International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 5, October 2023, pp. 5147~5155
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i5.pp5147-5155

Automatic optical inspection for detecting keycaps misplacement using Tesseract optical character recognition

Anisatul Munawaroh, Eko Rudiawan Jamzuri
Department of Electrical Engineering, Politeknik Negeri Batam, Batam, Indonesia

Article Info

Article history:

Received Jan 14, 2023
Revised Mar 28, 2023
Accepted Apr 7, 2023

Keywords:

Automatic optical inspection
Defect detection
Keycaps misplacement
Optical character recognition
Tesseract

ABSTRACT

This research study aims to develop automatic optical inspection (AOI) for detecting keycaps misplacement on the keyboard. The AOI hardware has been designed using an industrial camera with an additional mechanical jig and lighting system. Optical character recognition (OCR) using the Tesseract OCR engine is the proposed method to detect keycaps misplacement. In addition, captured images were cropped using a predefined region of interest (ROI) during the setup. Subsequently, the cropped ROIs were processed to acquire binary images. Furthermore, Tesseract processed these binary images to recognize the text on the keycaps. Keycaps misplacement could be identified by comparing the predicted text with the actual text on the golden sample. Experiments on 25 defects and 25 non-defected samples provided a classification accuracy of 97.34%, a precision of 100%, and a recall of 90.70%. Meanwhile, the character error rate (CER) obtained from the test on a total of 57 characters provided a performance of 10.53%. This outcome has implications for developing AOI for various keyboard products. In addition, the precision level of 100% signifies that the proposed method always offers correct results in detecting product defects. Such outcomes are critical in industrial applications to prevent defective products from circulating in the market.

Corresponding Author:

Eko Rudiawan Jamzuri
Department of Electrical Engineering, Politeknik Negeri Batam
St. Ahmad Yani, Teluk Tering, Batam Kota sub-district, Batam-29461, Indonesia
Email: ekorudiawan@polibatam.ac.id

1. INTRODUCTION

Quality control is an essential procedure to maintain product quality during the production process. Typically, a human operator or a machine performs this procedure. Human operators usually conduct this process by inspecting a product visually and manually utilizing additional tools. On the other hand, an automated optical inspection (AOI) machine can execute this inspection autonomously in a non-destructive manner [1]. More specifically, AOI integrates camera sensors, lenses, illumination lighting, and pattern recognition algorithms to evaluate product quality from a visual viewpoint. Sometimes AOI is combined with a robotic system to cover extensive inspection regions [2].

Many research studies encompass AOI applications on various manufacturing products. For instance, Fernandez *et al.* [3] investigated defects in fiber optic connectors, whereas Li *et al.* [4], Li *et al.* [5], and Rehman *et al.* [6] studied defects in printed circuit boards (PCBs). Moreover, several AOI-based research studies also included consumer products, such as bottles and cans by Rahman *et al.* [7] and Saad *et al.* [8], and keyboards by Huang and Ren [9] and Miao *et al.* [10], [11]. Each product carries a unique defect, so the proposed system must fine-tune the case of defects in each product. Moreover, the AOI hardware design must befit product features and requirements. Consequently, these design conditions push various proposed methodologies to address only specific defects, as discussed in a systematic review by Chen *et al.* [12].

Developing AOI systems for keyboard devices has remained the focus of extensive research studies. Some inspections related to this product include: checking the surface’s flatness, scratches, dents, and foreign objects, such as dust and particles. Huang and Ren [9] developed multilevel deep neural networks (MDNN) to detect keyboard surface abnormalities, where the proposed method employed two convolutional neural network (CNN) detection levels. The first level classified the image into five different defect categories. In the second level, two simultaneous CNN levels identified particles and dust. This multilayer approach’s primary benefit is rectifying misclassification cases at the first level, particularly for similar entities like dust and particles. On the diverse defect types, Miao *et al.* [10], [11] investigated the keycaps’ flatness defect using structured light imaging (SLI) approach. This methodology helps reduce the defects caused by mechanical motion in traditional three dimensional (3D) sensors. Le and Tu [13] investigated defects in similar products like remote controls in another research study. The investigated defects included lost buttons, missing characters, and inappropriate button positions. A pixel-matching approach could specify and categorize these defects. However, the study mainly concentrated on reducing computational time. Consequently, the proposed quick calibration method yielded a total processing time of 1,832 milliseconds during the experimental results.

Recent research studies indicate that automated defect detection on keyboards is a challenging task, albeit a well-studied area. However, many potential defects must still be investigated, namely keycaps misplacement. This misplacement potentially occurs if the size and color of the keycaps are the same. In addition, the potential for misplacement becomes pronounced if the operator performs the keyboard assembly process. As discussed above, Le and Tu [13] investigated the issue of misplacement defects; however, they only focused on minimizing the computational time. Therefore, the system’s accuracy in detecting misplacement defects has not been investigated yet. As a result, this study aims to develop an AOI approach to recognize keycaps misplacement and evaluate the AOI performance by quantifying accuracy, precision, and recall of the testing results.

The remainder of this paper has been organized as follows. First, section 2 describes the experimental setup, proposed hardware, and method to specify keycaps misplacement. Then, in section 3, we elaborated on the performance of the proposed method. Finally, in section 4, we conclude the overall findings of this research study.

2. METHOD

This section explains the methodology, materials, and processes utilized to conduct this research study. First, we discuss the experimental materials used in this investigation. Next, we elaborate on the AOI hardware design to identify keycap misplacement in materials. Finally, we propose the optical character recognition (OCR) technique using Tesseract OCR engine to detect the erroneous location of keycaps. In order to validate the success rate of our proposed hardware and software, we employed the testing and evaluation approach discussed in section 2.4.

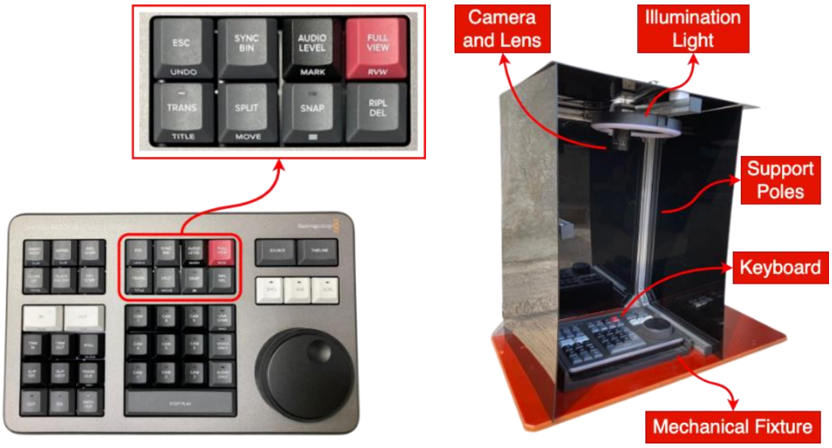
2.1. Experimental materials

Figure 1 illustrates the overall materials employed in this research, where Figure 1(a) is the experimental material and Figure 1(b) is the proposed AOI hardware. The detailed experimental material is depicted in Figure 1(a), which is a commercial video editing keyboard. This typical keyboard model is challenging for the study due to the numerous varieties of keycaps containing four unique keycap colors, white, gray, black, and red. Furthermore, the keycaps’ text is available in gray and white colors. In terms of keycap size, the keyboard has three distinct keycap sizes.

Various factual errors are probable throughout the keyboard assembly process. However, the critical defect is the misplacement of the keycaps, especially on the same size keycaps. Furthermore, if a human operator performs the keyboard assembly process, the likelihood of this defect increases substantially. In addition, text misprinting on the keycaps is another error occurring during production.

2.2. Hardware design

Figure 1 illustrates the AOI hardware utilized to inspect the experimental materials. As described in Figure 1(b), the hardware consists of mechanical fixtures, support poles, a camera and lens, and illumination lights. A mechanical jig is designed as a platform to hold the keyboard during the inspection, which minimizes product shifting, resulting in less optimal image capture. Furthermore, there is a support pole to hold the camera. The camera’s position can be mechanically adjusted to encompass the keyboard in the image-capturing area. We also installed an light-emitting diode (LED) illumination lamp adjacent to the camera to mitigate the influence of unsteady external lighting. The light beam was directed toward the product's surface, reducing shine and shadow. The camera utilized in this research study is a Basler acA1300-30 uC camera with a USB 3.0 interface directly linked to a PC. The maximum camera resolution is 1294×964 pixels, and its image capture speed is 30 frames per second (FPS).

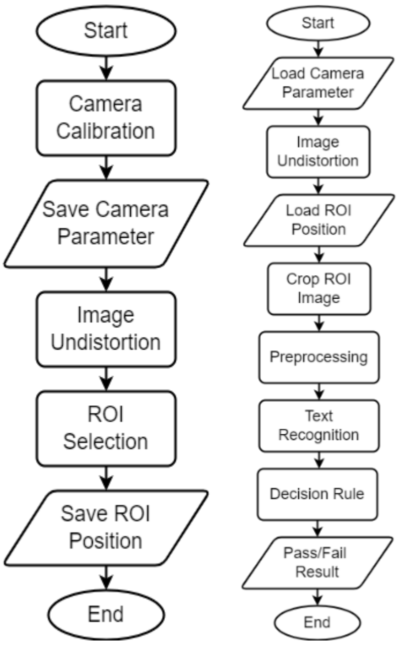


(a) (b)

Figure 1. Illustrates the overall materials employed in this research (a) keyboard model for research and (b) proposed AOI hardware

2.3. Detection system

Figure 2 illustrates the flowchart for the AOI detection system developed in this study, which comprises two primary processes: teaching and testing. As depicted in Figure 2(a), the teaching process specifies the region of interest (ROI) parameters conducted manually by expert users. Similarly, Figure 2(b) presents the testing process, which consists of loading ROI parameters and identifying the text on the ROI. Finally, the decision process compares the OCR output with the predefined text on the golden sample of the keyboard.



(a) (b)

Figure 2. Flowchart for AOI detection system on (a) teaching process and (b) testing process

At the initiation of the teaching process, the preprocessing stage minimizes distortion in the image. This stage requires a camera calibration process to extract essential parameters and intrinsic and extrinsic properties from the camera [14]. First, we performed camera calibration by recording 8×8 chessboard images from 10 perspectives. After acquiring the images, we identified the corner location on these images by exploring the sharp gradient between the black and white region. Subsequently, we derived the intrinsic camera features, distortion coefficient, rotation matrix, and translation vector from the corner coordinates. These parameters are signified by K and x in (1) and (2), given as:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$x = [k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3]^T \tag{2}$$

The K matrix in (1) formulates the camera's intrinsic properties, where c_x and c_y represent the optical centers and f_x and f_y indicate the focal lengths. The camera distortion coefficient is represented by the vector x in (2), which comprises $k_1, k_2, k_3, p_1,$ and p_2. Components k_1 through k_3 represent the radial distortion coefficient, whereas p_1 and p_2 specify the tangential coefficients. After attaining the K matrix and the vector x, these parameters are stored in a file and utilized in the subsequent picture deformation process.

After acquiring the K and x from the calibration phase, the following process is executed to rectify an obtained image from the camera. First, we refine K using a free scaling parameter α with a range between 0 and 1. If $\alpha = 0$, an undistorted picture with minimal undesired pixels is returned. However, if $\alpha = 1$, the process keeps all pixels on the corner with an additional black image. Moreover, this process yields a zone, which may be utilized to trim the image without distortion. This undistorted image is the primary input during the teaching and testing processes.

The ROI selection process-for evaluating the images inspection region-is performed after acquiring an undistorted image. To complete this process, the expert user must manually mark a text location within the image. Our ROI selection technique is not as adaptive as in [15] because the hardware design contains a mechanical jig to preclude object shifting, as illustrated in Figure 1(b). Therefore, the inspection area location is static and can be determined conveniently. The primary benefit of this technique is eliminating object detection processes that consume computational time. Nonetheless, the manual ROI selection must incorporate the text area in keycaps to facilitate text recognition. This process yields ROI parameters that consist of four box coordinates ($x_1, y_1, x_2,$ and y_2). These coordinates specify the position of the rectangle's upper left and lower right corners. After completing the ROI selection, these coordinates are stored in a file. Subsequently, ROI parameters are imported from this file during the testing process.

The product testing process involves multiple steps and concludes with a PASS/FAIL outcome. The process begins with opening files previously stored at the end of the teaching process. Next, the saved file's camera matrix, distortion coefficient, and ROI coordinates are loaded into variables. Then, these variables are utilized to rectify the camera image and acquire an undistorted image from the camera. This process is followed by the cropping operation from the ROI's predetermined coordinates. Finally, each ROI extracted undergoes a filtering process to generate binarized ROI pictures.

The preprocessing stage consists of several image-processing steps to obtain binary images. Initially, a color picture is transformed from RGB to grayscale using (3). In this, the variable *Y* specifies a grayscale image obtained from a colored image with red (*R*), green (*G*), and blue (*B*) channels. After the color transformation, we expanded the ROI picture by eight to create a larger image. In addition, a morphological filtering methodology employing the kernel K in (4) was utilized for this image. The filtering process initiates with four cycles of dilation, followed by a single erosion cycle. Finally, the image produced by morphological filtering is transformed into a black-and-white image using Otsu's thresholding technique. We chose this technique due to its proven ability to detect optical film product defects [16].

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{3}$$

$$K = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{4}$$

In the final step, the OCR algorithm is applied to this binary image to recognize the text printed on the keycaps. This method converts images of printed text into a computer- and human-readable format [17]. More specifically, we utilized Tesseract OCR engine given in [18], [19], which is a major Open-Source engine employed for extracting text. Many research studies widely utilize Tesseract OCR engine, and some of the applications include intelligent transportation systems for automatic license plate recognition (ALPR) [20], [21], text-to-brail code converter [22], and character recognition for non-English language documents [23], [24]. In particular, our research study utilized Tesseract 5.0 by assuming a single uniform text block in the keycap image. Furthermore, we employed the default OCR engine mode for configuration. Contrary to the technique employed in [25] which used character-level recognition-our recognition input is an image containing a text block. Thus, the recognition outcome is a text string or word printed in the image. After obtaining the prediction text from Tesseract OCR, we further process the text by trimming the white space character. This process results in one single-line text, although the actual keycaps text incorporates multiple lines. Using a single line of text makes the string comparison process more natural.

The text extracted by Tesseract OCR is then compared with the text given on the golden sample. The keycap is considered misplaced if the OCR-extracted text does not match the keycap text provided on the golden sample. In contrast, if the predicted text precisely matches the keycap text on the golden sample, the keycap is regarded to have the correct location. The final decision of product testing can be evaluated as follows. The keyboard is categorized as defective if the software detects one or more misplaced keycaps. Otherwise, the keyboard is considered a good product. Finally, these outcomes are displayed on the software user interface, including the misplaced keycap's location.

2.4. Testing and evaluation method

We performed the following tests and evaluations to validate the results of the proposed technique. Before evaluation, we established the image binarization settings. Subsequently, we manually adjusted the settings under constant illumination by trial-and-error technique to acquire the ideal OCR output. Next, we evaluated the test results using the classification metrics given by Vujovic [26]. In particular, the assessment metrics include accuracy, precision, and sensitivity/recall values. First, we computed the total number of true positives (*TP*), true negatives (*TN*), false positives (*FP*), and false negatives (*FN*) samples from the AOI prediction results. Among these metrics, TP specifies the number of predicted correct placement keycaps by the AOI when the ground truth is the correct location. Similarly, TN represents the number of predicted misplacement keycaps when the ground truth is misplacement. In contrast, FP signifies the number of predicted misplacement keycaps when the keycaps' positions are accurate, whereas FN determines the number of predicted correct placement keycaps when the actual keycaps are inaccurate. After acquiring these values, we calculated the accuracy, precision, and sensitivity/recall values using the (5) to (7).

$$Acc = \frac{TP+TN}{TP+FP+TN+FN} \tag{5}$$

$$Prec = \frac{TP}{TP+FP} \tag{6}$$

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

Because accuracy (*Acc*), precision (*Prec*), and sensitivity (*Recall*) metrics depend on OCR performance, we also analyzed the performance metrics of OCR. This performance can be quantified by computing the total number of substitutions, deletions, and insertions from the OCR output. Subsequently, these values can be utilized to calculate the OCR error rate. Drobac and Lindén [27], introduced these metrics for testing OCR performance while specifying a corpus of Finnish and Swedish historical newspapers and periodicals. The substitutions are represented by the total characters of OCR output that split from the actual characters. The deletions are defined by the number of characters missing from the text. Finally, the insertions are determined by the additional characters appearing on the OCR output. Consequently, the lower value of insertions, deletions, and substitutions specifies the excellent performance of the OCR. For the overall OCR performance, we quantify the character error rate (CER), which is formulated by (8), where the total number of substitutions is specified by S, the total number of deletions is represented by D, the total number of insertions denoted by I, and N is the number of characters in the actual text.

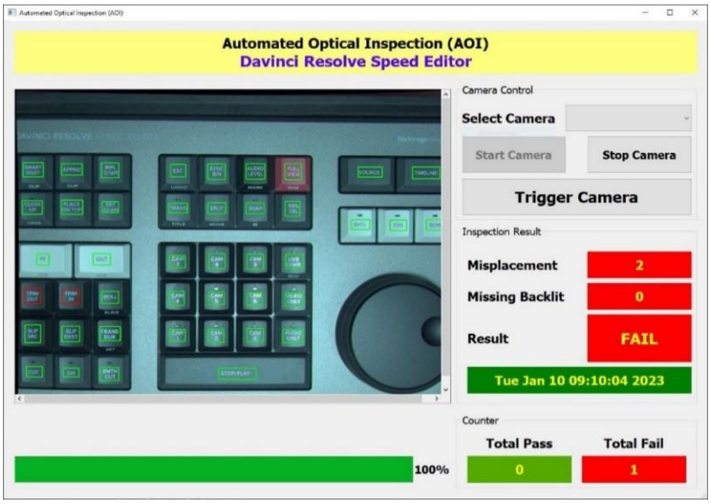
$$CER = \frac{S+D+I}{N} \tag{8}$$

3. RESULTS AND DISCUSSION

This section elaborates on our experimental results after implementing and executing our proposed approach. Firstly, the software visualization of the proposed AOI system is described. Next, we expand on the classification metrics gained during testing and evaluation. Lastly, we study the OCR output and its impact on the correctness of the system.

3.1. Visualization of AOI software

Figure 3 shows the user interface of the AOI software. The left side of the interface represents the image acquired by the camera for the inspection process. The interface's right side has specific controls for the inspection process. The start camera and stop camera buttons initiate and terminate image acquisition from the camera devices. Similarly, the trigger camera button begins the inspection process by grabbing an image and inspecting it. During the inspection, red or green bounding boxes appear on the keycaps to specify the correct or incorrect position, making it more straightforward for human operators to interpret the test results. The red box indicates the improper positioning of the keycap, whereas the green box specifies the appropriate positioning. For instance, in Figure 3, the software detected misplacement on keycaps "TRIMIN" and "TRIMOUT", which is swapped position. The AOI software can record misplacements and display them as "PASS" or "FAIL" in the result textbox. Furthermore, the product test outcomes are automatically customized to facilitate the operator to compute the excellent product's yield.



Automated Optical Inspection (AOI)
Davinci Resolve Speed Editor

Inspection Result
Misplacement: 2
Missing Backlit: 0
Result: FAIL
Tue Jan 10 09:10:04 2023

Counter
Total Pass: 0
Total Fail: 1

100%

Figure 3. Visualization of proposed AOI software

3.2. Test data evaluation

We performed the test by preparing 50 test keyboards comprising 25 accurate samples with correctly positioned keycaps and 25 faulty samples with improper keycap arrangements. In addition, with 2,100 keycap regions, Tesseract OCR was assessed for its ability to recognize text and detect placement problems. Table 1 presents the test results for the proposed AOI. The average accuracy score on the evaluation was 97.34%, the precision score was 100%, and the recall score was 90.70%. A 100% precision score implies that the proposed system never failed to detect misplaced keycaps. This outcome is crucial for the manufacturing industry since it ensures that poor-quality products are not delivered to customers. However, the proposed method has an average accuracy of 97.34%, indicating it occasionally misidentifies good samples. Thus, we can conclude that the AOI system still has the chance of incorrectly detecting perfectly positioned keycaps. According to Table 1, the keycaps with an accuracy of less than 90% contain the labels "*TRANSDUR*" (77.14% accurate), "*VIDEOONLY*" (80.00% accurate), "*SMTHCUT*" (82.86% accurate), and "*DIS*" (85.71% accurate). These results indicate that OCR misrecognition would cause the AOI system to fail to detect the precise position of keycaps.

We evaluated some keycaps with accuracy scores below 100% by assessing misrecognized OCR output. Table 2 presents the comparison results of the keycap's actual text to the text extracted by OCR. The substitution and insertion of characters result in several text recognition failures. From three rounds of trials on the specific keycaps with low accuracy, we noticed that the most caused misrecognition is character substitutions. In total, 16 characters were substituted from the OCR output. For instance, in keycaps "*APPND*", the 'D' was most frequently substituted with 'O'. Furthermore, letters with nonunique forms between capital and tiny letters were occasionally misidentified. For example, the character 'O' was swapped with 'o', the character 'U' was misrecognized as 'u', the character 'I' was identified with 'I', and vice versa.

This scenario frequently appeared when recognizing the "*VIDEOONLY*" keycap, consisting of four substitutions. On the other hand, some keycap texts are specified with additional characters. Consequently, we found that OCR output had an additional character "" at the end of string keycaps "*AUDIOONLY*" and "*SMTHCUT*". In addition, there were two insertions from the comprehensive test. In contrast, we did not find OCR output that contained deletions characters. This condition implies that the OCR output always results in more characters than the actual text's number of characters. From the test results on the text containing 30 samples with an accuracy of less than 100%, we achieved a CER of 10.53%.

Table 1. AOI performance metrics on the evaluation

Keycap	TP	TN	FP	FN	Accuracy (%)	Precision (%)	Recall (%)
SMARTINSRT	25	25	0	0	100.00	100.00	100.00
APPND	7	25	0	3	91.43	100.00	70.00
RIPLO/WR	25	25	0	0	100.00	100.00	100.00
ESC	25	25	0	0	100.00	100.00	100.00
SYNCBIN	25	25	0	0	100.00	100.00	100.00
AUDIOLEVEL	25	25	0	0	100.00	100.00	100.00
FULLVIEW	25	25	0	0	100.00	100.00	100.00
SOURCE	25	25	0	0	100.00	100.00	100.00
TIMELINE	25	25	0	0	100.00	100.00	100.00
CLOSEUP	25	25	0	0	100.00	100.00	100.00
PLACEONTOP	25	25	0	0	100.00	100.00	100.00
SRCO/WR	25	25	0	0	100.00	100.00	100.00
TRANS	25	25	0	0	100.00	100.00	100.00
SPLIT	25	25	0	0	100.00	100.00	100.00
SNAP	25	25	0	0	100.00	100.00	100.00
RIPLDEL	8	25	0	2	94.29	100.00	80.00
CAM7	25	25	0	0	100.00	100.00	100.00
CAM8	25	25	0	0	100.00	100.00	100.00
CAM9	25	25	0	0	100.00	100.00	100.00
LIVEO/WR	25	25	0	0	100.00	100.00	100.00
TRIMIN	25	25	0	0	100.00	100.00	100.00
TRIMOUT	25	25	0	0	100.00	100.00	100.00
ROLL	25	25	0	0	100.00	100.00	100.00
CAM4	25	25	0	0	100.00	100.00	100.00
CAM5	25	25	0	0	100.00	100.00	100.00
CAM6	25	25	0	0	100.00	100.00	100.00
VIDEOONLY	3	25	0	7	80.00	100.00	30.00
SLIPSRC	25	25	0	0	100.00	100.00	100.00
SLIPDEST	25	25	0	0	100.00	100.00	100.00
TRANSDUR	2	25	0	8	77.14	100.00	20.00
CAM1	7	25	0	3	91.43	100.00	70.00
CAM2	10	25	0	0	100.00	100.00	100.00
CAM3	10	25	0	0	100.00	100.00	100.00
AUDIOONLY	8	25	0	2	94.29	100.00	80.00
CUT	8	25	0	2	94.29	100.00	80.00
DIS	5	25	0	5	85.71	100.00	50.00
SMTHCUT	4	25	0	6	82.86	100.00	40.00
STOP/PLAY	25	25	0	0	100.00	100.00	100.00
IN	8	25	0	2	94.29	100.00	80.00
OUT	25	25	0	0	100.00	100.00	100.00
SHTL	25	25	0	0	100.00	100.00	100.00
JOG	25	25	0	0	100.00	100.00	100.00
SCRL	25	25	0	0	100.00	100.00	100.00
Average					97.34	100.00	90.70

Table 2. Tesseract OCR engine misrecognition results

Keycap text	Num.Chars	Tesseract OCR engine output			*S*	*D*	*I*
		Trial 1	Trial 2	Trial 3			
APPND	5	APPNO	APPNO	APPND	2	0	0
RIPLDEL	7	RIPLDEL	RIPLDEL	RIPLoEL	1	0	0
VIDEOONLY	9	VIDEOONLY	VIDEOoNLY	ViDeoONLY	4	0	0
TRANSDUR	8	TRANSOUR	TRANSDUR	TRANSOUR	2	0	0
CAM1	4	CAM1	CAM"	CAM1	1	0	0
AUDIOONLY	9	AUDIOoNLY'	AUDIOONLY	AUDIOONLY	1	0	1
CUT	3	CUT	CuT	CUT	1	0	0
DIS	3	DIS	DiS	DIS	1	0	0
SMTHCUT	7	SMTHCUT	SMTHCuT'	SMTHCuT	2	0	1
IN	2	IN	IN	IN	1	0	0
Total	57				16	0	2

4. CONCLUSION

This paper presents an AOI system to detect misplaced keycaps on the keyboard. We identified keycap placement errors by recognizing the text printed on the keycaps and comparing it with the actual text on the golden sample. The tests conducted on 25 accurate and 25 inaccurate (with misplacements) keyboard samples demonstrated that our proposed method produced an average accuracy rate of 97.34%, 100% precision, and 90.70% recall. Subsequently, we analyzed these results to evaluate the error rate in keycap character recognition with an accuracy below 100%. The experimental results of using ten different keycaps three times in each trial provided a CER value of 10.53%, with a total error of 16 characters exchanged and two additional characters. We observed that reducing the CER value as a system will impact the increase in accuracy. Currently, human experts obtain these metrics with optimal settings, where each expert may have different feelings toward parameter settings. Therefore, improving the proposed system by adding an optimization algorithm to explore the optimal configuration parameters generating maximum accuracy is necessary. In addition, it is essential to compare the CER performance of multiple OCR techniques to maximize the proposed system's accuracy level.

ACKNOWLEDGEMENTS

Politeknik Negeri Batam has partially funded this research through project-based learning (PBL) and final project schemes. The author would like to thank Politeknik Negeri Batam (Polibatam) and Barelang Robotics and Artificial Intelligence Lab (BRAIL) for offering the necessary resources and facilities for conducting this research study.

REFERENCES

[1] A. A. R. M. A. Ebayyeh and A. Mousavi, "A review and analysis of automatic optical inspection and quality monitoring methods in electronics industry," *IEEE Access*, vol. 8, pp. 183192–183271, 2020, doi: 10.1109/ACCESS.2020.3029127.

[2] T. Czimmermann *et al.*, "An autonomous robotic platform for manipulation and inspection of metallic surfaces in industry 4.0," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1691–1706, Jul. 2022, doi: 10.1109/TASE.2021.3122820.

[3] V. Fernandez, J. Chavez, and G. Kemper, "Device to evaluate cleanliness of fiber optic connectors using image processing and neural networks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 4, pp. 3093–3105, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3093-3105.

[4] D. Li, S. Li, and W. Yuan, "Positional deviation detection of silicone caps on FPCB," *Circuit World*, vol. 47, no. 1, pp. 23–29, Apr. 2020, doi: 10.1108/CW-07-2019-0067.

[5] J. Li, J. Gu, Z. Huang, and J. Wen, "Application research of improved YOLO V3 algorithm in PCB electronic component detection," *Applied Sciences*, vol. 9, no. 18, Sep. 2019, doi: 10.3390/app9183750.

[6] S. Ur Rehman, K. F. Thang, and N. S. Lai, "Automated PCB identification and defect-detection system (APIDS)," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 1, pp. 297–306, Feb. 2019, doi: 10.11591/ijece.v9i1.pp297-306.

[7] N. N. S. A. Rahman, N. M. Saad, and A. R. Abdullah, "Shape and level bottles detection using local standard deviation and hough transform," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 6, pp. 5032–5040, 2018, doi: 10.11591/ijece.v8i6.pp5032-5040.

[8] N. M. Saad, N. N. S. A. Rahman, A. R. Abdullah, and M. J. A. Latif, "Shape defect detection using local standard deviation and rule-based classifier for bottle quality inspection," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 8, no. 1, pp. 107–114, 2017, doi: 10.11591/ijeecs.v8.i1.pp107-114.

[9] X. Huang and J. Ren, "Quality control on manufacturing computer keyboards using multilevel deep neural networks," in *2020 IEEE 6th International Conference on Control Science and Systems Engineering (ICCSSE)*, Jul. 2020, pp. 184–188. doi: 10.1109/ICCSSE50399.2020.9171988.

[10] H. Miao, C. Xiao, M. Wei, and Y. Li, "Efficient measurement of key-cap flatness for computer keyboards with a multi-line structured light imaging approach," *IEEE Sensors Journal*, vol. 19, no. 21, pp. 10087–10098, Nov. 2019, doi: 10.1109/JSEN.2019.2928148.

[11] H. Miao, C. Xiao, Y. Li, S. Zou, and W. Huang, "Machine vision system for key-cap flatness measurement of computer keyboards," *Optical Engineering*, vol. 59, no. 3, Mar. 2020, doi: 10.1117/1.OE.59.3.033107.

[12] Z. Chen, J. Deng, Q. Zhu, H. Wang, and Y. Chen, "A systematic review of machine-vision-based leather surface defect inspection," *Electronics*, vol. 11, no. 15, Jul. 2022, doi: 10.3390/electronics11152383.

[13] H. Le and N. Tu, "A machine vision based automatic optical inspection system for detecting defects of rubber keypads of scanning machine," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 6, no. 18, Mar. 2019, doi: 10.4108/eai.28-3-2019.157121.

[14] A. K. Bedaka, S.-C. Lee, A. M. Mahmoud, Y.-S. Cheng, and C.-Y. Lin, "A camera-based position correction system for autonomous production line inspection," *Sensors*, vol. 21, no. 12, Jun. 2021, doi: 10.3390/s21124071.

[15] S. Wu, J. Liu, H. Luo, Z. Nie, H. Li, and J. Wu, "An automatic text region selection method on optical inspection for airborne instrument," in *2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS)*, Nov. 2021, pp. 504–508. doi: 10.1109/CCIS53392.2021.9754630.

[16] N. Tuyen Le, J.-W. Wang, M.-H. Shih, and C.-C. Wang, "Novel framework for optical film defect detection and classification," *IEEE Access*, vol. 8, pp. 60964–60978, 2020, doi: 10.1109/ACCESS.2020.2982250.

[17] S. S. R. Rizvi, A. Sagheer, K. Adnan, and A. Muhammad, "Optical character recognition system for nastalique urdu-like script languages using supervised learning," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 10, Sep. 2019, doi: 10.1142/S0218001419530045.

[18] R. Smith, "An overview of the Tesseract OCR engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, Sep. 2007, pp. 629–633. doi: 10.1109/ICDAR.2007.4376991.

[19] R. W. Smith, "History of the Tesseract OCR Engine: What worked and what didn't," in *SPIE Proceedings*, Feb. 2013. doi: 10.1117/12.2010051.

[20] Y. Gyu Jung and H. Wan Kim, "Design and implementation of lightweight vehicle license plate recognition module utilizing open CV and Tesseract OCR library," *International Journal of Engineering and Technology*, vol. 7, no. 3.3, Jun. 2018, doi: 10.14419/ijet.v7i2.33.14184.

[21] K. Thammarak, P. Kongkla, Y. Sirisathitkul, and S. Intakosum, "Comparative analysis of Tesseract and Google cloud vision for thai vehicle registration certificate," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, pp. 1849–1858, Apr. 2022, doi: 10.11591/ijece.v12i2.pp1849-1858.

[22] R. G. De Luna, "A Tesseract-based optical character recognition for a text-to-braille code conversion," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 10, no. 1, Feb. 2020, doi: 10.18517/ijaseit.10.1.9956.

[23] S. Banerjee, S. K. Singh, A. Das, and R. Bag, "Recognition of hindi and bengali handwritten and typed text from images using Tesseract on android platform," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1, pp. 3507–3516, Nov. 2019, doi: 10.35940/ijitee.A5252.119119.

[24] B. Wang, Y. W. Ma, and H. T. Hu, "Hybrid model for Chinese character recognition based on Tesseract-OCR," *International Journal of Internet Protocol Technology*, vol. 13, no. 2, 2020, doi: 10.1504/IJIPT.2020.106316.

[25] M. P. Muresan, P. A. Szabo, and S. Nedevschi, "Dot matrix OCR for bottle validity inspection," in *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Sep. 2019, pp. 395–401. doi: 10.1109/ICCP48234.2019.8959762.

[26] Ž. Đ. Vujovic, "Classification model evaluation metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021, doi: 10.14569/IJACSA.2021.0120670.

[27] S. Drobac and K. Lindén, "Optical character recognition with neural networks and post-correction with finite state methods," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 23, no. 4, pp. 279–295, 2020, doi: 10.1007/s10032-020-00359-9.

BIOGRAPHIES OF AUTHORS

Anisatul Munawaroh received a B.A.Sc. in Electrical Engineering from Politeknik Negeri Batam, Indonesia, in 2022. Her research interests include process control, image processing, and Computer Vision. Currently, she works as technical support in the PT. Blackmagic Design Manufacturing, Batam, Indonesia. She can be contacted at anisatulmunawaroh8@gmail.com.

Eko Rudiawan Jamzuri received a B.A.Sc. in Electrical Engineering from Bandung Institute of Technology in 2013 and earned an M.Sc. in Electrical Engineering from National Taiwan Normal University in 2020. Currently, he is a Lecturer at the Department of Electrical Engineering at Politeknik Negeri Batam. His research interests include humanoid robotics, computer vision, machine learning, and deep learning. He can be contacted at ekorudiawan@polibatam.ac.id.