

Research Paper

Visual Inspection Improvement of Engine Components Using Deep Learning with Pre-processed Dataset Augmentation: Case Study

Steven Salim¹, Sandy Wiratama¹, Alfian Sarifudin¹✉, Eka Prita Yuliatin²

¹Department of Computer Engineering, Bina Nusantara University, Jakarta, 11480, Indonesia

²Department of Industrial Engineering, Bina Nusantara University, Jakarta, 11480, Indonesia

✉ alfan.sarifudin@binus.ac.id

🌐 <https://doi.org/10.31603/ae.14207>

Published by Automotive Laboratory of Universitas Muhammadiyah Magelang

Article Info

Submitted:

19/07/2025

Revised:

21/10/2025

Accepted:

01/11/2025

Online first:

05/12/2025

Abstract

Lighting instability, sharp shadows, and visual disturbances caused by mechanical vibrations are significant challenges in the application of computer vision-based visual inspection systems in automotive industrial environments. This study aims to enhance the accuracy and robustness of the YOLOv8 object detection model for detecting machine component completeness by applying an adaptive pre-processing strategy. The techniques employed include grayscale conversion, brightness adjustment, and blurring to simulate common visual conditions encountered in real-world production processes. The model was trained using 1,281 instances from 52 component classes and evaluated based on the metrics of precision, recall, mAP@50, and mAP@50–95. The results show an average precision of 0.971, a recall of 0.990, and mAP@50 of 0.991, with spatial variation reflected in the standard deviation of mAP@50–95 of 0.149. The pre-processing technique improves the detection precision of shape-based components by up to 19% and colour-based components by up to 31%. Testing on ten appearance variations showed 100% detection accuracy with no misclassification, indicating the model's generalizability to data in the training distribution. These findings confirm that visual modification of training data significantly improves the reliability and efficiency of the YOLOv8-based automated inspection system. Further implications include reduced human intervention, accelerated production flow, and optimization of operational energy consumption through faster and more accurate detection. Therefore, this system contributes to energy-efficient and sustainable innovative manufacturing practices.

Keywords: Object detection; Automatic inspection; Automotive factory intelligence; Engine components; Smart manufacturing; Sustainable production

1. Introduction

The engine is a core component in a vehicle that enables the entire system to operate according to its designed functions. Engine variations differ significantly depending on vehicle size, fuel type, and emission standards, resulting in diverse component configurations [1]. Consequently, the process of checking the completeness and correctness of components during engine assembly becomes a complex and time-consuming task, particularly at the final inspection stage known as the quality line. This highlights the limitations of conventional manual

inspection methods, which often suffer from inefficiencies and inconsistent accuracy.

To address these issues, a robotic based inspection system utilizing the YOLO (You Only Look Once) object detection method has been developed to automatically identify engine components on the quality line. The system is intended to reduce the workload of human operators by detecting visible components in real time. However, despite these advancements, the current implementation remains suboptimal. One of the main challenges is lighting instability, where low ambient light causes artificial



This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

illumination to reflect excessively off metallic surfaces, resulting in overexposed images. Additionally, camera motion or robot arm vibration can cause image blurring, which reduces object recognition accuracy. These visual distortions lead to false positives or missed detections, producing incorrect inspection results such as "NG" (Not Good) or "Checking" even when components are correctly installed. Such errors diminish system efficiency and still require manual verification, which contradicts the goal of full automation.

Most existing studies such as research on visual inspection systems for product label placement using deep learning have primarily focused on enhancing model accuracy under ideal environmental conditions, without addressing real-world challenges like unstable lighting and mechanical vibration in robotic systems [2]. Furthermore, few studies have examined the influence of image pre-processing techniques including grayscale conversion, brightness adjustment, and blur filtering on YOLO's detection performance metrics such as precision, recall, mAP, and confidence stability [3]. This gap highlights the need for a more robust and adaptive detection framework that can maintain performance under varying environmental conditions.

Therefore, this study proposes an Enhanced Visual Inspection of Machine Components based on the YOLOv8 Object Detection Algorithm. The proposed enhancement focuses on improving detection robustness by training the YOLOv8 model with a modified dataset that incorporates realistic pre-processing augmentations, including grayscale conversion, brightness variation ($\pm 20\%$), and blur effects (3 to 5 pixels kernels). These augmentations simulate actual industrial lighting and motion conditions, enabling the system to detect components more consistently and accurately in non-ideal visual environments. Through this approach, the enhanced YOLOv8 model is expected to achieve higher robustness and stability in real-world quality line inspections.

2. Literature Study

2.1. YOLOv8

YOLO (You Only Look Once) is an artificial intelligence (AI) object detection algorithm based

on deep learning (DL) that has undergone significant developments since its initial introduction to the public. YOLO is widely known for its ability to detect objects in real-time with high efficiency [4]. Since its initial version, this algorithm has evolved into various advanced versions such as YOLOv3, YOLOv5, and even the most recent one, YOLOv11. However, the one used in this study is YOLOv8 because this version is considered lighter, more stable, and more efficient for small to medium-scale datasets that do not require high complexity like YOLOv11 [5].

YOLOv8 uses a Convolutional Neural Network (CNN) as the backbone to extract visual features from images. However, unlike previous versions of YOLO that rely on anchor boxes, YOLOv8 adopts an anchor-free approach, which allows the model to predict the position and size of objects directly from image features without requiring anchor references [5]. This approach simplifies the training process and enhances the flexibility and precision of object detection for objects with diverse shapes and sizes [6, 7].

In its training process, YOLOv8 divides the dataset into several batches. Each batch contains a specific amount of data (for example, 10 images) that is processed simultaneously (for example, if there are 50 training data points, then there will be five batches). This process is carried out in stages in a unit of time called an epoch, which is one complete cycle of the model learning the entire dataset [5]. In each batch, the model performs a forward pass to produce predictions, followed by a backward pass using a loss function to calculate the error between the prediction and the actual label. This error is used to update the network weights through an optimization algorithm [6].

The training result weights at the end of each batch will be updated and then continued to the next batch until one epoch is completed. After all epochs are completed, the final weights obtained represent the trained model. These weights are then used in the inference process to perform object detection on test data or new data [8]. Figure 1 illustrates the training process of the YOLO model over two epochs, where each batch of training data is validated against itself, and the model with the highest validation accuracy is selected as the final model.

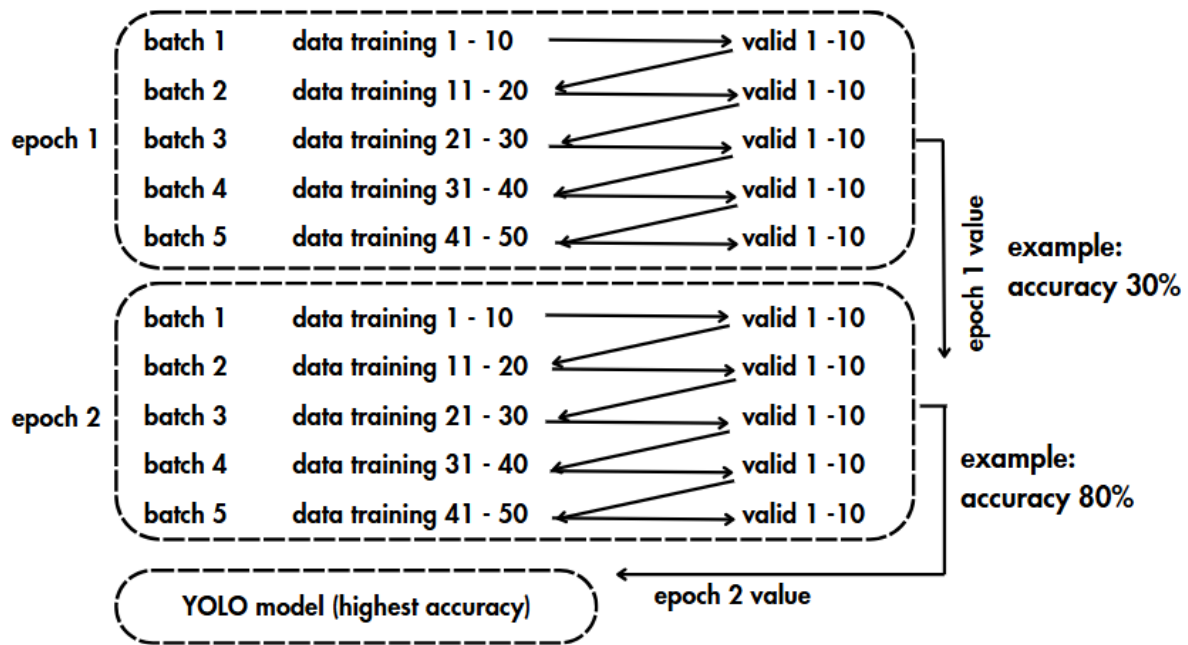


Figure 1. YOLO Architecture

YOLOv8 maintains high efficiency in both inference and training processes, while supporting advanced tasks such as instance segmentation, image classification, and precise object counting. This capability makes it very relevant for industrial applications, especially in the field of robotics. One example is the completeness detection system for machine components, which requires fast and accurate object detection even in conditions with limited computing resources [8].

2.2. CNN

The Convolutional Neural Network (CNN) is one of the most influential and widely used deep learning architectures in the field of digital image processing, particularly in tasks such as pattern recognition, object classification, and object detection. CNN is designed to efficiently process grid-shaped data, such as two-dimensional images, by applying convolutional layers that can automatically extract spatial features [9]. The hierarchical structure of CNN enables increasingly complex feature representations from layer to layer, making it highly effective in understanding visual information at different levels of abstraction.

CNN consists of several main components, namely convolution layers, activation layers (such as ReLU), pooling layers, and fully connected

layers. The convolution layer functions to extract features such as edges, textures, or patterns from images by applying filters or kernel [9]. The pooling layer is used to reduce data dimensions and prevent overfitting by retaining important information, while the fully connected layer plays a role in the final stage of classification or regression [10].

2.3. Grayscale

Grayscale is a technique for converting coloured images (RGB) to grayscale images that store only light intensity information, without involving colour elements, as illustrated in Figure 2. In the context of deep learning model-based object detection, such as YOLO, grayscale images can significantly impact the visual feature extraction process because the model focuses more on texture and shape patterns than colour variations [11]. The grayscale approach is advantageous when the system is applied to environments with inconsistent lighting or complex visual conditions, where dependence on colour can reduce detection accuracy.

The removal of colour information forces the model to focus more on the shape, texture, and edge characteristics of the object. This approach has proven effective in scenarios where the shape or contour of the object is the primary factor in identification, such as in detecting machine

components, hardware, or objects that have similar colours but distinct shapes [12]. Several studies have also demonstrated that models trained on grayscale data tend to exhibit a sharper spatial focus in recognizing structural patterns compared to models trained on colour images [13].

In this study, a grayscale approach is applied as part of a dataset modification strategy to improve object detection performance. Converting coloured images to grayscale is done to test the hypothesis that emphasizing shape, texture, and edge features can help the model ignore visual noise such as light reflections, sharp shadows, or colour fluctuations on the object's [13], [14]. Through this approach, it is expected that the model will be more robust and consistent

in performing detection, especially in non-ideal lighting conditions or when the object has visual characteristics that resemble the background.

2.4. Brightness

Brightness is a crucial element in the visual representation of digital images, as it directly affects the perception of objects within the image. In the development of deep learning-based object detection models, brightness adjustment is often used as part of the pre-processing or data augmentation process. The goal of this technique is to improve the model's robustness to unstable lighting variations, such as sharp shadows, overexposed light, or low-light conditions, which are common in industrial and real-world environments [6], [15]. This adjustment enables

Example of detection part

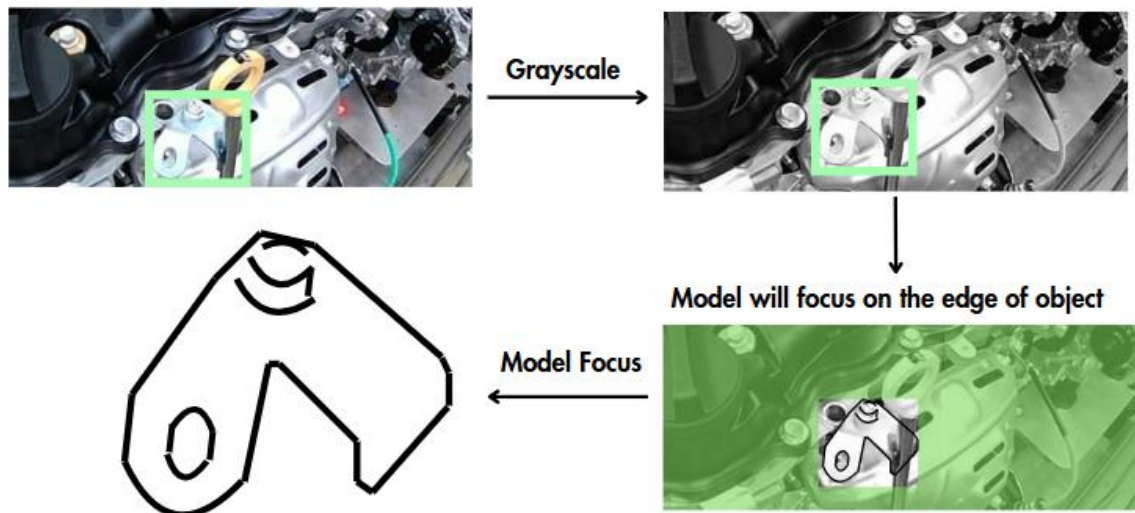


Figure 2. Effect of grayscale on weighting

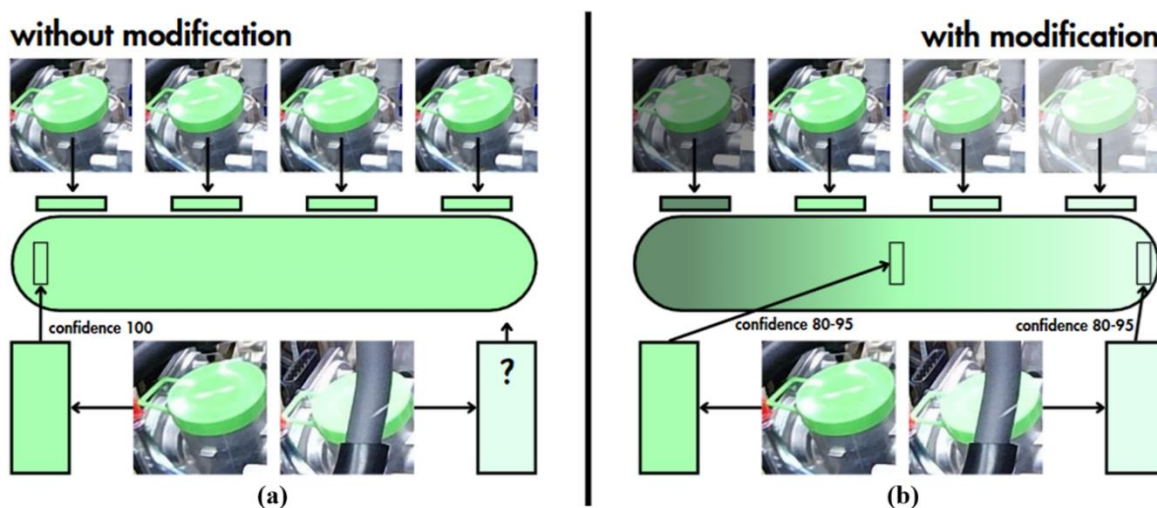


Figure 3. Influence of brightness on weighting (a) without modification and (b) with modification

the model to learn to recognize objects not only based on ideal light intensity but also in extreme scenarios that may not be represented in the initial dataset.

Increasing the brightness of an image can help object detection models better utilize colour information and tonal variations as key distinguishing features as illustrated in [Figure 3](#). The brightness approach becomes particularly relevant when observing objects with colour-dependent visual characteristics, such as packaging labels, color-coded cables, or visually marked machine components. By making controlled brightness adjustments during the training process, the model can learn to ignore visual noise, such as shadows or excessive highlights, and still accurately detect objects even in less-than-ideal lighting conditions [\[6\]](#).

3. Methodology

This study aims to implement the YOLOv8 algorithm in a robotic system to detect the completeness of components on an engine. The system formed works automatically, starting from image detection by the camera to verifying the conformity of components with standard configurations. The approach used involves several main stages, starting with the integration of YOLOv8 on the robot's camera, setting the robot's movements to produce a systematic view, and evaluating the detection results based on completeness standards.

3.1. YOLOv8 Based Detection System

The trained YOLOv8 model is used to detect engine components in real time. This model is integrated into a camera mounted on the robot. Therefore, when the robot moves and takes pictures, the system immediately detects objects visible in each view. YOLO will process each image taken and provide results in the form of bounding boxes, identity labels, and confidence of each detected component. This information is then processed by the system to be compared with the list of elements that should appear on that engine type.

3.2. Visual Data Acquisition System

The visual data acquisition system serves as the main component in collecting image data used for object detection within the robot's workspace.

In this study, the visual acquisition system consists of the following components:

3.2.1. Camera

The vision module employs a webcam with a 1080p resolution (1920 × 1080 pixels) and a 150° field of view. The camera captures real-time images of the robot's working area. This type of camera was selected for its wide viewing angle, allowing a broader coverage of the workspace without frequent repositioning.

3.2.2. Lighting system

A circular fixed light (ring light) with a diameter of 18.5 cm and power of 9 watts was used to provide uniform illumination. The lighting ensures stable brightness and minimizes shadows on the observed objects, improving image quality and consistency during computer vision-based detection.

3.2.3. Robot arm

The detection platform utilizes a demo robotic arm with six degrees of freedom (6-DOF). The robot functions as a movable base that positions the camera at predetermined viewing angles. It enables repetitive image capture from consistent positions, ensuring systematic data collection across experiments.

3.2.4. Control and computing system

All data acquisition processes are managed by an industrial computer that interfaces with both the camera and the robotic arm through a serial communication link. The acquired images are stored digitally and later processed using computer vision algorithms for object detection and evaluation.

The physical configuration of the robotic visual detection system including the camera, lighting, and robot manipulator assembly is illustrated in [Figure 4](#). This setup ensures stable mechanical positioning and consistent visual data acquisition during experimental trials.

3.3. Robot Movement and View Detection

[Figure 5](#) shows the workflow of a robot designed to perform engine component completeness detection through ten predetermined views. The process begins with the initial detection of the presence of an engine in the

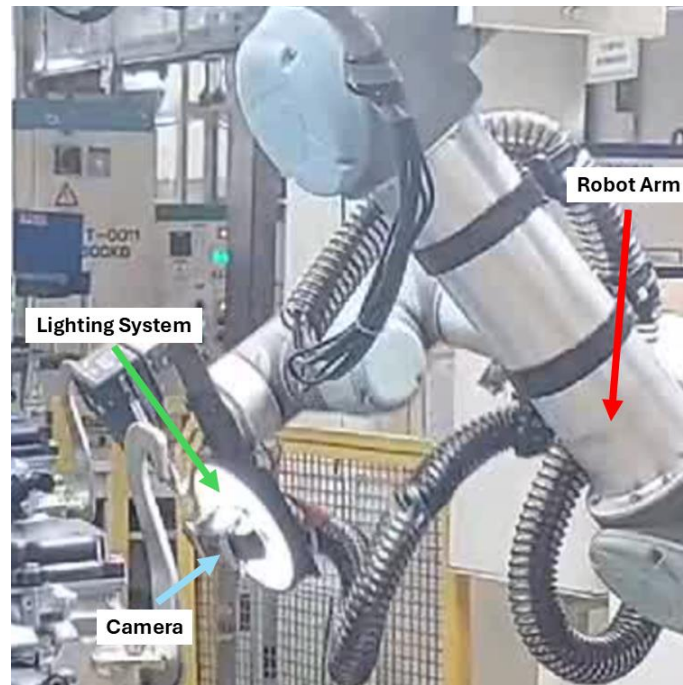


Figure 4. Robot physical configuration

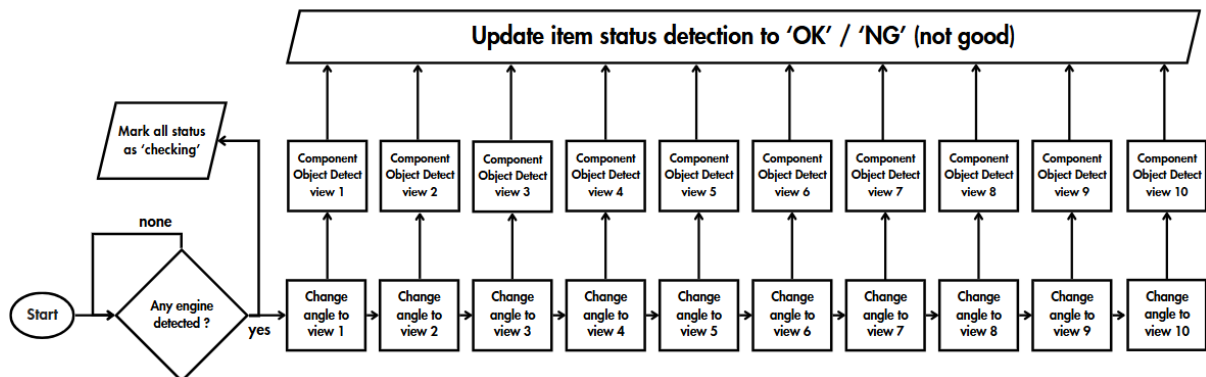


Figure 5. Flowchart of engine component completeness detection robot

work area; if detected, the robot will initiate a series of stepwise movements to each viewpoint. In each view, the system will use a YOLOv8-based detection model to identify the presence of a specific component according to the configuration. The detection result of each view was then updated to an 'OK' or 'NG' (not good) status, depending on the match between the detected component and the expected component. This approach allows the system to assess completeness comprehensively, even under varying object position conditions.

3.4. Components Checked in Each View

Each view generated from the robot's movement focuses on a specific area of the engine that contains each component. These components

have been previously annotated in the dataset and serve as references during the model training process, as well as the primary reference when visual disturbances, such as overexposure or blur, occur during the detection process. Details of the components checked in each view are presented in [Table 1](#).

3.5. Completeness Evaluation

After the detection process was carried out on each view, the system evaluates the detection results to verify the conformity between the identified components and the standard configuration of the engine type being checked. Any components that are not detected, are in the wrong position, or do not meet specifications will be assigned the status 'NG' (Not Good).

Table 1. List of item components detected in each view.

View Num.	Component Name			
	Components with shape distinction	Shape based distinction labels	Components with colour distinction	Colour based distinction labels
1	HOLE_MOUNTING	MOUNTING_PINTHREADTHREAD MOUNTING_THREADPINTHREAD	-	-
	STAMPLE_INDIKASI_TESTER	STAMPLE_OK_TESTER		
2	TENSIONER_VRIBBED	VRIBBEDWITHOUT_HOLE VRIBBEDWITH_HOLE	-	-
	STAMPLE_INDIKASI_TESTER	STAMPLE_OK_TESTER		
3	BRACKET_TOBLOCK	BRACKET_BLOATED BRACKET_FLAT	SOCKET_KNOCKSENSOR	SOCKET_GRAY SOCKET_BLACK
	FACE_PULLEY	PULLEYWITH_GROOVE PULLEYWITHOUT_GROOVE		
4	GUIDE_OILLEVEL	BRACKET_FACINGUP BRACKET_FACINGDOWN	STICK_OILLEVEL	STICKOIL_BLACK STICKOIL_GREEN STICKOIL_PINK
			AF_SENSOR	AFSENSOR_GREENWIRE AFSENSOR_WHITEWIRE
5	FACE_HEADCOVER_STUD2	STUD_2	PCV	PCV_ORANGE
	WAYOF_CLIPVENTILATION	CLIP_FACINGFRONT	HOSE_VENTILATION_TOPCV THROTLE_BODY	HOSEPCV_MARK_YELLOW CAP_WHITE CAP_GREEN CAP_YELLOW CAP_ORANGE
6	BRACKET_CHC	WITH_BRACKETCHC	PCV	PCV_ORANGE PCV_BLACK
	FACE_HEADCOVER_STUD1	STUD_1	HOSE_VENTILATION_TOPCV	HOSEPCV_MARK_YELLOW
	FACE_HEADCOVER_STUD2	STUD_2	THROTLE_BODY	CAP_WHITE CAP_GREEN CAP_YELLOW CAP_ORANGE
	WAYOF_CLIPVENTILATION	CLIP_FACINGREAR	-	-
7	FACE_INTAKE	INTAKE_BOLTLH INTAKE_BOLTRH INTAKE_SPECIALHYBRID	HOSE_VENTILATION_TOIN TAKE	HOSEINTAKE_MARK_WHITE HOSEINTAKE_MARK_PINK
8	WAYOF_CLIPVENTILATION	CLIP_FACINGRIGHT	PCV	PCV_ORANGE PCV_BLACK
	FACE_HEADCOVER_STUD1	STUD_1	HOSE_VENTILATION_TOPCV	HOSEPCV_MARK_GREEN
	FACE_HEADCOVER_STUD2	STUD_2	THROTLE_BODY	CAP_WHITE CAP_GREEN CAP_YELLOW CAP_ORANGE
9	BRACKET_HARNESS_REAR	B_HARNESS_NR B_HARNESS_KAI	BOLT_DP	BOLT_DP_BLACK BOLT_DP_WHITE
	DRIVE_PLATE	DP_NR DP_NRK	-	-
	COVER_DISC_CLUTCH	COVERCODE_N2 COVERCODE_O2 COVERCODE_EXEDY COVERCODE_AISIN	THROTLE_BODY	CAP_WHITE CAP_ORANGE
10	-	-	MARKING_EXHAUST	EXHMARK_PINK EXHMARK_PURPLE EXHMARK_ORANGE EXHMARK_BLUE EXHMARK_GREEN

Meanwhile, components that have not been processed will remain in the 'checking' status or be recorded as 'Null'. The system uses the results of this evaluation to determine the overall completeness status of all components. Based on the aggregation of these statuses, the system then determines whether the engine can be declared complete and meets the standards or still has deficiencies that need to be addressed.

3.6. Dataset Modification and Model Retraining

To improve the accuracy and robustness of the YOLOv8 model to unstable lighting conditions and camera movements Modifications were made to the original dataset through three image pre-processing approaches, namely grayscale conversion, brightness adjustment, and blur effect.

3.6.1. Grayscale conversion

The image was converted to grayscale to emphasize the shape, texture, and edge of each component [16]. This technique was particularly useful for components with dominant structural or geometric features rather than colour-dependent characteristics. In this study, grayscale conversion was applied to 50% of the training dataset, enabling the model to learn representations that are both colour-independent and structure-focused. The grayscale intensity was set to 100%, meaning that all colour information was completely transformed into a range of grey tones, ensuring consistency in luminance while preserving essential spatial details for object recognition [17].

3.6.2. Brightness adjustment

Brightness was randomly increased or decreased within a range of 0.8 to 1.2 times the original intensity, representing a maximum adjustment of $\pm 20\%$ [18], [19]. This augmentation was designed to simulate real environmental lighting variations, particularly to enable the model to detect objects accurately under actual field conditions. In many practical scenarios, overexposure frequently occurs due to artificial lighting, especially when the surrounding ambient illumination was low [20]. The selected brightness range effectively mimics these realistic lighting fluctuations without introducing extreme visual distortion, ensuring that the model remains

robust and capable of maintaining reliable object recognition performance [17].

3.6.3. Blur effect

A light blur was applied to simulate the effect of camera movement or slight image capture distortion, enabling the model to recognize objects even when motion blur or minor shaking occurs from the robot [18], [21]. In this study, the blur kernel size was set between 3×3 and 5×5 pixels (equivalent to approximately 3–5 pixels blur intensity), representing a light blur commonly used in image augmentation research. This range was sufficient to mimic mild camera vibration or focus drift without significantly degrading object features [21] [22].

The modified dataset was then used to retrain the YOLOv8 model. Training was done by dividing the data into several batches and epochs. At each epoch, the model weights are updated based on the results of the previous training, until the optimal final weights are achieved. This trained model was then used in the robot system to detect engine components more accurately and stably.

3.7. Precision and Recall Results

Precision is a metric used to measure the level of accuracy of optimistic predictions from a model, specifically the extent to which positive prediction results align with actual conditions. This metric is calculated as the proportion of true positives to all positive prediction results, as formulated in Equations (1) and (2). In this context, true positives refer to data that was predicted positive by the model and was indeed included in the positive category based on the training label. Meanwhile, the overall positive results encompass all data predicted to be positive, including both true positives and false positives [6].

$$precision = \frac{\text{number of correct guesses of label } x}{\text{total number of guesses label } x} \quad (1)$$

$$precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (2)$$

In general, a precision value of >0.9 is considered very good, as it indicates that most positive predictions are correct. Precision with a value above 0.5 is generally still acceptable as a minimum standard in evaluating model performance. Conversely, precision below 0.5

indicates poor positive prediction accuracy, because more than half of the positive results are errors [5].

Recall is a metric that measures the extent to which a model can detect all positive cases that are present in the data, as formulated in Equations (3) and (4). In other words, recall calculates the proportion of true positives to all data that should have been classified as positive. This metric becomes very important in the context of applications where failure to detect positive objects can have serious consequences, such as in the detection of product defects or uninstalled machine components [3], [5], [6].

$$\text{recall} = \frac{\text{number of correct guesses of label } x}{\text{number of labels } x \text{ that should be detected}} \quad (3)$$

$$\text{recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (4)$$

Like precision, recall values are generally considered very good if they reach a value of ≥ 0.9 , as this indicates that the model can detect almost all existing positive cases. Recall that a value above 0.5 is usually still accepted as the minimum standard limit in evaluating model performance. Conversely, a recall below 0.5 indicates poor detection performance, as it means that most positive cases fail to be recognised by the model [3], [5], [6].

The relationship between precision and recall reflects the trade-off in the performance of a detection model. When recall is high, but precision is low, it means that the model successfully detects almost all positive objects, but many of them are incorrectly detected (false positives). Conversely, when precision is high, but recall is low, the model only makes predictions when it is very confident (high confidence), making it more accurate but at the risk of missing many objects that are positive (false negatives) [5], [24].

3.8. mAP50 and mAP50-95

The performance evaluation of the object detection model was conducted using the mean average precision (mAP) metric at an intersection over union (IoU) threshold of 0.5, also known as mAP@50. This metric is used to assess the success rate of the YOLO model in automatically detecting engine components, based on the agreement between the prediction and the actual position of

the object [6]. Before the evaluation process, the dataset was manually annotated to determine the ground-truth bounding box of each component to be detected. The YOLO model was then run to generate object position predictions, and the prediction results were compared with the ground truth data to calculate the level of spatial accuracy. The IoU value between the predicted bounding box and the ground truth was calculated using the formula shown in Equation (5) [25].

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of union}} \quad (5)$$

Each detection result that has an intersection over union (IoU) value of at least 0.5 or 50% of the ground truth is categorised as a true positive. In contrast, detections that do not meet this threshold are considered false positives or false negatives, depending on whether the actual object is present and its position in the image [6]. Based on these results, the precision and recall values are calculated for each class, which are then used to form a precision-recall curve. The area under the curve is calculated as the average precision (AP) for each class. The average of all AP values for all classes is then used as the mean average precision value at an IoU threshold of 0.5 (mAP@50), serving as an indicator of overall model performance [22].

Unlike mAP@50, which only relies on one IoU value as an evaluation reference, mAP@50-95 is calculated by taking the average of ten mean average precision values at various IoU thresholds, ranging from 0.50 to 0.95 with an interval of 0.05 [24]. The mAP@50-95 approach provides a more comprehensive assessment of the model's ability to detect objects precisely at various levels of overlap tolerance. Therefore, mAP@50-95 is considered more representative in measuring model performance in general, especially in complex scenarios [25]. The results of both metrics, namely mAP@50 and mAP@50-95, are used to evaluate the model's accuracy and robustness to variations in lighting conditions and visual disturbances, such as blur [21].

4. Results and Discussion

4.1. Modelling Results

In the evaluation stage, the YOLOv8 model's performance was tested for detecting machine components using a modified dataset that has

undergone image pre-processing techniques. This process involves converting images to grayscale, adjusting brightness levels, and applying blur effects. The primary purpose of this modification is to enhance the model's robustness to varying visual conditions in real-world environments. The dataset used includes 52 component labels, which are categorised into 23-part classes and have been manually annotated before being used in training. Performance evaluation is carried out using precision, sensitivity (also known as recall), and mean Average Precision (mAP) metrics at two IoU thresholds, namely mAP50 and mAP50–95. **Table 2** presents the evaluation results for all tested components. In aggregate, the model was tested on 1,281 instances from 52 classes and achieved an average mAP50 value of 0.991, a precision of 0.971, and a recall of 0.990. These values indicate the model's tendency to recognise objects with high accuracy and low error rates.

The distribution of metrics shows that most parts exhibit very high precision and recall, indicating the model's strong ability to correctly identify component presence. As many as 75% of all parts achieved precision values above 0.955 and a recall of 1.000. The highest precision value reached 1.000, and the lowest was 0.929, while recall values ranged from 0.809 to 1.000. These improvements can be attributed to the applied pre-processing techniques, particularly grayscale conversion, brightness adjustment, and controlled blur filtering which effectively enhanced the dataset's visual robustness.

By converting certain images to grayscale, the model learned to focus more on structural and textural features rather than colour dependency. The controlled brightness adjustment (range 0.8–1.2) improved adaptability to various lighting conditions, reducing false detections in overexposed areas. Meanwhile, the slight blur addition (3–5 pixels) helped the model generalise better to real-world motion blur caused by robot vibrations.

As a result, the mAP50 values remained highly uniform, with most parts achieving 0.995, demonstrating consistent spatial detection performance. In contrast, the mAP50–95 metric showed greater variation (minimum 0.453, maximum 0.995, average 0.868), reflecting the model's sensitivity to fine-grained boundary localisation at higher IoU thresholds.

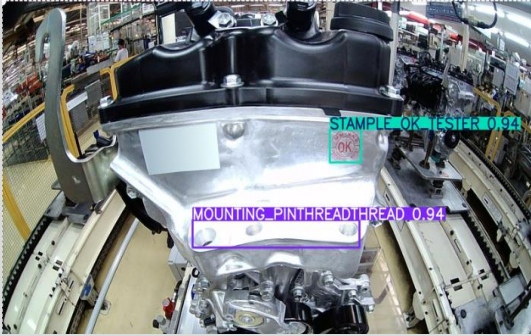
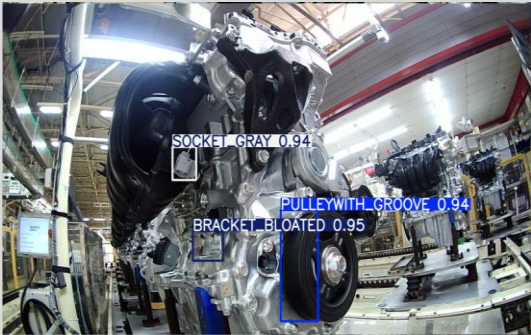
The relatively low standard deviations of precision (0.021), recall (0.035), and mAP50 (0.017) confirm stable performance across all classes. Although the larger standard deviation in mAP50–95 (0.149) indicates spatial variation, this variability is expected given the visual diversity of component textures and shapes. Overall, the results demonstrate that the integration of grayscale, brightness, and blur augmentations significantly improves the YOLOv8 model's precision, recall, and spatial accuracy under realistic industrial lighting and motion conditions. Following the initial training and evaluation process on the training and validation datasets, further testing was conducted to assess the YOLOv8 model's generalisation capability in real-world conditions. This testing was performed using a test dataset composed of ten variations of views representing various combinations of machine component placement and orientation, as listed in **Table 3**. Each view consists of several test images with unique visual configurations, allowing for the evaluation of model detection consistency across spatial and contextual variations.

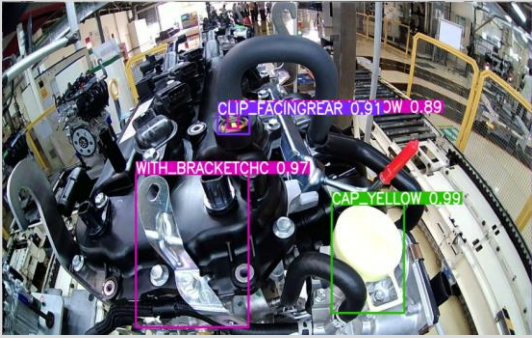
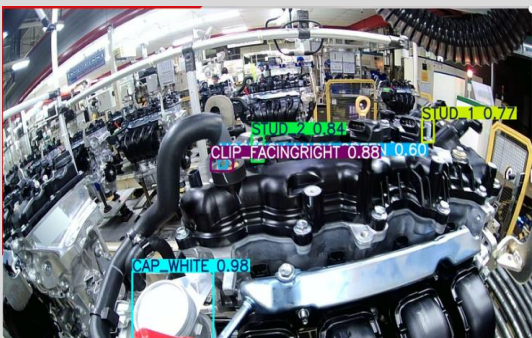
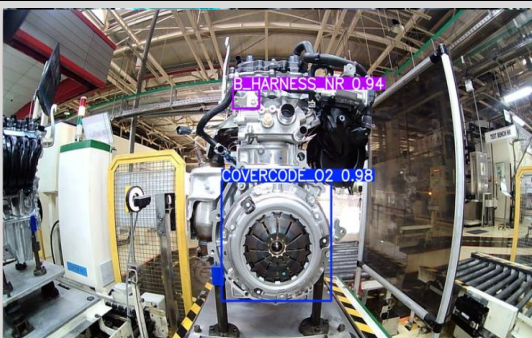
The results of the test, conducted with total of 320 images or 32 images for each view test under varying environmental conditions, demonstrate that the model achieved a flawless detection accuracy of 100%. The images were taken across different times of the day, including during cloudy weather and at night, which naturally caused variations in lighting. In the conditions with low ambient light can lead to excessive reflections from metallic surfaces, resulting in overexposed images. However, the model consistently performed well, showcasing its robust ability to adapt to such lighting challenges. Therefore, the model's resilience was primarily attributed to the image pre-processing techniques employed, including grayscale conversion, brightness adjustment, and blur effects, which enhanced the model's resilience to common real-world visual disturbances such as lighting fluctuations and surface texture variations. The perfect detection accuracy, even under complex and challenging conditions, indicates that the model's generalization capabilities have been successfully strengthened. It can now reliably detect objects even in scenarios that previously posed difficulties, such as low-light environments

Table 2. Results of modelling

Detail Part Name	Precision	Recall	mAP50	mAP50-95
AFSENSOR_GREENWIRE	0.986	1.000	0.995	0.856
AFSENSOR_WHITEWIRE	0.965	1.000	0.995	0.759
BOLT_DP_BLACK	0.972	1.000	0.995	0.965
BOLT_DP_WHITE	0.940	1.000	0.995	0.983
BRACKET_BLOATED	1.000	1.000	0.995	0.839
BRACKET_FACINGDOWN	0.986	1.000	0.995	0.986
BRACKET_FACINGUP	0.960	1.000	0.995	0.945
BRACKET_FLAT	0.987	1.000	0.995	0.802
B_HARNESS_KAI	0.986	1.000	0.995	0.919
B_HARNESS_NR	0.981	1.000	0.995	0.828
CAP_GREEN	0.993	1.000	0.995	0.956
CAP_ORANGE	0.961	1.000	0.995	0.935
CAP_WHITE	0.992	1.000	0.995	0.932
CAP_YELLOW	0.952	1.000	0.995	0.980
CLIP_FACINGFRONT	0.945	0.955	0.978	0.605
CLIP_FACINGREAR	0.989	1.000	0.995	0.770
CLIP_FACINGRIGHT	1.000	0.996	0.995	0.562
COVERCODE_AISIN	1.000	1.000	0.995	0.995
COVERCODE_EXEDY	0.946	1.000	0.995	0.995
COVERCODE_N2	0.945	1.000	0.995	0.995
COVERCODE_O2	0.946	1.000	0.995	0.995
DP_NR	0.939	1.000	0.995	0.995
DP_NRK	0.972	1.000	0.995	0.995
EXHMARK_BLUE	0.955	1.000	0.995	0.995
EXHMARK_GREEN	0.956	1.000	0.995	0.995
EXHMARK_ORANGE	0.955	1.000	0.995	0.995
EXHMARK_PINK	0.955	1.000	0.995	0.995
EXHMARK_PURPLE	0.956	1.000	0.995	0.987
HOSEINTAKE_MARK_PINK	1.000	0.977	0.995	0.541
HOSEINTAKE_MARK_WHITE	1.000	0.849	0.966	0.554
HOSEPCV_MARK_GREEN	0.929	0.933	0.88	0.453
HOSEPCV_MARK_YELLOW	0.963	0.809	0.952	0.508
INTAKE_BOLTLH	0.984	1.000	0.995	0.915
INTAKE_BOLTRH	0.990	1.000	0.995	0.909
INTAKE_SPECIALHYBRID	0.934	1.000	0.995	0.908
MOUNTING_PINTHREADTHREAD	0.986	1.000	0.995	0.908
MOUNTING_THREADPINTHREAD	0.963	1.000	0.995	0.860
PCV_BLACK	0.948	1.000	0.995	0.766
PCV_ORANGE	0.994	1.000	0.995	0.824
PULLEYWITHOUT_GROOVE	0.938	1.000	0.995	0.718
PULLEYWITH_GROOVE	0.989	1.000	0.995	0.840
SOCKET_BLACK	0.993	1.000	0.995	0.871
SOCKET_GRAY	0.964	1.000	0.995	0.866
STAMPLE_OK_TESTER	0.990	1.000	0.995	0.921
STICKOIL_BLACK	0.986	1.000	0.995	0.958
STICKOIL_GREEN	0.978	1.000	0.995	0.988
STICKOIL_PINK	0.943	1.000	0.995	0.914
STUD_1	0.972	0.977	0.979	0.648
STUD_2	0.977	0.981	0.986	0.728
VRIBBEDWITHOUT_HOLE	0.961	1.000	0.995	0.988
VRIBBEDWITH_HOLE	0.982	1.000	0.995	0.991
WITH_BRAKETCHC	0.989	1.000	0.995	0.995
Average	0.971	0.990	0.991	0.868
Standard Deviation	0.021	0.035	0.017	0.148

Table 3. Test results with the model

View Number	View Display Example	Number of Photos
1	 A photograph of an engine assembly on a production line. A green bounding box highlights a component labeled 'STAMPLE OK (PASTER) 0.94'. A purple bounding box highlights a component labeled 'MOUNTING PIN THREAD 0.94'.	32
2	 A photograph of an engine assembly. A blue bounding box highlights a component labeled 'VRIBBED WITH HOLE 0.99'.	32
3	 A photograph of an engine assembly. Three bounding boxes highlight components: 'SOCKET GRAY 0.94' (white), 'PULLEY WITH GROOVE 0.94' (blue), and 'BRACKET BLOATED 0.95' (blue).	32
4	 A photograph of an engine assembly. Three bounding boxes highlight components: 'AF SENSOR GREEN WIRE 0.96' (blue), 'STICK OIL BLACK 0.98' (blue), and 'BRACKET FACING DOWN 0.99' (purple).	32
5	 A photograph of an engine assembly. Three bounding boxes highlight components: 'CAP GREEN 0.96' (green), 'ORANGE 0.94' (orange), and 'STUD 2 TORX' (green).	32

6		32
7		32
8		32
9		32
10		32

or during overcast weather. These advancements highlight the model's potential for practical applications, such as on assembly lines or in computer vision-based automated inspection systems. With its high inference speed and precision, the model was well-suited for supporting real-time decision-making processes in the manufacturing sector.

However, it is important to note that the 100% accuracy rate is indicative of the model's performance within the scope of the test data's distribution. To fully assess the model's robustness, further testing under extreme conditions such as uncontrolled lighting, random rotations, and environmental noise remains necessary. These additional tests will help evaluate its performance in more unpredictable, out-of-distribution scenarios, ensuring its reliability in real-world field applications.

4.2. Effect of Grayscale and Blur on Shape-Based Component Detection

Evaluation of the YOLOv8-based object detection system on machine component images, focusing on shape, reveals that the application of grayscale and blur pre-processing techniques enhances the stability and clarity of the model's detection results. Based on the visual results shown in Figure 6, the model without pre-processing tends to experience instability in recognising object shape boundaries, especially in areas with visual disturbances or geometric irregularities. Components with uneven shapes, angular slopes, or complex contours are often not easily detected.

In contrast, the results shown in Figure 7 indicate that the application of grayscale as a pre-processing enhances the model's ability to recognise the contour and spatial structure of objects without being disturbed by the

surrounding colour variables. By eliminating colour information, the model focuses more on the shape patterns, slopes, and sharp or curved corners that are characteristic of shape-based parts. Meanwhile, blur is proven to be able to reduce local noise and minor artefacts that can interfere with the detection process, without obscuring the main features of the analysed object.

Quantitatively, the model's precision value increased from 0.821 to 0.973 after pre-processing, equivalent to an increase of 18.51%. Meanwhile, recall rose from 0.979 to 0.997, or an increase of 1.83%. This increase reflects an improvement in the model's ability to consistently detect shape-based objects while reducing misclassification under non-ideal visual conditions.

4.3. Effect of Brightness Adjustment and Blur on Colour-Based Component Detection

The YOLOv8 model was also evaluated on colour images with a focus on components that have dominant characteristics in colour elements. In conditions without pre-processing, as shown in Figure 8, the system experienced a decrease in detection stability, with fluctuating confidence values in parts of the image that have uneven lighting or visual disturbances due to reflections and vibrations. Good image quality can achieve 94% accuracy but low image quality can reduce accuracy by 10% to 50% [21]. Furthermore, the data without pre-processing indicates that the model is quite sensitive to lighting variations when no visual adjustments are given to the input image [26].

After applying brightness adjustment and blur, as shown in Figure 9, the detection performance has improved both in terms of prediction stability and object boundary clarity. Brightness adjustment helps equalise exposure between image samples, allowing critical visual



Figure 6. Model detection results without grayscale and blur pre-processing



Figure 7. Model detection results with grayscale and blur preprocessing.

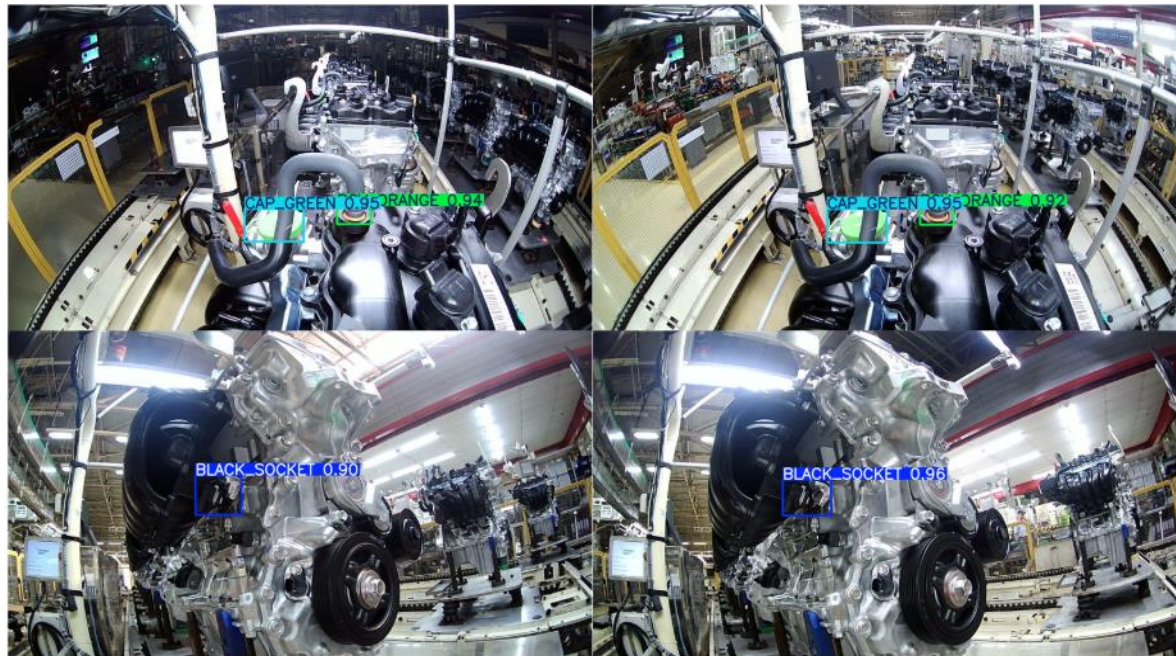


Figure 8. Model detection results without brightness and blur pre-processing



Figure 9. Model detection results with brightness and blur pre-processing

features to be consistently recognised by the model [18]. Meanwhile, blur plays a role in filtering out small, irrelevant details without eliminating the main characteristics of the object, thereby making the detection process more focused and precise.

The evaluation results show that the model's precision increased from 0.739 to 0.968, representing a 30.98% improvement. Recall also increased from 0.791 to 0.980, reflecting an increase of 23.89%. These values indicate that the applied pre-processing techniques have contributed to improving the accuracy and consistency of the model in detecting colour-based objects, which were previously more susceptible to visual distortion [18].

4.4. Implications for Quality Line Performance

The upgrade of the YOLOv8-based automated inspection system equipped with the latest dataset and visual pre-processing techniques has a positive impact on the smoothness of the inspection line process (quality line). The updated model can recognise more components with broader visual coverage and higher reliability. This robotic system in this research can now detect previously unrecognised parts, making the inspection process faster up to 2.0 minutes faster (ranging from 1.4 minutes to 2.0 minutes per engine check). Therefore, manual visual inspections which are usually carried out by humans and often experience errors can be replaced by robotic systems with higher accuracy [27].

With the system's increased ability to detect parts automatically, the operator's role in the inspection process becomes more focused on the final validation stage or checking components that require human intervention. This research finding contributes to accelerating the inspection flow because the next machine unit does not have to wait too long to be inspected. The distribution of workload between the automated system and the operator becomes more balanced, allowing for increased throughput on the production line.

Ultimately, it will save time, energy, and production costs, allowing the company to expand its profit margin or make product prices more competitive compared to its competitors in the market. The speed and broader detection coverage support the integration of this system

into the automation and digitalisation-based production ecosystem. The detection model, which is responsive to various visual conditions, strengthens the reliability of the inspection process while maintaining overall production efficiency. Thus, this system has the potential to be adopted more widely in industrial scenarios that demand high accuracy and seamless process continuity.

5. Conclusion

This study successfully improves the accuracy and robustness of the YOLOv8-based visual inspection system by employing an adaptive training strategy with dataset modification. Pre-processing techniques such as grayscale conversion, brightness adjustment, and blurring are applied to represent common visual disturbances often encountered in industrial environments. This approach significantly improves the model's detection performance in scenarios with unstable lighting and mechanical visual disturbances.

The evaluation results on 1,281 instances from 52 component classes show an average precision of 0.971, recall of 0.990, and mAP@50 of 0.991. The model's precision ranges from 0.929 to 1.000, while recall mostly reaches a maximum value of 1.000, indicating very high detection consistency. Although mAP@50 shows almost uniform performance, the mAP@50–95 values are more varied, ranging from 0.453 to 0.995 with a standard deviation of 0.149, reflecting the spatial challenges in detecting complex objects.

The effectiveness of dataset modification is reflected in the improvement in model performance in detecting shape and colour-based components. Grayscale and blur techniques increase precision from 0.821 to 0.973 and recall from 0.979 to 0.997, indicating an improvement in the model's ability to recognise object contours. In the colour-based component, brightness and blur adjustments increase precision from 0.739 to 0.968 and recall from 0.791 to 0.980, indicating the model's robustness to lighting fluctuations and light reflections.

Testing on ten robotic displays yielded a detection accuracy increase from an average of 72% to 100%, with no misclassifications, either false positives or false negatives. These results indicate that the model has generalised very well

to data that remains within the training distribution. However, it should be acknowledged that testing on extreme scenarios or new distribution conditions is still necessary to assess the model's robustness limits thoroughly.

Overall, this study demonstrates that integrating pre-processing methods into object detection model training not only enhances accuracy but also improves the stability and adaptability of the system to variations in visual conditions. These improvements have a direct impact on the efficiency of the inspection path, reducing the need for manual verification and accelerating overall production throughput. With these findings, the optimised YOLOv8-based visual detection system has the potential to be widely adopted in the implementation of automated inspections in the innovative manufacturing industry sector.

Acknowledgements

This research was supported by Universitas Bina Nusantara through the scheme of Penelitian Mandiri.

Author's declaration

Authors' contributions and responsibilities

SS: Conceptualization, Methodology, Formal Analysis, Investigation, Visualization. **SW:** Software, Formal Analysis, Data curation, Writing- Original draft preparation, Investigation. **AS:** Writing - Review & Editing, Formal Analysis, Funding acquisition, Project administration, Supervision. **EP:** Validation, Resources, Project administration, Supervision.

AI Declaration

During the preparation of this work, the authors used Grammarly (free version) to rephrase and check the grammar of the original manuscript. After using this tool, the authors reviewed and edited the content as needed and takes full responsibility for the content of the published article.

Conflict of Interest Disclosure

The authors have no conflicts of interest to declare.

Availability of data and materials

All data are available from the authors.

Competing interests

The authors declare no competing interest.

Additional information

No additional information from the authors.

References

- [1] S. Grzesiak and A. Sulich, "Car Engines Comparative Analysis: Sustainable Approach," *Energies*, vol. 15, no. 14, p. 5170, Jul. 2022, doi: 10.3390/en15145170.
- [2] M. Mazzetto, M. Teixeira, É. O. Rodrigues, and D. Casanova, "Deep Learning Models for Visual Inspection on Automotive Assembling Line," *Int. J. Adv. Eng. Res. Sci.*, vol. 7, no. 3, pp. 473–494, 2020, doi: 10.22161/ijaers.74.56.
- [3] U. Sirisha, S. P. Praveen, P. N. Srinivasu, P. Barsocchi, and A. K. Bhoi, "Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, p. 126, Aug. 2023, doi: 10.1007/s44196-023-00302-w.
- [4] M. Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection," *Machines*, vol. 11, no. 7, p. 677, Jun. 2023, doi: 10.3390/machines11070677.
- [5] M. Safaldin, N. Zaghdien, and M. Mejdoub, "An Improved YOLOv8 to Detect Moving Objects," *IEEE Access*, vol. 12, pp. 59782–59806, 2024, doi: 10.1109/ACCESS.2024.3393835.
- [6] X. Liu, C. Liu, X. Zhou, and G. Fan, "Enhancing low-light object detection with En-YOLO: leveraging dual attention and implicit feature learning," *Multimed. Syst.*, vol. 31, no. 3, p. 249, Jun. 2025, doi: 10.1007/s00530-025-01820-7.
- [7] Z. Sun et al., "An Anchor-Free Detection Method for Ship Targets in High-Resolution SAR Images," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 14, pp. 7799–7816, 2021, doi: 10.1109/JSTARS.2021.3099483.
- [8] M. Yeung, E. Sala, C.-B. Schönlieb, and L. Rundo, "Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation," *Comput. Med. Imaging Graph.*, vol. 95, p. 102026, Jan. 2022, doi: 10.1016/j.compmedimag.2021.102026.
- [9] X. Xu et al., "Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN," *Sensors*, vol. 22, no. 3, p.

- 1215, Feb. 2022, doi: 10.3390/s22031215.
- [10] D. Devi, S. Sophia, and S. R. Boselin Prabhu, "Deep learning-based cognitive state prediction analysis using brain wave signal," in *Cognitive Computing for Human-Robot Interaction*, Elsevier, 2021, pp. 69–84. doi: 10.1016/B978-0-323-85769-7.00017-3.
 - [11] D. Wan et al., "YOLO-MIF: Improved YOLOv8 with Multi-Information fusion for object detection in Gray-Scale images," *Adv. Eng. Informatics*, vol. 62, p. 102709, Oct. 2024, doi: 10.1016/j.aei.2024.102709.
 - [12] I. Zeger, S. Grgic, J. Vukovic, and G. Sisul, "Grayscale Image Colorization Methods: Overview and Evaluation," *IEEE Access*, vol. 9, pp. 113326–113346, 2021, doi: 10.1109/ACCESS.2021.3104515.
 - [13] J. Wang and S. Lee, "Data Augmentation Methods Applying Grayscale Images for Convolutional Neural Networks in Machine Vision," *Appl. Sci.*, vol. 11, no. 15, p. 6721, Jul. 2021, doi: 10.3390/app11156721.
 - [14] Y. Ge, Q. Zhang, Y. Sun, Y. Shen, and X. Wang, "Grayscale medical image segmentation method based on 2D&3D object detection with deep learning," *BMC Med. Imaging*, vol. 22, no. 1, p. 33, Dec. 2022, doi: 10.1186/s12880-022-00760-2.
 - [15] S. Wang, "Evaluation of impact of image augmentation techniques on two tasks: Window detection and window states detection," *Results Eng.*, vol. 24, p. 103571, Dec. 2024, doi: 10.1016/j.rineng.2024.103571.
 - [16] X. Liu, H. Liu, and D. Zhou, "Multi-scale defect detection on product surface in grayscale images using YOLOv8-Ms algorithm," *Nondestruct. Test. Eval.*, pp. 1–25, Mar. 2025, doi: 10.1080/10589759.2025.2478259.
 - [17] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.
 - [18] G.-J. Son, H.-C. Jung, and Y.-D. Kim, "Temporal-Quality Ensemble Technique for Handling Image Blur in Packaging Defect Inspection," *Sensors*, vol. 24, no. 14, p. 4438, Jul. 2024, doi: 10.3390/s24144438.
 - [19] M. D. Cirillo, D. Abramian, and A. Eklund, "What is The Best Data Augmentation For 3D Brain Tumor Segmentation?," in *2021 IEEE International Conference on Image Processing (ICIP)*, IEEE, Sep. 2021, pp. 36–40. doi: 10.1109/ICIP42928.2021.9506328.
 - [20] K. Chao, W. Song, S. Shao, D. Liu, X. Liu, and X. Zhao, "CUI-Net: a correcting uneven illumination net for low-light image enhancement," *Sci. Rep.*, vol. 13, no. 1, p. 12894, Aug. 2023, doi: 10.1038/s41598-023-39524-5.
 - [21] C.-T. Chien, R.-Y. Ju, K.-Y. Chou, E. Xieerke, and J.-S. Chiang, "YOLOv8-AM: YOLOv8 Based on Effective Attention Mechanisms for Pediatric Wrist Fracture Detection," *IEEE Access*, vol. 13, pp. 52461–52477, 2025, doi: 10.1109/ACCESS.2025.3549839.
 - [22] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
 - [23] G. M. Foody, "Challenges in the real world use of classification accuracy metrics: From recall and precision to the Matthews correlation coefficient," *PLoS One*, vol. 18, no. 10, p. e0291908, Oct. 2023, doi: 10.1371/journal.pone.0291908.
 - [24] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," 2014, pp. 740–755. doi: 10.1007/978-3-319-10602-1_48.
 - [25] L. He, Y. Zhou, L. Liu, W. Cao, and J. Ma, "Research on object detection and recognition in remote sensing images based on YOLOv11," *Sci. Rep.*, vol. 15, no. 1, p. 14032, Apr. 2025, doi: 10.1038/s41598-025-96314-x.
 - [26] Z. Yuan et al., "Research on Calf Behavior Recognition Based on Improved Lightweight YOLOv8 in Farming Scenarios," *Animals*, vol. 15, no. 6, p. 898, Mar. 2025, doi: 10.3390/ani15060898.
 - [27] N. Hütten, M. Alves Gomes, F. Hölken, K. Andricevic, R. Meyes, and T. Meisen, "Deep Learning for Automated Visual Inspection in Manufacturing and Maintenance: A Survey of Open- Access Papers," *Appl. Syst. Innov.*, vol. 7, no. 1, p. 11, Jan. 2024, doi: 10.3390/asi7010011.