  $\alpha$  with  $d_{P_4P'_4}$  is the distance from  $P_4$  to  $P'_4$ :

$$\alpha = \cos^{-1} \cdot \frac{2d_1^2 - d_{P_4P'_4}^2}{2d_1^2} \quad (6)$$

Rotating the input image with angle  $\alpha$ , we obtain a new image rotated in the correct direction. After corner points have been identified and target points have been calculated, we use the `getPerspectiveTransform` and `warpPerspective` functions to transform and align the input image. This image processing step, which includes markers, produces a transformed image that only contains the answer sheet. Additionally, the image is correctly rotated.

We will obtain an image that only includes the answer sheet and has been transformed accurately. We tested our model on a variety of image samples, including different orientations and lighting conditions, to obtain an image that only contains the answer sheet. The model still works well in most cases. The algorithm used in this procedure is shown in Algorithm 1.

---

**Algorithm 1** Image Preprocessing

---

**INPUT:** Input image  
**OUTPUT:** Preprocessed image  
1: **begin**  
2: Read *input\_image*  
3: Recognition *input\_image*  
4: Position marker  
5: PM = Initial array position marker  
6: **if** (PM has 3 marker1) and (PM has 1 marker2) **then**  
7: Find  $d_{P_4} = \min \{d_1, d_2, d_3\}$   
8: Find  $\alpha = \cos^{-1} \cdot \frac{2d_{P_4}^2 - d_{P_4P'_4}^2}{2d_{P_4}^2}$   
9: Rotate *input\_image* with angle  $\alpha$   
10: Find destination corners (DC):  
DC = [ [0, 0], [maxWidth, 0], [maxWidth, maxHeight], [0, maxHeight] ]  
11: Get the background removed image from the DC: *image\_extracted*. Using `getPerspectiveTransform` and `warpPerspective`  
12: return *image\_preprocessed* = *image\_extracted*  
13: **else:**  
14: break  
15: **end**

---

Fig. 8 shows the final result of Algorithm 1. With this processed image, the following segmentation and identification of each component is much easier.

Get the image after preprocessing, crop the image to get: column answer image and student information image, recognition these images. Based on recognition results, we can extract the information of students and the answers from each answer sheet of the exam. Subsequently, comparisons with the correct answers associated with each exam class code were made, allowing each candidate to receive an automated scoring process. In addition, a threshold coefficient called  $\theta$  was introduced. This threshold is the decisive parameter that determines the confidence level required to consider a prediction to be

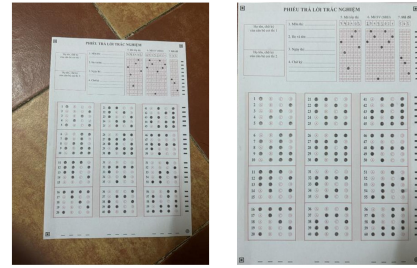


Fig. 8. Input image and pre-processed image.

"correct". If the confidence exceeds the specified threshold, the result of the recognition will be confirmed as accurate. On the contrary, if the confidence falls below the threshold, the prediction is considered wrong. The algorithm below describes the systematic identification and scoring process:

---

**Algorithm 2** Recognition and Grading

---

**INPUT:** Pre-processed image  
**OUTPUT:** Recognized images and mark  
1: **begin**  
2: Read *preprocessed\_image*  
3: Segmentation *preprocessed\_image*: *info\_student* and *column\_answer*  
4: Recognition *info\_student* and *column\_answer*  
5: Threshold =  $\theta$   
6: **If** confidence  $\geq$  threshold **then**  
7: Insert to database  
8: Write mark  
9: **else**  
10: Issue a warning  
11: **end:**

---

### III. EXPERIMENTAL RESULTS

A set of experiments was performed to evaluate the accuracy of the methods presented and the automated scoring systems. With an algorithm written in Python, the automatic scoring system is evaluated on a laptop running on an Intel Core i5 11<sup>th</sup> processor with 16GB RAM, recognizing input images including three parts that need to be recognized: marker, information, student, and answer information. The system will take a variable amount of time for the recognition process proportional to the number of questions on the answer sheet. When identifying votes with fewer answers, it will take less time. We experimented and calculated that the average time to recognize a 60-question answer sheet is 1.2 seconds.

The model is used to predict results for new tests. These results are transmitted to the scoring system to produce the final results shown in Fig. 9:

After testing and refining the model and continuing to train, our team achieved the following results after training in Fig. 10:

The Confusion Matrix chart shows the confusion between classes in the entire system. It can be seen that the confusion model is very little, shown in points other than the main diagonal (representing noise) and mainly confusion. between the

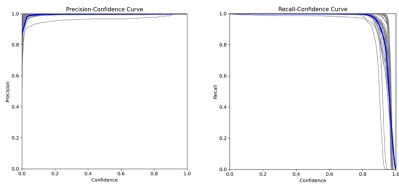


Fig. 9. Precision and recall.

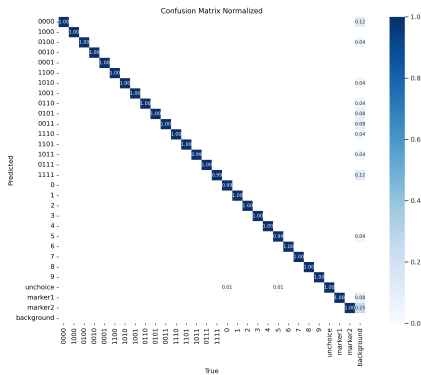


Fig. 10. Confusion matrix normalized chart.

background and the labels, not between the labels themselves, and this confusion level has very low reliability (0.01, 0.02, 0.03...).

According to the chart, we can see that the main diagonal is very thick and almost reaches 1, which means the model has high accuracy because the main diagonal of the matrix represents the number of cases in which the model correctly classifies objects into corresponding classes. The Fig. 11 depicts many graphs of the results after training the process.

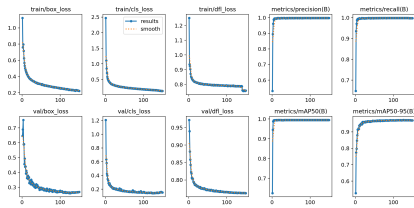


Fig. 11. Training results chart.

The *train/box\_loss* and *val/box\_loss* plots help us evaluate how well the model locates classes by measuring the difference between the predicted bounding-box coordinates and the class in reality. The decreasing trend in these two charts shows that the model is improving its accuracy in identifying the correct location of labels in the training and validation data sets.

The *train/cls\_loss* and *val/cls\_loss* plots illustrate the classification error. This histogram measures the difference between the predicted class probability and the actual label. The decreasing *cls\_loss* plot shows that the model is improving its ability to identify classes correctly.

The *train/dfl\_loss* and *val/dfl\_loss* plots illustrate the imbalance between classes with many labels and classes with

few labels. This histogram corrects the difference between the predicted probability and the target probability, especially on data sets with class imbalance. The decreasing *dfl\_loss* plot shows that the model made more balanced predictions and improved performance.

The precision and recall charts have curves near the highest curve, showing that the model achieves high precision and recall when changing the probability threshold. This indicates that the object recognition model can detect and locate objects.

mAP50 plots: This chart depicts the average accuracy; the model achieves a high value (approaching 1), showing that the model achieves high accuracy in object recognition; mAP50-95 chart If the model reaches a high value, it shows that the model can recognize objects well on many different levels of probability threshold. By monitoring these graphs and continuously improving the model based on the insights we gain, we can train the YOLOv8 model to accurately detect, locate, and classify classes for multiple choice exams.

The test set contains new tests that are not used to train the model. Based on Formula 3 and the results after training, the system can evaluate the performance of a model in a new test and calculate the parameters as described in Table IV:

TABLE IV. MEAN AVERAGE PRECISION

Parameter	Value
Evaluation Metric	mAP
Best Metric Scores	0.996

After completing the training phase, analyzing the input image and applying the recognition process, rectangles will be drawn on points that the model recognizes: around markers, student information and selected sentences (fill). The results obtained are shown in Fig. 12.

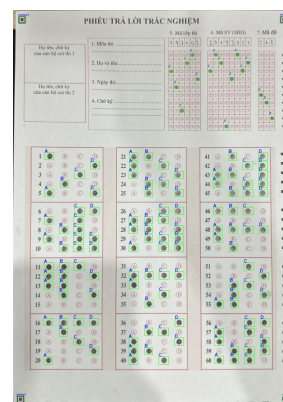


Fig. 12. Input image prediction.

To get an accurate experiment, we will use a data set such that input images are from different angles and resolutions, taken from many types of devices with diverse light intensities. In addition, on each of those answer sheets, the number of answers varies, with multiple answer choices. To make the test data set diverse, accurate, and most importantly, "realistic," we

used this system to automatically score 300 real-life multiple-choice tests with the semester's final exam at Hanoi University of Science and Technology.

Choosing the threshold value is an integral part of evaluating model performance. The input image is likely wrong when one of the detected model objects has a confidence level below the Threshold threshold that we have previously chosen. Perform the first rough tuning experiment: Choose  $0.6 \leq \theta \leq 0.9$ , each increment of 0.05 each (see Fig. 13).

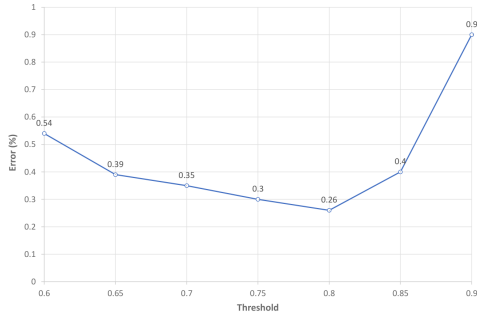


Fig. 13. Relationship between error and threshold  $\theta$ .

According to the results of the diagram above, perform the second experiment for fine tuning (see Fig. 14): Choose  $0.75 \leq \theta \leq 0.85$ , each mark is 0.01 apart. The purpose is to choose the most suitable  $\theta$  value: Our system shows

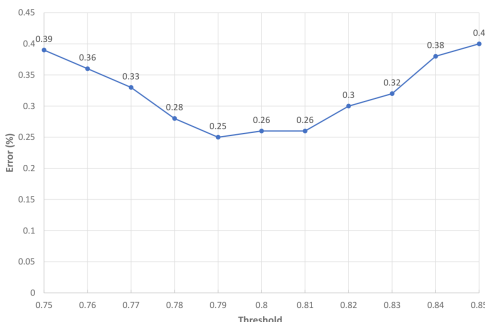


Fig. 14. Relationship between error and threshold  $\theta$  (fine tuning).

high accuracy when input images are of good quality, such as answer sheets taken in suitable, flat, and transparent lighting conditions. However, the system needs better-quality input images. For example, when the answer sheets are blurry, uneven, or have a lot of extra lines due to uneven scanning, the model needs help identifying the answers and information from the student. Especially when the input image lacks corners, markers are lost, leading to the image preprocessing process being unable to process. The accuracy of the system can be significantly reduced if necessary information is lost or contaminated. Based on the analysis results of Fig. 14, we decided to choose  $\theta = 0.79$  - a value large enough to have the lowest probability of errors occurring in the data set.

#### IV. CONCLUSION

This paper presented an automated paper-based multiple choice grading system with the utilization of using fast object detection algorithm. Research and experimental results on actual college exams have shown that employing YOLOv8 model together with pre-processing techniques improved the performance of the OMR system with the error rate less than 0.5% and processing time on stand personal computer around 1 second. This can help educational and assessment organizations perform test administration tasks more effectively. The article also highlights challenges and potential development directions. Integrating the system into real-world applications and improving the real-time application ability are also challenges worth considering in the future.

#### REFERENCES

- [1] E. M. de Elias, P. M. Tasinaffo, and R. H. Jr., "Optical mark recognition: Advances, difficulties, and limitations," *IEEE*, vol. 2, no. 367, 05 July 2021.
- [2] L. Zhang, B. Li, Q. Zhang, and I.-H. Hsiao, "Does a distributed practice strategy for multiple choice questions help novices learn programming?" *IEEE*, vol. 15, no. 18, September 2020.
- [3] P. Sanguansat, "Robust and low-cost optical mark recognition for automated data entry," *IEEE*, 20 August 2015.
- [4] V. Ware, N. Menon, P. Varute, and R. Dhannawat, "Cost effective optical mark recognition software for educational institutions," *IEEE*, vol. 5, no. 2, 21 September 2018.
- [5] N. C. D. Kumar, K. V. Suresh, and R. Dinesh, "Automated parameterless optical mark recognition," *IEEE*, vol. 43, pp. 185 – 195, 05 November 2018.
- [6] R. H. Hasan and E. A. Kareem, "An image processing oriented optical mark reader based on modify multi-connect architecture mmca," *INJMTER*, March 2015.
- [7] A. H. S. Saad, M. S. Mohamed, and E. H. Hafez, "Coverless image steganography based on optical mark recognition and machine learning," *IEEE*, vol. 9, pp. 16522 – 16531, 11 January 2021.
- [8] A. B. Talib, N. B. Ahmad, and W. Tahar, "Omr form inspection by web camera using shape-based matching approach," *IJRES*, vol. 3, no. 3, March 2015.
- [9] R. S. K. Atal, and A. Arora, "Cost effective optical mark reader," *IEEE*, vol. 74, no. 2, 31 October 2022.
- [10] G. R. I. Rasiq, A. A. Sefat, and M. F. Hasnain, "Mobile based mcq answer sheet analysis and evaluation application," *IEEE*, 16 June 2020.
- [11] R. Ahad, R. Toufiq, and S. U. Zaman, "Information retrieval from damage omr sheet by using image processing," *IEEE*, 31 December 2020.
- [12] J. L. Pérez-Benedito, E. Q. Aragón, and J. A. A. L. Medic, "Optical mark recognition in student continuous assessment," *IEEE*, vol. 9, no. 4, pp. 133 – 138, November 2014.
- [13] M. Afifi and K. F. Hussain, "The achievement of higher flexibility in multiple-choice-based tests using image classification techniques," *IJDAR*, vol. 22, pp. 127 – 142, 19 March 2019.
- [14] O. Espitia, A. Paez, Y. Mejia, M. Carrasco, and N. Gonzalez, "Optical mark recognition based on image processing techniques for the answer sheets of the colombian high-stakes tests," *IEEE*, vol. 1052, pp. 167 – 176, 09 October 2019.
- [15] E. M. de Elias, P. M. Tasinaffo, and R. Hirata, "Alignment, scale and skew correction for optical mark recognition documents based," *IEEE*, 21 October 2019.
- [16] H. Tjahyadi, S. Lukas, S. Albert, and D. Krisnadi, "Automated scoring of multiple-choice test using template matching technique," *ILCA*, vol. 55, pp. 1–5, 21 September 2018.
- [17] D. Chai, "Automated marking of printed multiple choice answer sheets," *IEEE*, 13 February 2017.

- [18] S. Hussmann and P. W. Deng, "A high-speed optical mark reader hardware implementation at low cost using programmable logic," *IEEE*, vol. 11, no. 1, pp. 19 – 30,, November 2015.
- [19] B. Haskins, "Contrasting classifiers for software-based omr responses," *IEEE*, 17 December 2015.
- [20] 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," *IEEE*, 26 September 2011.
- [21] R. Patel, S. Sanghavi, D. Gupta, and M. S. Raval, "Checkit - a low cost mobile omr system," *IEEE*, 07 January 2016.
- [22] F. de Assis Zampirolli, "Automatic correction of multiple-choice tests using digital cameras and image processing," *IEEE*, vol. 13, April 2013.
- [23] O. Gorokhovatskyi, "Neocognitron as a tool for optical marks recognition," *IEEE*, 06 October 2016.
- [24] H. Tjahyadi, S. Lukas, S. Albert, and D. Krisnadi, "A novel optical mark recognition technique based on biogeography based optimization," *IJITK*, vol. 5, no. 2, pp. 331–333, December 2012.
- [25] P. D. Tinh and B. H. Hoang, "Wifi indoor positioning with genetic and machine learning autonomous war-driving scheme," *IJACSA*, vol. 13, no. 2.
- [26] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.
- [27] A. AL-Marakeby, "Multi-core processors for camera based omr," *ILCA*, vol. 68, no. 13, April 2013.
- [28] H. Atasoy, E. Yildirim, Y. Kutlu, and K. Tohma, "Webcam based real-time robust optical mark recognition," *IEEE*, November 2015.
- [29] P. D. Tinh, B. H. Hoang, and N. D. Cuong, "A genetic based indoor positioning algorithm using wi-fi received signal strength and motion data," *IAES IJAI*, vol. 12, no. 1.
- [30] S. Tiwari and S. Sahu, "A novel approach for the detection of omr sheet tampering using encrypted qr code," *IEEE*, 07 September 2015.
- [31] W. Zhang, "A fruit ripeness detection method using adapted deep learning-based approach," *IJACSA*, vol. 14, no. 9.
- [32] H. T. Ngoc, N. N. Vinh, N. T. Nguyen, and L.-D. Quach, "Efficient evaluation of slam methods and integration of human detection with yolo based on multiple optimization in ros2," *IJACSA*, vol. 14, no. 11.