APPLICATIONS OF
MODELLING AND SIMULATION
http://arqiipubl.com/ams eISSN 2600-8084 VOL 7, 2023, 38-48

Decoding Dot Peen Data Matrix Code with Deep Learning Capability for Product Traceability

Siu Hong Loh[1*], Peh Chiong Teh[1], Jia Jia Sim[2], Chian Kwang Tai[1], Kim Ho Yeap[1], Yu Jen Lee[1] and Ahmad Uzair Mazlan[1]

[1]Faculty of Engineering and Green Technology, Universiti Tunku Abdul Rahman, 31900 Kampar, Perak, Malaysia
[2]Faculty of Business and Finance, Universiti Tunku Abdul Rahman, 31900 Kampar, Perak, Malaysia

*Corresponding author: lohsh@utar.edu.my

Submitted 03 March 2023, 19 April 2023, Accepted 25 April 2023, Available online 26 April 2023.

Abstract: An approach for recognizing and decoding the industrial-based dot peen data matrix code is presented in this paper. Dot peen marking is a type of direct part marking (DPM). Due to the reduced contrast characteristic, it could be difficult to read a DPM code. Additionally, the readability of a DPM code may deteriorate over time due to partial degradation on the product surface. A deep-learning-based method using You-Only-Look-Once (YOLO) v5 model is proposed. Firstly, a large dataset of dot peen data matrix symbols was prepared to initiate the YOLOv5 model training. Image data augmentation was then applied to the training images to increase the size of the training dataset. The YOLOv5 model training was processed with a batch size of 16 and the epochs number of 60 due to its high accuracy (97.79%). All dot peen data matrix codes were detected accurately within one second, fulfilling our intention to design a high-speed reader for industrial-based dot peen data matrix. With ANOVA analysis, we observed that the brightness level and the camera distance significantly affect the decoding process. Additionally, our developed model can successfully decode a partially damage code if the level of damage is below 30%.

Keywords: Data matrix code; Deep learning; Direct part marking; Dot peen marking.

1. INTRODUCTION

Automatic product identification has been widely adopted in various sectors for more than two decades [1-3]. The data on component tracking is critical for people who manufacture, store, or transport goods along the supply chain since it is utilized to calculate manufacturing output, inventory management, revenue forecasts, and other company ventures. These barcodes are attached to goods through labels or as an integral part of the product packaging. With this, the product tracking could be implemented throughout the entire process life cycle, allowing it to be identified from starting until the end of the cycle. To achieve a complete life cycle traceability, manufacturers physically apply labelling on products' surface with 2D codes such as data matrix symbol. This allows the item identification to be made automatically throughout the production and supply chain activities [4].

A data matrix symbol is a 2D barcode used to store a large amount of data in a grid of black and white squares that can be placed in either a square or rectangular form [5]. Data matrix symbols are preferred in direct part marking (DPM) applications owing to their compact size, error correction capability, and data storage capacity [6]. These two-dimensional codes are applied physically on the component's part in various ways, including dot peen marking, inkjet, and laser etch, based on surface types [7-8]. Many manufacturers utilize traceability data to establish the tracking history of a part's production process for subsequent use in the procurement chain. A good product traceability will result in achieving success in enhancing quality, lowering cost, and streamlining operations. The manufacturer source, place of origin, production time and date, lot number, part, model and serial number, expiration date are among the kinds of data collected. All this information can be compiled and stored into a data matrix code and marked onto the part surface with DPM.

Over time, due to reduced contrast, changes in component surfaces, and partial degradation caused by environmental factors, DPM codes may be difficult to read [9]. An alternative method such as RFID could be adopted for product traceability and tracking [9]. Recently, successful DPM implementations have required the incorporation of computer vision system to achieve the necessary outcomes [10-11]. In this study, we focus on the readability of dot peen data matrix code. The dot peen marking is an intrusive DPM method using carbide needle to permanently engrave patterns on the product surface [12-13]. DPM created using dot peen marking is more robust as compared to other DPM techniques [7,13]. With relatively low equipment and maintenance cost, it is an economical option to be implemented in production process for product traceability

[14].

In this work, we used You-Only-Look-Once (YOLO) v5 model [15-16] together with the OpenCV library [17-18], and libdmtx package [19] to locate and decode the dot peen data matrix from the images. This code detection and decoding technique employ four distinct methods, namely localization, pre-processing, barcode decoding, and data displaying. There are several contributions of this work. Firstly, we implement a dot peen data matrix code reader that can recognize and decode industrial 2D dot peen data matrix symbols in real-time. The YOLOv5 model was employed as the object detection algorithm in this work with high accuracy of 97.79%. From the simulated results, all dot peen data matrix symbols were detected accurately within one second. Next, statistically, we have performed a comprehensive test using ANOVA technique and have successfully determined the factors that affect the performance of the decoding algorithm. As far as we are aware, this study is among the first to conduct a comprehensive statistical analysis on the developed decoding algorithm of dot peen data matrix code.

2. RELATED WORK

Using classical approaches and various conventional algorithm, previous studies have been conducted to detect and decode data matrix symbols. Liu et al. performed localization of data matrix code using Radon transform and boundary tracking [20]. To enhance the decoding performance under complex background, Huang et al. adopted bar code border fitting and finder pattern detection [21]. Combining grayscale and binary conversion with edge detection, a study to compare the decoding robustness of data matrix and QR codes was conducted [22]. Karrach and Pivarčiová examined various approaches of recognizing data matrix codes. From the experimental results, they observed that the adaptive thresholding methods were superior to the edge detection methods in terms of recognition rate [23-24].

Decoding a DPM's data matrix code could be a different scenario in comparison to decoding symbols printed on a plain paper. A DPM's data matrix code such as those printed on a metal surface could be difficult to decode due to reflective and uneven surface of the metal. Dai et al. proposed an improved Otsu algorithm for image binarization and Hough transform to overcome the challenges faced during the decoding process [25]. In another study to detect DPM's data matrix codes, Karrach and Pivarčiová proposed an adaptive thresholding method which could efficiently decode a laser-marked data matrix code for production engineering [26]. Most recently, Matuszczyk and Weichert presented an efficient method to decode a DPM's data matrix code created on polymer-based selective laser sintering (SLS) manufactured parts [11]. As the data matrix code was created on a surface which is completely white, the low-contrast condition was very challenging for the decoding process of the code. A deep learning (DL) approach was adopted for the localization and extraction process [11].

Using DL approach in barcode recognition could enhance the accuracy and speed of the detection and decoding process considerably [27]. Almeida et al. conducted a comprehensive evaluation on several types of Deep Neural Networks (DNNs) for the detection of data matrix landmarks [28]. By considering the speed of recalling and processing, and the average precision, it was concluded that YOLOv4 [29] outperformed Faster R-CNN [30] and Single-Shot Multibox Detector (SDD) [31]. In a work to develop a 1D barcode reader, Xiao and Ming proposed an approach to integrate DL and geometric methods, achieving a fast total process time of 19.2 ms [32]. Matuszczyk and Weichert have demonstrated that using DL approach, a high value of mean average precision (97.38%) for localization process can be achieved on low-contrast DPM's data matrix code detection [11].

There were limited studies on the detection of dot peen markings using DL approach previously. Hannes conducted a project to study optical character recognition (OCR) on dot peen markings using DL technique [33]. However, the object detection rate was only 60%. Dragičević et al. conducted a readability experiment on dot peen data matrix code printed on metal surfaces [13]. The decoding part was however performed using industrial PC application and the DL technique was not engaged in this work. Despite the importance and robustness of dot peen data matrix used in the manufacturing industry, there were limited number of studies conducted on the detection of this type of DPM. Therefore, a DL approach to detect and decode a dot peen data matrix symbol was presented in this study.

3. METHODOLOGY

We used a digital microscope to capture the dot peen symbol on the lead frame due to the smaller data matrix size. The digital microscope is suitable for industrial inspection due to its high magnification range (up to 1000x magnification). In this work, two main approaches were employed to read the 2D barcodes: locating the dot peen data matrix symbol on the image and decoding the data from the symbol. After obtaining an image, the primary step is to localize the dot peen data matrix symbol within the image. Once the symbol is localized with the detection algorithm, it must be decoded to acquire the data stored in the data matrix code. This process was accomplished by coding in Python language on Visual Studio Code. The YOLOv5 model was utilized in conjunction with the OpenCV library and libdmtx package to locate and decode the data matrix from the images. This suggested barcode detection and decoding technique employ four distinct methods: localization, pre-processing, barcode decoding, and data displaying.

3.1 Localization of Data Matrix

After inserting an image, a CNN-based detector is required to detect and locate the dot peen data matrix code. For carrying out the first stage of our concept, a dataset of dot peen data matrix symbols is required to train the YOLOv5 model. The data matrix detector was implemented by training the YOLOv5 model on the dot peen data matrix dataset using transfer learning on the Google Colab. Data can be gathered from two sources: primary source and secondary source. A data matrix dataset could be produced using practical symbols or software-generated symbols. However, software-generated data matrix symbols are not suitable to achieve our purpose. This is because the industrial dot peen data matrix symbol differs from the software-generated symbol in terms of the background region. Therefore, the data matrix dataset must consist of a large amount of dot peen data matrix symbol.

3.2 Pre-processing

Once the AI model correctly locates the data matrix symbol, it was cropped as a separate image to be further processed. The cropped image will be saved in a specific directory. The image processing technique was applied to the cropped image to provide a clearer image for decoding purposes. Firstly, the cropped image was read and transformed into grayscale representation with OpenCV function. Secondly, gaussian blur was applied to reduce noise in the image. Then, the blurred image will be further processed into a binary image which consists of black and white regions using the OpenCV threshold function. Simple thresholding, adaptive thresholding, and Otsu binarization [34] were employed to perform the image thresholding in this project. These thresholding methods are usually utilized in computer vision to process grayscale images. Thus, the image must first be transformed into a grayscale representation. By performing thresholding, the dot peen data matrix in the image can be easily decoded. The image was transformed to a ZXing and libdmtx compatible format, which eases the decoding process.

3.3 Barcode Decoding

In this phase, the decoding algorithms were employed to perform the decoding of the data matrix. The ZXing [35] and libdmtx [19] libraries were compared to select the optimal decoding algorithm. Both libraries are freely accessible and available for download. The libdmtx library was chosen as the decoding algorithm based on the trade-off between the decoding success rates and processing time. The decode function was imported from the pylibdmtx package at the beginning of the code. The decode function accomplished the decoding process when a data matrix was detected from the image. Several lines of code were written to develop a proper decoding process. The decrypted data was stored in a variable to be displayed.

3.4 Data Displaying

The next step after the decoding process was to display the decoded data. The data was displayed via the system graphical user interface (GUI) and Ubidot's dashboard [36]. Ubidots is a software platform that offers a safe and simple approach to develop IoT technologies. It can be used to transfer data to the cloud from any Internet-connected devices, trigger actions and alerts as well as visualize data. On the GUI, the decoded data was displayed in the text box with green color indicating the success of the decoding process, whereas the red color represented the failure of the decoding process. Additionally, the data matrix image and decoded data will be uploaded to Ubidots' cloud. Ubidots' cloud was utilized for storing raw data of data matrix, images in base64 format and image source. In the dashboard, two metric widgets were created to display the image source and data decoded from the data matrix. Meanwhile, the data matrix image was displayed with the HTML canvas widget. With this implementation, the user can perform history tracking on the previous data, which is suitable for the manufacturing industry.

3.5 YOLOv5 Object Detection

In this work, the YOLOv5 model is trained with dot peen data matrix datasets on Google Colab to detect and crop the data matrix for the decoding process. The steps to train a YOLOv5 model from scratch involve preparation of dataset, augmentation of the dataset, annotation of training images and training YOLOv5 model for custom objects. Figure 1 shows the flowchart of data matrix detection process.

4. RESULTS

4.1 The Adopted Training Model

During the model training process, we varied the number of epochs and batch size to observe the optimum combination. Firstly, the model was trained with the batch size fixed as 16 and the epochs number varied as 20, 40 and 60. Next, the model was trained with the epochs size fixed as 60 and the batch size varied as 20, 40 and 60. We observed that the best performing model was the retrained YOLOv5 model with the batch size of 16 and the epochs number of 60. The experimental results showed that the model achieved a high accuracy of 97.79%. Thus, the YOLOv5 model with the batch size of 16 and epochs number of 60 has been employed as the detection algorithm to recognize the dot peen data matrix symbol. The sample of prediction results for YOLOv5 model is depicted in Figure 2. The data matrix symbol in each image was drawn with a bounding box. The model was extracted to a local window with GUI. In the GUI, the model was tested with several images to examine the accuracy and the execution time, as illustrated in Figure 3.

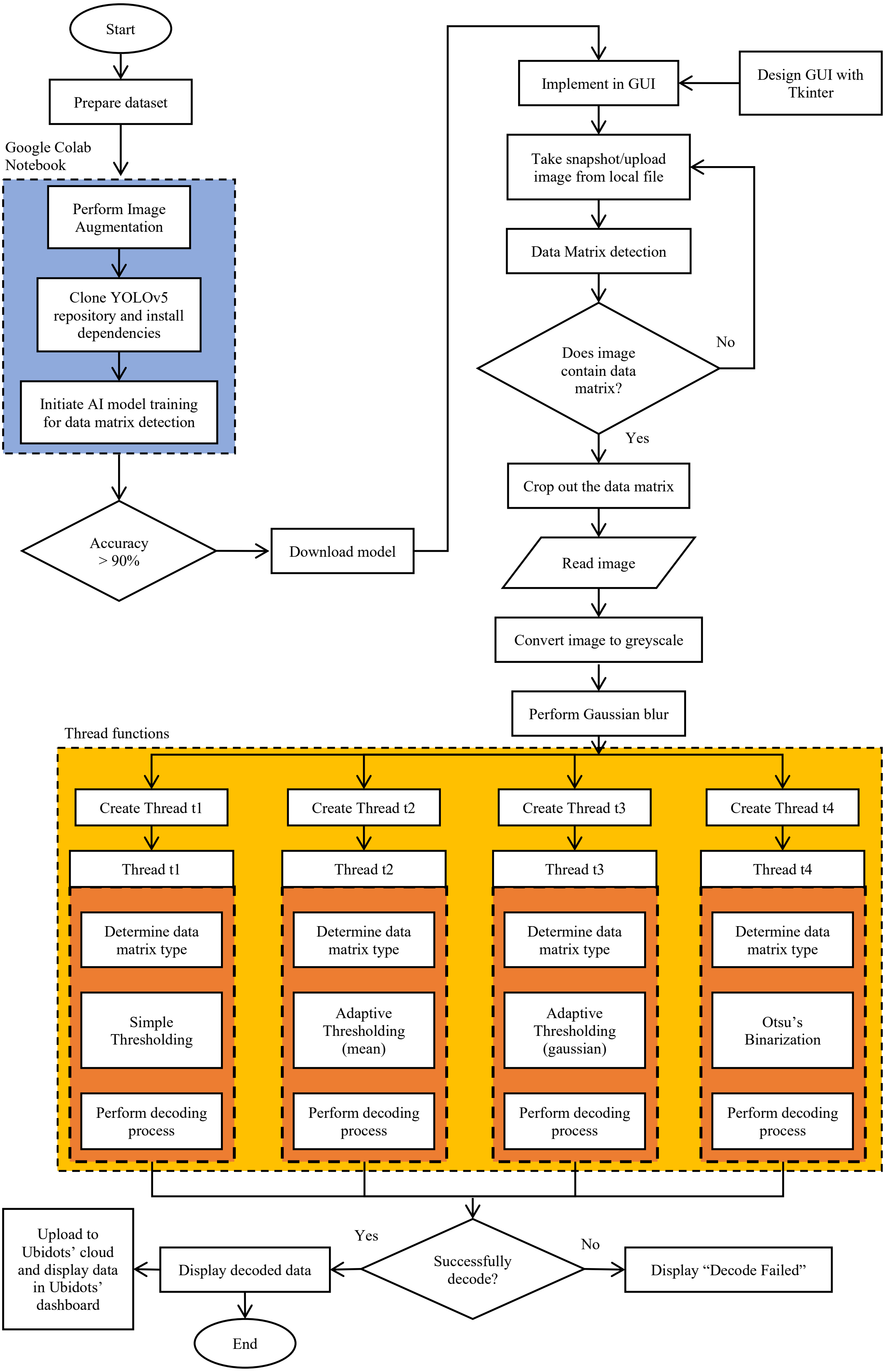


Figure 1. Flowchart of the data matrix detection process

Figure 2. Sample of prediction results for YOLOv5 model

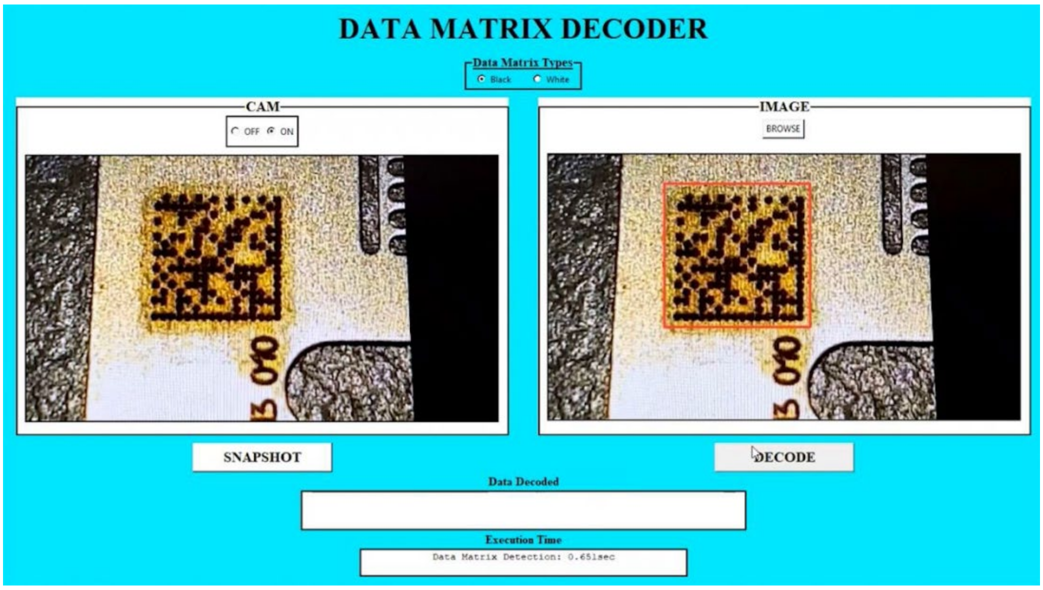


Figure 3. The GUI of real-time data matrix detection

4.2 Decoding Results with Different Libraries

We evaluated the decoding outcomes with ZXing [35] and libdmtx [19] libraries. A total of 25 test images were used to compare the decoding results and the process time of both libraries. The data analysis for ZXing and libdmtx libraries are summarized in Table 1. The libdmtx decoding library had a higher successful decoding rate as compared to the ZXing decoding library (100% versus 72%). However, the average process time of the libdmtx library was slightly higher by comparison (0.664s versus 0.549s). the main drawback of the ZXing library is that it was less effective when decoding the industrial-based data matrix symbol. From Table 1, seven failed test cases have caused the overall performance of the model implemented using the ZXing library to decline. Based on our test results, for dot peen data matrix symbols printed on a complex background like the metal surface, it might take a longer time for the libdmtx to isolate the foreground of the data matrix from the background region. Since the success rate of the decoding process is prioritized, the libdmtx decoding library was adopted.

4.3 ANOVA Analysis

The performance of the data decoding process of the dot peen data matrix symbols were affected by several factors. We conducted one-way Analysis of Variance (ANOVA) [37], a statistical technique to obtain the optimal condition for the data decoding process. Four different factors were considered and evaluated in this analysis, namely: lighting conditions, orientation of the dot-peen based data matrix symbol, camera distance and partially damaged code.

4.3.1 Lighting Conditions

Firstly, the brightness level was adjusted to study its effect on the decoding process. The other variables such as data matrix orientation, camera distance, and partially dam-aged percentage were held constant. Each condition was tested 10 times and the results are shown in Table 2. The ten trials were conducted by manually capturing the dot peen data matrix on the lead frame with a digital microscope. The LED light on the digital microscope was adjusted to control the brightness level on the lead frame, as shown in Figure 4.

The summary of the ANOVA analysis is shown in Table 3. The F ratio value and the p-value are 1023.885 and 0.000, respectively, indicating that there is a statistically significant difference between groups with varied lighting conditions. Therefore, the brightness level is one of the factors affecting the decoding process, and the decoding algorithm will perform better at a high brightness level.

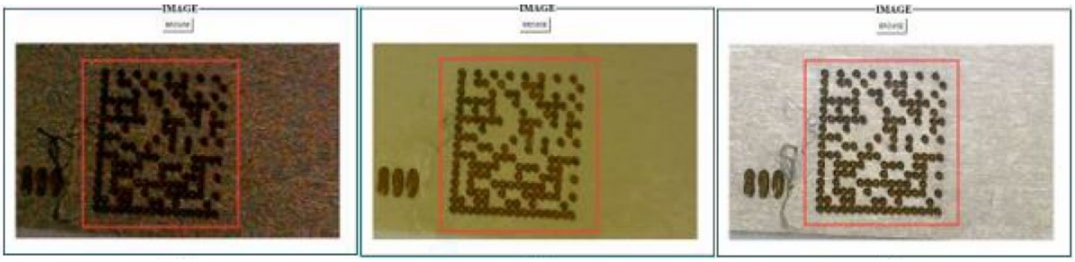
Table 1. Data analysis for ZXing and libdmtx libraries

	ZXing	**libdmtx**
Decoding Results	No. of test images = 25 No. of successful decode = 18 No. of failed decode = 7 Success rate: 18/25*100 = 72%	No. of test images = 25 No. of successful decode = 25 No. of failed decode = 0 Success rate: 25/25*100 = 100%
Average Process Time	Average process time = 0.549s	Average process time = 0.664s

Table 2. Decode status and processing time for each trial under different lighting conditions

No. of trials	**Low brightness**		**Medium brightness**		**High brightness**	
	Status	**Process Time (s)**	**Status**	**Process Time (s)**	**Status**	**Process Time (s)**
1	Fail	-	Success	0.559	Success	0.181
2	Fail	-	Success	0.578	Success	0.192
3	Fail	-	Success	0.467	Success	0.232
4	Fail	-	Success	0.634	Success	0.189
5	Success	2.102	Success	0.601	Success	0.222
6	Fail	-	Success	0.498	Success	0.251
7	Fail	-	Success	0.521	Success	0.161
8	Success	2.322	Success	0.558	Success	0.196
9	Fail	-	Success	0.467	Success	0.208
10	Fail	-	Success	0.611	Success	0.221
	SR = 20%	Mean = 2.212	SR = 100%	Mean = 0.549	SR = 100%	Mean = 0.205

SR: Success rate.



(a) (b) (c)

Figure 4. Brightness level of the dot peen data matrix symbol: (a) low; (b) medium; (c) high

Table 3. Summary of ANOVA analysis for decoding process under different lighting conditions

Source	**DF**	**SS**	**MS**	**F**	**p-value**
Between groups	2	6.712	3.356	1023.885	0.000
Within Groups	19	0.062	0.003		
Total	21	6.774			

DF: Degrees of freedom; SS: Sum of squares; MS: Mean square

4.3.2 Orientation of the Dot Peen Data Matrix Symbol

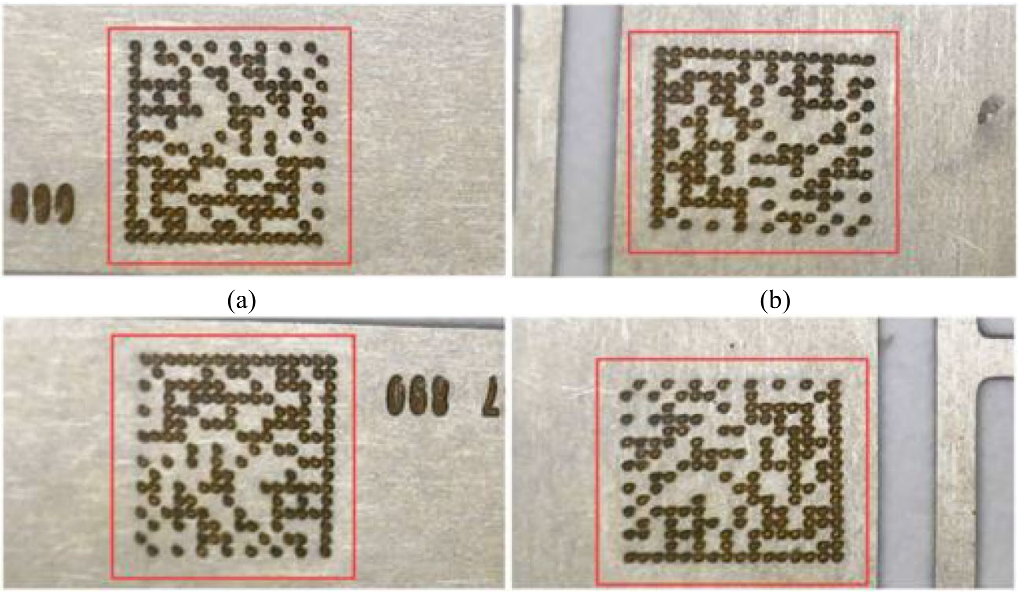
Secondly, an experiment was conducted to study the effect of the data matrix orientation on the decoding process. The other variables such as brightness level, camera distance and partially damaged percentage were fixed at high brightness level, close distance and 0% damage, respectively. There were four evaluated orientations: rotation angle of 0°, 90°, 180° and 270°. Each condition was tested 10 times and the results are shown in Table 4. The ten trials were conducted by manually capturing the dot peen data matrix on the lead frame with a digital microscope. The dot peen data matrix was rotated clockwise, 90° at a time, as shown in Figure 5.

From Table 5, the F ratio value and the p-value are 0.028 and 0.994, respectively, im-plying a low variation of mean values among groups with different orientation. Therefore, it could be concluded that the data matrix orientation does not affect the performance of the decoding process.

Table 4. Decode status and processing time for each trial under different orientations of the dot peen data matrix symbol

No. of trials	Orientation							
	0°		90°		180°		270°	
	Status	Process Time (s)	Status	Process Time (s)	Status	Process Time (s)	Status	Process Time (s)
1	Success	0.186	Success	0.183	Success	0.164	Success	0.194
2	Success	0.200	Success	0.176	Success	0.174	Success	0.173
3	Success	0.166	Success	0.221	Success	0.205	Success	0.170
4	Success	0.221	Success	0.204	Success	0.187	Success	0.223
5	Success	0.179	Success	0.182	Success	0.249	Success	0.196
6	Success	0.203	Success	0.230	Success	0.167	Success	0.161
7	Success	0.227	Success	0.223	Success	0.203	Success	0.219
8	Success	0.229	Success	0.193	Success	0.243	Success	0.199
9	Success	0.176	Success	0.206	Success	0.198	Success	0.236
10	Success	0.183	Success	0.170	Success	0.212	Success	0.215
	SR = 100%	Mean = 0.204	SR = 100%	Mean = 0.199	SR = 100%	Mean = 0.200	SR = 100%	Mean = 0.199

SR: Success rate.



(a) (b) (c) (d)

Figure 5. Orientation of the dot peen data matrix symbol: (a) 0°; (b) 90°; (c) 180°; (d) 270°

Table 5. Summary of ANOVA analysis for decoding process of data matrix at different orientation

Source	DF	SS	MS	F	p-value
Between groups	3	0.000	0.000	0.028	0.994
Within Groups	36	0.022	0.001		
Total	39	0.022			

DF: Degrees of freedom; SS: Sum of squares; MS: Mean square

4.3.3 Camera Distance

The third experiment was conducted to study the effect of the camera distance on the decoding process. Other variables like brightness level, data matrix orientation and partially damaged percentage were fixed at high brightness level, 0° rotation and 0% damage, respectively. The three different camera distances evaluated were close distance, medium distance and far distance. Each condition was tested 10 times and the results are shown in Table 6. The size of the data matrix in the field of view was manually varied by adjusting the distance of the digital microscope from the lead frame, as illustrated in Figure 6.

The F ratio value and the p-value are 931.082 and 0.000, respectively, as shown in Table 7. These values indicate that there is a statistically significant difference between groups with varying camera distances. Therefore, the camera distance from the dot peen data matrix symbol is one of the factors affecting the decoding process. It is observed from the process time that the decoding algorithm will perform better at a closer camera distance.

Table 6. Decode status and processing time for each trial under different camera distances

No. of trials	Close distance		Medium distance		Far distance	
	Status	Process Time (s)	Status	Process Time (s)	Status	Process Time (s)
1	Success	0.200	Success	0.563	Fail	-
2	Success	0.169	Success	0.693	Fail	-
3	Success	0.204	Success	0.664	Fail	-
4	Success	0.192	Success	0.637	Fail	-
5	Success	0.159	Success	0.540	Success	1.813
6	Success	0.150	Success	0.626	Fail	-
7	Success	0.200	Success	0.535	Success	1.709
8	Success	0.187	Fail	-	Fail	-
9	Success	0.179	Success	0.486	Fail	-
10	Success	0.227	Success	0.493	Success	1.846
	SR = 100%	Mean = 0.187	SR = 90%	Mean = 0.582	SR = 30%	Mean = 1.789

SR: Success rate.

(a) (b) (c)

Figure 6. Camera distance of the dot peen data matrix symbol: (a) close; (b) medium; (c) far

Table 7. Summary of ANOVA analysis for decoding process under different camera distances

Source	DF	SS	MS	F	p-value
Between groups	2	5.930	2.965	931.082	0.000
Within Groups	19	0.061	0.003		
Total	21	5.991			

DF: Degrees of freedom; SS: Sum of squares; MS: Mean square

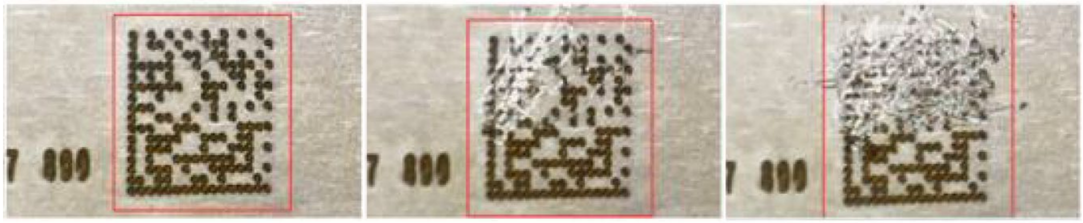
Table 8. Decode status and processing time for each trial under different damage conditions

No. of trials	0% damage		30% damage		50% damage	
	Status	Process Time (s)	Status	Process Time (s)	Status	Process Time (s)
1	Success	0.227	Success	0.287	Fail	-
2	Success	0.196	Success	0.254	Fail	-
3	Success	0.164	Success	0.222	Fail	-
4	Success	0.199	Success	0.251	Fail	-
5	Success	0.166	Success	0.288	Fail	-
6	Success	0.216	Success	0.224	Fail	-
7	Success	0.222	Success	0.280	Fail	-
8	Success	0.210	Success	0.280	Fail	-
9	Success	0.216	Success	0.254	Fail	-
10	Success	0.179	Success	0.221	Fail	-
	SR = 100%	Mean = 0.200	SR = 100%	Mean = 0.256	SR = 0%	-

SR: Success rate.

4.3.3 Partially Damage Code

The final experiment was conducted to study the effect of the partially damaged code on the decoding process. The other variables such as brightness level, data matrix orientation and camera distance were set at high brightness level, 0° rotation and close distance, respectively. Three conditions with different level of damage on the data matrix code were evaluated. This includes the conditions of the code with 0% level of damage, 30% level of damage and 50% level of damage. Each condition was tested 10 times and the results are shown in Table 8. As illustrated in Figure 7, the damage was intentionally applied to the data matrix code by scratching the symbol surface with a sharp metal.



(a) (b) (c)

Figure 7. Level of damage of the dot peen data matrix symbol: (a) 0% damage; (b) 30% damage; (c) 50% damage

Table 9. Summary of ANOVA analysis for decoding process under different damage conditions

Source	DF	SS	MS	F	p-value
Between groups	1	0.016	0.016	25.559	0.000
Within Groups	18	0.011	0.001		
Total	19	0.027			

DF: Degrees of freedom; SS: Sum of squares; MS: Mean square

The ANOVA analysis is summarized in Table 9. The analysis was performed on only two groups in this experiment because the third group does not contain any valid data. The F ratio value and the p-value are 25.558 and 0.000, respectively, indicating that partially damaged dot peen data matrix code is one of the factors affecting the decoding process.

5. DISCUSSION

Summarizing our results, the best performing developed YOLOv5 model (batch size of 16 and the epochs number of 60) was able to achieve a high accuracy of 97.79%. We have considered different conditions that might affect the decoding results of the dot peen data matrix code. This includes lighting conditions, orientation of the data matrix code, camera distance and partially damaged code. Based on the results discussed in section 3.3.1, the decoding algorithm performed better at the high brightness level condition with the fastest average processing time of 0.205 seconds (Table 2). For poor lighting condition, the dot peen data matrix can be decoded at some point, but it took longer time to process the image, as noticed during the fifth and eighth trials (Table 2). Since the dot peen data matrix is formed by dark and light modules, increased brightness will enhance the contrast level between the modules, and hence easing the decoding process. For different orientation of the data matrix code, a decoding success rate of 100% was achieved (Table 4). The decoding algorithm determines the data matrix's position and orientation by defining the finder pattern. Therefore, the libdmtx library was still able to decode the data matrix no matter how much the data matrix was being rotated. Additionally, the times to decode the data matrix at different orientations were approximately the same, implying that the data matrix orientation will not affect the decoding process.

For variation of camera distance, an average processing time of 0.187 seconds was required to decode the dot peen data matrix at a close camera distance (Table 6). However, the average processing time was extended to 0.582 seconds at a medium camera distance. The success rate of the decoding process dropped to only 30% at far camera distance (Table 6). Therefore, the variation of camera distance significantly affected the decoding process. For a partially damaged dot peen data matrix code, the developed model was still able to achieve decoding rate of 100% even with 30% level of damage on the code (Table 8). Data matrix employs Reed-Solomon codes for error correction and data restoration, enabling it to recover from damaged messages. However, our test indicates that it was not possible to decode the dot peen data matrix code with 50% level of damage.

Figure 8 illustrates the box plot of the four factors (lighting conditions, orientation of the data matrix code, camera distance and partially damaged code) that might affect the decoding results of the dot peen data matrix code. From the box plot, the first (lighting conditions) and the third (camera distance) groups have larger dispersion (taller boxes) as compared to the second (orientation of the data matrix code) and the fourth (partially damaged code) groups. Large data dispersion here indicates that there was a significant variation in the processing time of the decoding process. Small data dispersion is preferred as the processing time for each trial under different conditions is rather similar. Therefore, it can be concluded that the brightness level and the camera distance affect the decoding process significantly. For partially damaged code, as long as the level of damage is below 30%, the decoding rate of 100% with reasonable processing time can be achieved. In contrast, the lead frame which contains the dot peen data matrix code can be rotated randomly during the decoding process because the data matrix orientation will not affect the decoding process.

The results from our analysis agree with previous studies, particularly for those related to DPM's data matrix code. For the data matrix code printed on a SLS part, Matuszczyk and Weichert observed that the light conditions and camera position are important during the detection process [11]. For a dot peen data matrix code printed on a metal surface, Dragičević et al. concluded that the readability with smartphone camera was not reliable due to the lack of proper light source [13]. Hannes observed that the variation of lightning and shadows will affect the detection results of the dot peen markings [33].

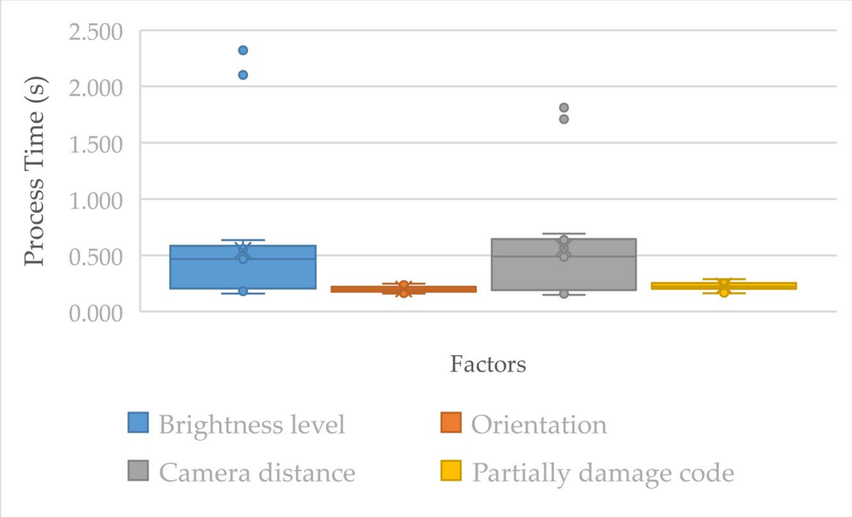


Figure 8. Box plot for ANOVA analysis

6. CONCLUSION

In this paper, an approach for recognizing and decoding the industrial-based dot peen data matrix code is presented. A detection model was successfully implemented to improve the reading speed of the dot peen data matrix code reader. YOLOv5 model was employed as the object detection algorithm in this work due to its fast detection speed and accurate results. Training a CNN-based model with a large dataset can produce a high-performance model. The YOLOv5 model training was processed with a batch size of 16 and the epochs number of 60 due to its high accuracy of 97.79%. From the simulated results, all the dot peen data matrix were detected accurately within one second, fulfilling our intention to design a high-speed code reader capable of detecting and decoding the industrial dot peen data matrix symbols.

In addition, several factors were considered during the data matrix decoding process, including brightness level, data matrix orientation, camera distance, and partially dam-aged code. With ANOVA analysis, it could be concluded that the brightness level and the camera distance are two factors that significantly affect the decoding process. In contrast, the orientation of the dot peen data matrix code does not affect the decoding process. The optimal conditions to decode a dot peen data matrix symbol are high brightness level and a close camera distance. It was demonstrated that our developed model can successfully decode a partially damage code if the level of damage is below 30%. Lastly, the data decoded from the dot peen data matrix was successfully uploaded to Ubidot's cloud for data visualization purposes.

REFERENCES

[1] T. Sun and D. Zhou, Automatic identification technology - Application of two-dimensional code, *2011 IEEE International Conference on Automation and Logistics (ICAL)*, Chongqing, China, 2011, 164-168.
[2] B. Santra and D. P. Mukherjee, A comprehensive survey on computer vision based approaches for automatic identification of products in retail store, *Image and Vision Computing*, 86, 2019, 45-63.
[3] M. Liukkonen and T. -N. Tsai, Toward decentralized intelligence in manufacturing: Recent trends in automatic identification of things, *International Journal of Advanced Manufacturing Technology*, 87(9-12), 2016, 2509-2531.
[4] R. Schuitemaker and X. Xu, Product traceability in manufacturing: A technical review, *Procedia CIRP*, 93, 2020, 700-705.
[5] C. Plain-Jones, Data matrix identification, *Sensor Review*, 15(1), 1995, 12-15.
[6] E. Ahearne, Engineering the surface for direct part marking (DPM), *CIRP Journal of Manufacturing Science and Technology*, 29, 2020, 1-10.
[7] C. Moss, S. Chakrabarti and D. W. Scott, Parts quality management: Direct part marking of data matrix symbol for mission assurance, *2013 IEEE Aerospace Conference*, Big Sky, MT, 2013, 1-12.
[8] K. Czerwińska and A. Pacana, Analysis of the implementation of the identification system for directly marked parts - DataMatrix code, *Production Engineering Archives*, 23(23), 2019, 22-26.
[9] D. M. Segura Velandia, N. Kaur, W. G. Whittow, P. P. Conway and A. A. West, Towards industrial internet of things: Crankshaft monitoring, traceability and tracking using RFID, *Robotics and Computer-Integrated Manufacturing*, 41, 2016, 66-77.
[10] F. A. Eyahanyo, *Computer Vision Reading on Stickers and Direct Part Marking on Horticultural Products: Challenges and Possible Solutions*, Ph.D. Thesis, Leibniz Universität Hannover, Hanover, Germany, 2020.
[11] D. Matuszczyk and F. Weichert, Reading direct-part marking data matrix code in the context of polymer-based additive manufacturing, *Sensors*, 23(3), 2023, 1619.
[12] Dot Peen Marking, https://www.mecco.com/application-dot-peen (accessed Mar. 03, 2023).
[13] D. Dragičević, S. Tegeltija, G. Ostojić, S. Stankovski and M. Lazarević, Reliability of dot peen marking in product traceability, *International Journal of Industrial Engineering and Management,* 8(2), 2017, 71-76.

[14] F. Deng, R. Li, S. Klan and W. Volk, Comparative evaluation of marking methods on cast parts of Al–Si Alloy with image processing, *International Journal of Metalcasting*, 16(3), 2022, 1122-1139.
[15] G. Jocher, YOLOv5 by Ultralytics, https://github.com/ultralytics/yolov5, 2020 (accessed Mar. 03, 2023).
[16] L. Gur Arie, The practical guide for object detection with YOLOv5 algorithm, *Medium*, 2023. https://towardsdatascience.com/the-practical-guide-for-object-detection-with-yolov5-algorithm-74c04aac4843 (accessed Mar. 03, 2023).
[17] OpenCV, https://opencv.org/ (accessed Mar. 03, 2023).
[18] I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, A brief introduction to OpenCV, *Proceedings of the 35th International Convention MIPRO*, 2012, 1725-1730.
[19] dmtx/libdmtx, https://github.com/dmtx/libdmtx, 2023 (accessed Feb. 26, 2023).
[20] F. Liu, A. Liu, M. Wang and Z. Yang, Robust and fast localization algorithm for data matrix barcode, *International Conference on Optoelectronics and Image Processing*, Hainan, China, 2010, 356-359.
[21] Q. Huang, W. -S. Chen, X. -Y. Huang and Y. -Y. Zhu, Data matrix code location based on finder pattern detection and bar code border fitting, *Mathematical Problems in Engineering*, 2012, 1-13.
[22] R. Kulshreshtha, A. Kamboj and S. Singh, Decoding robustness performance comparison for QR and data matrix code, *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*, Coimbatore, India, 2012.
[23] L. Karrach and E. Pivarciova, The analyse of the various methods for location of data matrix codes in images, *ELEKTRO*, Mikulov, Czech Republic, 2018, 1-6.
[24] L. Karrach and E. Pivarčiová, Comparative study of data matrix codes localization and recognition methods, *Journal of Imaging*, 7(9), 2021, 163.
[25] Y. Dai, L. Liu, W. Song, C. Du and X. Zhao, The realization of identification method for datamatrix code, *International Conference on Progress in Informatics and Computing (PIC)*, Nanjing, China, 2017, 410-414.
[26] L. Karrach and E. Pivarčiová, Options to Use data matrix codes in production engineering, *Management Systems in Production Engineering*, 26(4), 2018, 231-236.
[27] R. Wudhikarn, P. Charoenkwan and K. Malang, Deep learning in barcode recognition: A systematic literature review, *IEEE Access*, 10, 2022, 8049-8072.
[28] T. Almeida, V. Santos, O. M. Mozos and B. Lourenço, Comparative analysis of deep neural networks for the detection and decoding of data matrix landmarks in cluttered indoor environments, *Journal of Intelligent & Robotic Systems*, 103, 2021, 13.
[29] A. Bochkovskiy, C. -Y. Wang and H. -Y. M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, arXiv: 2004.10934, 2020.
[30] S. Ren, K. He, R. Girshick and J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, arXiv: 1506.01496, 2015.
[31] W. Liu et al., SSD: Single Shot MultiBox Detector, arXiv: 1512.02325, 2015.
[32] Y. Xiao and Z. Ming, 1D barcode detection via integrated deep-learning and geometric approach, *Applied Sciences*, 9(16), 2019, 3268.
[33] E. Hannes, *OCR of Dot Peen Markings: With Deep Learning and Image Analysis*, Master Thesis, Luleå University of Technology, Sweden, 2018.
[34] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man and Cybernatics*, 9(1), 1079, 62-66.
[35] zxing/zxing, https://github.com/zxing/zxing, 2023 (accessed Feb. 26, 2023).
[36] Ubidots - Powerful but simple Industrial IoT, https://ubidots.com/ (accessed Mar. 07, 2023).
[37] R. M. Heiberger and E. Neuwirth, One-Way ANOVA, in *R Through Excel*, New York: Springer New York, 2009, 165-191.