MOTIVECTION
Journal of Mechanical Electrical and Industrial Engineering

Volume : 5, Number : 2, 2023
ISSN : 2655 – 7215 [printed] ; 2685-2098 [online]
DOI : 10.46574/motivection.v5i2.230

Computer Vision Implementation in Scratch Inspection and Color Detection on The Car Roof Surface

Wahyu A. Candra[1]*, Adhitya S. Sunarya[1], Wening S. Saraswati[1]

Abstract

The automotive industry in Indonesia is one of the important pillars in the manufacturing sector. High speed productivity and high-level quality of their products are required for increasing the company value. The car roof inspection is still in low speed and inadequate quality control method for most of the car company's manufacturing production line. This inspection process is highly dependent to human expertise and skill of the assembly operators. This study purposes to increase the productivity in the inspection line by eliminate human error using computer vision. This research aims to find an automatic and accurate method for visual inspection of the color quality and scratch defect on car roof surfaces, using the data generated from the standardized database QR Code. The proposed solution uses MobileNet-SSD method of computer vision in recognizing scratch defects, while color detection of the roof is attained through hue saturation method. The study employs data capture from a camera and compares the amount of object inspected in real time. The scratch inspection with 70 image samples and 4000 training steps results in a 3.33% error with color detection and inspection success rate of 80%.

Keywords
Computer Vision, Surface Inspection, Color Detection, MobileNet-SSD

[1] Bandung Polytechnic of Manufacturing, Dept. Of Mechatronics and Automation Engineering
Jl. Kanayakan No.21, Dago, Bandung, Jawa Barat - 40135

* wahyu@ae.polman-bandung.ac.id

Submitted : March 26, 2023. Accepted : April 12, 2023. Published : April 13, 2023.

INTRODUCTION

The automotive industry in Indonesia is one of the important pillars in the manufacturing sector, due to the large number of car manufacturing plants operating in the country. The automotive industry is an indicator of economic growth and a driving force for other industries. The development of the car industry in Indonesia is increasingly advanced with many variations and innovations issued by well-known car manufacturers. Currently, car sales in Indonesia have increased. Indonesia has an outstanding profile in the automotive industry in the Southeast Asia region. In 2015, the total quantity sales of four-wheeled vehicles reached 1.2 million units [1].

In this era of globalization, the car companies continue to improve quality and quantity, either by controlling the quality directly to the number of products produced or by carrying out routine activities that analyze the quality control procedure. The quality of a product is one of the criteria that customers consider in choosing a product. Product quality is also an important indicator for companies to be able to stand out in the midst of an intense competition in the industrial world[2], [3].

By advanced science and technology, manual devices and human operations are transitioning to being automated and digitized. The improvement to increase productivity and quality are overwhelmed by automating the manufacturing process either by single machine or multiple agents or machines which perform cooperative distributed control[4],[5]. In conjunction with the development of agencies, organizations, companies and other places that use designs to be able to detect a form of object carefully and accurately such as the human brain demonstrated on a computer device, then this manual object shape detector can be replaced with automatic shape detection system[6].

This research aims to find an automatic and accurate method for visual inspection of the color quality and scratch defect on car roof surfaces, using the data generated from the standardized database QR Code.

In addition to the development of information technology, now the application of technology in the field of Machine or Computer Vision is able to overcome visual inspection problems. Computer Vision is a part of artificial intelligence that focuses on training computers to interpret and understand the visual world. The machine can then identify and perform scratch and color inspections accurately from captured digital images and deep learning methods[7], [8].

The detection system on the roof of the car that will be developed at this time is based on scratch and color inspection. Data will be extracted from the images that can provide some information such as color, shape, number and location of the objects being evaluated at once. Many applications use shape sensors such as digital cameras[9].

The study addresses the problem of carrying out an automated inspection on the surface of the roof of the car with the aid of a computer vision system that can increase accuracy in detecting scratch and color of the surface from the roof of the car. Thus, to overcome this problem, computer vision using a camera is needed by applying the Single-Shot Detection method [10],[11], particularly MobileNet-SSD model Deep Learning method system for checking the scratch on the roof surface of the car. Detection of color inspection will use the RGB to HSV filtering method which will compare the color of the object of each image pixel within a specific color[12]. The process with the deep learning method requires a dataset that can be further divided into training data and test data to support the computer learning process. The color inspection process in each image has a pixel color component that is compared with a specific color threshold that has been determined. If the pixel color component does not match it will be changed to a background color which is usually black[13]. The real-time computer vision system uses a camera as an image capturing device, mounted on the top of the car being accessed.

Therefore, this research will focus on the application of deep learning algorithms using the MobileNet-SSD library Tensorflow model for scratch inspections and the RGB to HSV filtering method for color inspections and utilizing the OpenCV library using the python programming language supported by computer with GPU (Graphics Processing Units)[14], [15].

Hence, the contribution of this research is by combining the application of deep learning algorithms using the MobileNet-SSD library Tensorflow model for scratch inspections and the RGB to HSV filtering method for color inspections on the car roof surfaces and to generate the QR code to database center[16], [17]. All those methods are applied to improve reliability and uniformity result on the car roof inspection line that have been manually performed by human operators in a car manufacturing plant.

The rest of this paper is organized as follows. The second section describes the research method of how to inspect surface and generate database of the inspection result. The third section addresses the implementation validation results and discussion. Finally, conclusion is found in the fourth section.

RESEARCH METHOD

The system is designed to inspect scratch and evaluate the color on the roof surface of the car with the aim of detecting defects for each object. The description from each step is stated in the following points.

The preliminary stage is the preparatory stage. Preparation is done by determining the topic and identifying the formulation of the problem with the research objectives to be achieved based on the methodology to be employed. Next, analyze the requirements of the hardware and the criteria of the software that will be used.

In the literature review, references were collected for scratch and color inspection system on the car roof surface using deep learning methods and RGB to HSV filtering. And the stage of data collection and data processing include choosing the data gathering procedure, outlining the experimental design, and interpreting the study.

After obtaining the training model from the data training process, the model is applied to the proposed scratch and color inspection system. Then the system as a whole will be evaluated. System testing will be carried out on all data, totaling 70 images that have been labeled and classified. The findings from the system assessment will result in the accuracy percentage of the image detection system.

As shown in the Figure 1, the system flow can be explained as followed; First camera taken a picture of car roof surface on the running conveyor belt then send it to machine/ computer vision. The computer will split the in two inspection sub-system which running real time during the car roof conveyed. The scratch inspection system used deep learning methods, MobileNet-SSD and TensorFlow library to collect and data processing including data collection, preprocessing design, training process and decision process. While at the same time RGB to HSV filtering method used to inspect the color of the surface. Finally, at the end both resulted outputs are gathered to generate the database and QR Code.

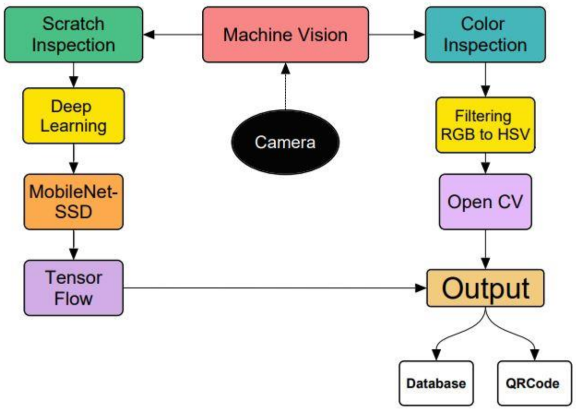


Figure 1. System Flow Diagram

After obtaining the training model from the data training process, the model is applied to the scratch and color inspection system. Then the system as a whole will be tested. System testing will be carried out on all data, totaling 70 images that have been labeled/classified. The results of the system test will result in the accuracy of the image detection system.

Development of the Detection Method

The detection method in this research is developed with the implementation of the Deep Learning Method with the MobileNet-SSD model using the TensorFlow framework which aims to inspect scratch car roof surface objects in real time, and to detect color using the OpenCV library using RGB to HSV method. The inspection results that are observed from the roof of the car will be entered in the database table and will return a barcode output.

Data Gathering Procedure

The first step is to collect custom data of 70 images which contained silver markers that are assumed to be the scratches using a webcam in the form of a jpg output. The image is labeled as one of the pre-determined classification types using graphical image annotation tool named LabelImg. The labelling process took a long time for resulting a good detection accuracy. The labelImg output will then be automatically saved as an image in an XML format.

Figure 2. Image Labelling Process

As shown in Figure 2, the car roof sample scratches are registered to the system. Several possibility shapes of scratch are introduced, such as curves, lines, S-shapes, X-shapes, random shapes. The image labelling process is performed manually for each single identified scratch. On the LabelImg GUI, each image introduced and labelled as required (whether it is classified as a scratch or others). Prior to labeling process, the roof area is set. These shapes were saved for the references of next training process.

Data Training Process

The creation of the architectural models from the scratch and color inspection systems was achieved using Tensorflow Object Detection. Figure 3 is an architectural scenario using Tensorflow Object Detection.

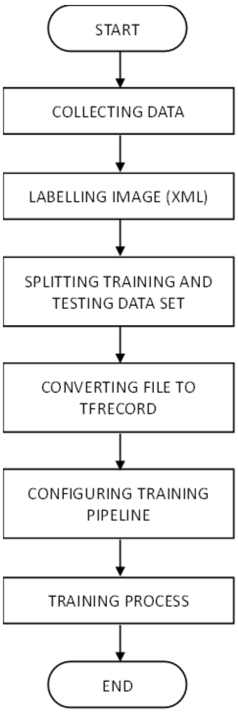


Figure 3. Flowchart of Training Data Process

In the training process, the data will be split into two categories, namely the train folder as much as 80% of the overall image and the test folder as much as 20% of the remaining photos. The data in the training folder is the data that will be trained by the machine using the deep learning method with the MobileNet-SSD model, whereas the data in the test folder will be used as a comparison with the data that has been trained.

The system will first resize the image into 300x300 pixels. Then, the second step is the administration of the convolutional process to get the feature of the image. The convolutional iteration should be done repeatedly until it reaches the smallest part of the image. The SSD method that is implemented in this training is derived from the MultiBox objective, but it is extended to handle multiple object categories.

After achieving the extraction process in the convolutional layer, the third step is accomplished by using the artificial neural network method to classify the image. The fully connected layer with a backpropagation method was used to identify the features of the image. The activation function that is applied to the classification is ReLu. The training process would take long time to get an identification model due to huge quantity of the images that are processed. Therefore, another alternative is required to process the training data quickly.

System Testing Technique

After the training process, the next step is to analyze the data. The result that has been obtained from the training process is the model pattern. Those models will be tested to measure accuracy.

During the testing phase, the camera system will be enabled for frame scanning based on real-time view. Furthermore, the detection by the system runs using the MobileNet-SSD model. When the car has been examined by the camera, the anaconda window will then display the reading results of the percentage of scratch accuracy found on the roof of the car. For color detection, the OpenCV will already have a pre-determined HSV value that becomes a reference threshold for whether the color is as desired or not. If the color matches the display, then the CMD will say the color type.

Evaluation

The last step in this research is the evaluation of the result. The assessment of findings may show a high accuracy for identifying the surface of car roof. However, if the evaluation result shows poor accuracy, then the model will re-test starting from the data processing step with changes in parameter values. In evaluating the color detection accuracy, in the case that if the color detection is not sensitive enough, the frame calculation value will reset.

RESULT AND DISCUSSION

Total Loss and Learning Rate in The Training Process

In the training process, the resulting data will produce a Total Loss Value and a Learning Rate Value that depends on the number of steps and the number of dataset images. Total Loss is a value that represents the summation of errors in our model. It measures the performance of our model whether in good or bad condition. Proportionally while the errors are high, the loss will be high, which means that the model does not perform a good job and vice versa. Learning Rate is the parameter scales the magnitude of our weight updates in order to minimize the network's loss function. While the learning rate is set excessively low, training will progress very slowly as tiny as updates to the weights in the network. However, if the learning rate is set excessively high, it can cause undesirable divergent behavior in the loss function.

As stated in Table 1, the process is carried out to measure the performance of training process of the dataset, wherein the total loss value of 2000 steps with a total of 40 images have 25 images with scratch and 15 images without scratch.

Table 1. Results of Total Loss and Learning Rate with 2000 training steps and a Dataset of 40 Images.

Step	Total Loss	Learning Rate
100	0,651022000	0,031999400
200	0,417736470	0,037332800
300	0,361596880	0,042666200
400	0,357730570	0,047999598
500	0,348939420	0,053333000
600	0,297554250	0,058666400
700	0,282085540	0,063999800
800	0,266177860	0,069333320
900	0,296564400	0,074666604
1000	0,324495800	0,080000000
1100	0,243310730	0,079999180
1200	0,261235900	0,079996705
1300	0,277814030	0,079992600
1400	0,249597340	0,079986850
1500	0,257811520	0,079979450
1600	0,226984140	0,079970405
1700	0,221932750	0,079959720
1800	0,233045790	0,079947400
1900	0,247260400	0,079933420
2000	0,213699000	0,079917810

The result, as shown in Table 2, is carried out to measure the performance of the training process of the dataset, wherein, the total loss value of 4000 steps with a total of 70 images have 55 images with scratch and 15 images without scratch.

Table 2. Results of Total Loss and Learning Rate Values with 4000 training steps and a Dataset of 70 Images

Step	Total Loss	Learning Rate
100	0,633093600	0,031999400
200	0,492506600	0,037332800
300	0,385595680	0,042666200
400	0,449717550	0,047999598
500	0,322948550	0,053333000
600	0,386253200	0,058666400
700	0,281608940	0,063999800
800	0,371614250	0,069333196
900	0,311488450	0,074666604
1000	0,352006170	0,080000000
1100	0,291992460	0,079999180
1200	0,346836660	0,079996705

Step	Total Loss	Learning Rate
1300	0,267274470	0,079992600
1400	0,277472800	0,079986850
1500	0,258418470	0,079979450
1600	0,286116240	0,079970405
1700	0,261133250	0,079959720
1800	0,269174370	0,079947400
1900	0,282159150	0,079933420
2000	0,252294630	0,079917810
2100	0,281912920	0,079900560
2200	0,242778400	0,079881670
2300	0,280990030	0,079861140
2400	0,207917600	0,079838970
2500	0,282485400	0,079815164
2600	0,243755520	0,079789720
2700	0,246104390	0,079762640
2800	0,229219990	0,079733920
2900	0,220751970	0,079703580
3000	0,244971990	0,079671600
3100	0,203461050	0,079637990
3200	0,291243970	0,079602750
3300	0,217565060	0,079565880
3400	0,214891550	0,079527386
3500	0,231438650	0,079487270
3600	0,199030920	0,079445526
3700	0,197361600	0,079402160
3800	0,198115440	0,079357184
3900	0,223956030	0,079310580
4000	0,230891820	0,079262360

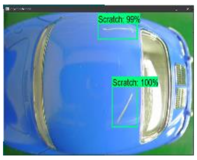
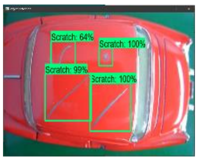
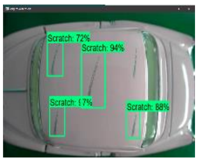
Based on the findings from the total loss of 40 images and 70 images in the training process, it shows that the number of images and steps affect the total loss in the training process. The training process only takes about 30 minutes. The largest Total Loss value occurs when there are 70 images with 2000 steps.

Scratch Inspection Result

System testing is carried out by setting several threshold values which will affect the sensitivity of frame reading with a maximum of 10 bounding boxes.

The best results of the scratch inspection test, as shown in Table 3, on the roof surface of the car based on the number of images, the number of steps and the thresh value, can be concluded that the smallest average error is 3.33% with a total of 70 images, a total of 4000 steps with a thresh value of 0.5.

Table 3. Scratch Inspection Testing Results of 0.5 threshold, 4000 training steps and 70 images.

No.	Threshold	Step	Image Number	Reading Result	*Error (%)*	*Average Error (%)*
1.	0,5	4000	70		0%	3,33%
2.	0,5	4000	70		0%	
3.	0,5	4000	70		20%	

Color Inspection Result

In real world images there is always variations in the image color values due to various lightening conditions, shadows and, even due to noise added by the camera. To over the above color variations, the color detection using HSV color space is proposed. The initial process of color inspection is performed by determining the desired upper and lower color values. If the object is colored according to the actual value, it will display that correct color type on the terminal, otherwise it will declare not detected as shown in Figure 4. All color detection processes are already contained in the OpenCV Library.

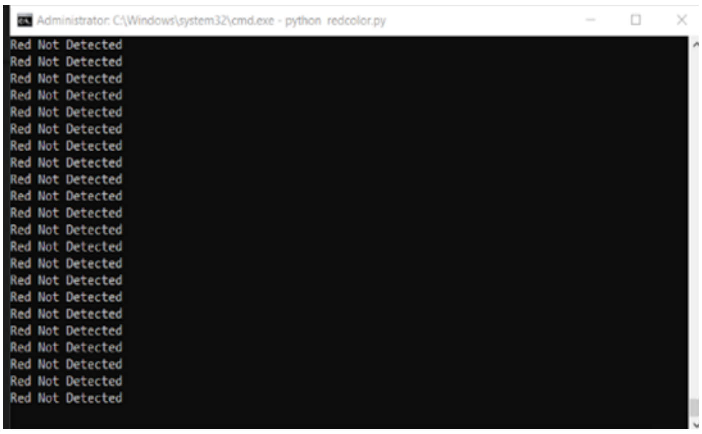


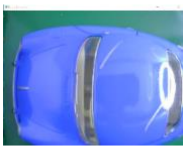
Figure 4. Color Inspection Result with threshold value > 600 for detecting other than red color

From the results of the color test evidence using the RGB to HSV method, it can be concluded that the color detection value is quite large compared to the color calculation itself. Wherein, when the value is greater than 600, the output prompt window will detect no appropriate color.

Overall System Testing

The overall system testing results are shown in Table 4. In testing the software architecture as a whole, the Python programming will run on the Anaconda terminal and will return prediction data results that include scratch inspection and color detection as an output on the HMI display.

Table 4. Overall System Test Results with a Threshold value of 0.5

No	Detection Result	Data Result	Notes	Prediction
1			No Scratch Detected, Red Color (PASS)	Correct
2			Scratch Detected, Red Color (REJECT)	Correct
3			No Scratch Detected, Blue Color (PASS)	False
4			Scratch detected, No Color detected (REJECT)	Correct
5			Scratch detected, Blue Color detected (REJECT)	Correct

From Table 4, the overall system test was carried out five times, where the results of correct predictions totaled 4 (four) trials and incorrect predictions totaled 1 (one) trial. Consequently, it can be concluded accordingly that the success of the scratch and color inspection system is 80% and able to run-in real-time condition.

Database Result

In the process of retrieving data stored in the database, the study used SQLite as selected. In the database display there are 4 rows, date (consists of the time also), scratch detection, color detection, and identification satisfactory results which concludes if the object inspection passed or rejected as depicted in Figure 5.

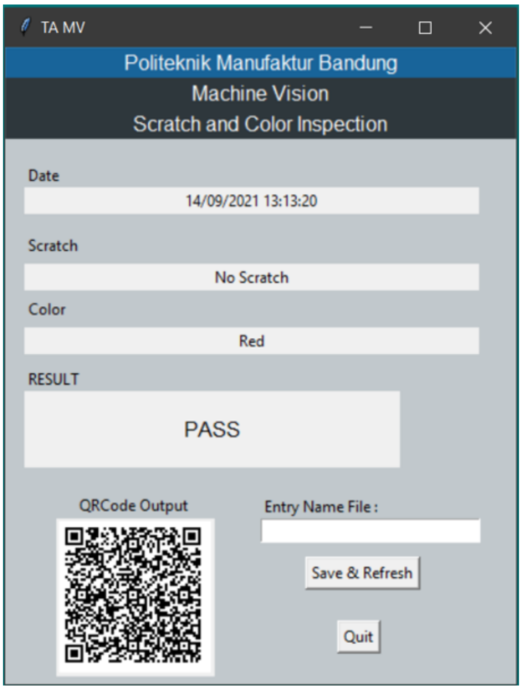


Figure 5. One inspection sample of Database Result GUI

Discussion

After the training process has been done, the result is a model that will be used to recognize the presence of a scratch on the surface of car roof. The MobileNet SSD method has been successfully construe the problem on scratch inspection fast, real-time, accurate and robust. Meanwhile, the HSV filtering technique is able to determine the color inspection problem fast, efficient and robust.

Classification loss is the loss value that appears when classifying an object. The smaller the value of classification value loss, the more accurate the system would be in predicting a scratch on the object. However, the more steps used, the lesser loss value becomes, which in this case had decreased to the value of 0. This is because the system learns to recognized patterns on the surface of the car roof.

CONCLUSION

Conclusion

In this research, it can be concluded that with the Mobile Net SSD for scratch inspection could be combined with HSV method for color inspection of car roof surface. The number of training steps affects the performance parameters of the system and the amount of data provided in the training process will affect the reading accuracy. Scratch Inspection with 70 image samples and 4000 training steps resulted in a 3.33% error, whereas, the total succeeded percentage for color inspection and detection is 80%.

Future Work

For future research, increasing the number of images will achieve improved and high accuracy. Adding car types would result in a more varied data classification. Including light flux will result in a better color inspection strategy. The current system control prototype is IOT ready and therefore set for the next development and web-based data processing purposes.

REFERENCES

[1] H. P.-J. of A. B. and Technology and undefined 2020, "Influence of NPM, PBV, DER, TATO, and EPS on Stock Prices of Automotive Sub Sector Companies and Its Components Listed on IDX in 2014-2018," *e-jabt.org*, vol. 2020, no. 3, pp. 151–162, Accessed: Mar. 08, 2023. [Online]. Available: https://e-jabt.org/index.php/JABT/article/view/41

[2] A. Muñoz, X. Mahiques, J. E. Solanes, A. Martí, L. Gracia, and J. Tornero, "Mixed reality-based user interface for quality control inspection of car body surfaces," *J Manuf Syst*, vol. 53, pp. 75–92, Oct. 2019, doi: 10.1016/J.JMSY.2019.08.004.

[3] A. Waluya, ... M. I.-I. J. of, and undefined 2019, "How product quality, brand image, and customer satisfaction affect the purchase decisions of Indonesian automotive customers," *inderscienceonline.com*, vol. 10, no. 2, pp. 177–193, 2019, doi: 10.1504/IJSEM.2019.100944.

[4] P. Anggraeni, W. A. Candra, M. Defoort, and M. Djemai, "Experimental implementation of fixed-time leader-follower axial alignment tracking," in *2019 International Conference on Mechatronics, Robotics and Systems Engineering (MoRSE)*, 2019, pp. 86–91.

[5] J. Inovasi Vokasional dan Teknologi, P. Anggraeni, and dan MTA Asshydiqi, "Penerapan Algoritma ORB SLAM-2 Pada Sistem Pemetaan Lingkungan Multi Robot," *INVOTEK: Jurnal Inovasi Vokasional dan Teknologi*, vol. 20, no. 3, pp. 123–134, Oct. 2020, doi: 10.24036/INVOTEK.V20I3.854.

[6] A. Chouchene, A. Carvalho, T. M. Lima, F. Charrua-Santos, G. J. Osório, and W. Barhoumi, "Artificial Intelligence for Product Quality Inspection toward Smart Industries: Quality Control of Vehicle Non-Conformities," *ICITM 2020 - 2020 9th International Conference on Industrial Technology and Management*, pp. 127–131, Feb. 2020, doi: 10.1109/ICITM48982.2020.9080396.

[7] A. Younis, L. Shixin, J. N. Shelembi, and Z. Hai, "Real-time object detection using pre-trained deep learning models mobilenet- SSD," *ACM International Conference Proceeding Series*, pp. 44–48, Jan. 2020, doi: 10.1145/3379247.3379264.

[8] C. Döring, A. Eichhorn, X. Wang, and R. Kruse, "Improved classification of surface defects for quality control of car body panels," *IEEE International Conference on Fuzzy Systems*, pp. 1476–1481, 2006, doi: 10.1109/FUZZY.2006.1681903.

[9] Q. Zhou, R. Chen, B. Huang, C. Liu, J. Yu, and X. Yu, "An Automatic Surface Defect Inspection System for Automobiles Using Machine Vision Methods," *Sensors 2019, Vol. 19, Page 644*, vol. 19, no. 3, p. 644, Feb. 2019, doi: 10.3390/S19030644.

[10] W. Liu, D. Anguelov, D. Erhan, ... C. S.-C. V., and undefined 2016, "Ssd: Single shot multibox detector," *Springer*, Accessed: Mar. 07, 2023. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2

[11] Y. Li, H. Huang, Q. Xie, L. Yao, and Q. Chen, "Research on a Surface Defect Detection Algorithm Based on MobileNet-SSD," *Applied Sciences 2018, Vol. 8, Page 1678*, vol. 8, no. 9, p. 1678, Sep. 2018, doi: 10.3390/APP8091678.

[12] N. Mohd Ali, N. Khair Alang Md Rashid, and Y. Mohd Mustafah, "Performance comparison between RGB and HSV color segmentations for road signs detection," *Trans Tech Publ*, 2013, doi: 10.4028/www.scientific.net/AMM.393.550.

[13] H.-C. Kang *et al.*, "HSV color-space-based automated object localization for robot grasping without prior knowledge," *mdpi.com*, 2021, doi: 10.3390/app11167593.

[14] J. Tang, X. Peng, X. Chen, and B. Luo, "An Improved Mobilenet-SSD Approach for Face Detection," *Chinese Control Conference, CCC*, vol. 2021-July, pp. 8072–8076, Jul. 2021, doi: 10.23919/CCC52363.2021.9549245.

[15] L. Arnal, J. E. Solanes, J. Molina, and J. Tornero, "Detecting dings and dents on specular car body surfaces based on optical flow," *J Manuf Syst*, vol. 45, pp. 306–321, Oct. 2017, doi: 10.1016/J.JMSY.2017.07.006.
[16] X. Wei, A. Manori, N. Devnath, ... N. P.-I. J. of, and undefined 2017, "QR code based smart attendance system," *academia.edu*, Accessed: Mar. 26, 2023. [Online]. Available: https://www.academia.edu/download/56635087/1.pdf
[17] A. Espejel-Trujillo, ... I. C.-C.-P., and undefined 2012, "Identity document authentication based on VSS and QR codes," *Elsevier*, Accessed: Mar. 26, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212017312002551