TIERS Information Technology Journal
Vol.6, No.1, June 2025, pp. 74~86
P-ISSN: 2723-4533 / E-ISSN: 2723-4541
DOI: https://doi.org/10.38043/tiers.v6i1.6499

Comparative Analysis of YOLOv5n and YOLOv8n Deep Learning Models for Precision Detection of *Klowong* Defects in Batik Fabric

Rifqi Restu Hamidi[1], Muhammad Kusumawan Herliansyah[2], Denny Sukma Eka Atmaja[3], Andi Sudiarso[4]
rifqirestuhamidi@mail.ugm.ac.id[1], herliansyah@ugm.ac.id[2], denny.sukma.e@mail.ugm.ac.id[3], dennysukma@telkomuniversity.ac.id[3], a.sudiarso@ugm.ac.id[4]
[1,2,3,4]Department of Mechanical and Industrial Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia
[3]School of Industrial and Systems Engineering, Telkom University, Bandung, 40257, Indonesia

ABSTRACT

This study presents a comparative analysis of two deep learning object detection models, YOLOv5n and YOLOv8n, for the precies identification of *Klowong* defects in batik fabric. The evaluation was carried out using a custom dataset consisting of 3,138 annotated images, with 921 allocated for testing and containing 1,295 defect instances across nine defect classes. The main findings show that YOLOv8n outperforms YOLOv5n in both speed and accuracy. YOLOv8n achieved a higher F1-score of 0.87 at a lower confidence threshold (0.297), compared to YOLOv5n's F1-score of 0.86 at a higher threshold (0.46). In addition, YOLOv8n reduced training time significantly (0.320 hours vs. 0.868 hours) and delivered much faster inference speed (2.9 ms/image), nearly three times quicker than YOLOv5n. Although both models performed well in detecting common defects, YOLOv8n showed more stable results on complex defect types. These improvements make YOLOv8n more suitable for real-time applications in batik production environments. Its efficiency and accuracy support the development of fast and reliable automated quality control systems in traditional textile industries. This research emphasizes the importance of using modern lightweight architectures like YOLOv8n to enhance defect detection performance in practical manufacturing settings.

Keywords: Batik; Deep Learning; *Klowong;* YOLOv5n; YOLOv8n

Article Info

Received : 15-02-2025
Revised : 11-04-2025
Accepted : 25-06-2025

This is an open-access article under the CC BY-SA license.

Correspondence Author:

Muhammad Kusumawan Herliansyah
Department of Mechanical and Industrial Engineering,
Universitas Gadjah Mada,
Jl.Grafika No.2, Yogyakarta 55281, Indonesia.
Email: herliansyah@ugm.ac.id

1. INTRODUCTION

Batik was officially recognized by UNESCO in 2009 as a Masterpiece of the Intangible and Oral Heritage of Humanity, emphasizing its importance in the context of Indonesian culture [1], [2]. Since the recognition, the Batik industry has experienced significant growth and has become a vital component of the national creative economy [1]. Beyond national significance, batik has also gained international appreciation as a textile art form characterized by the use of dyes and wax to create intricate patterns and culturally meaningful designs [3], [4]. The making of batik not only requires artistic craftsmanship but also represents the philosophical and symbolic identity of the communities that produce it [5], [6].

Tradionally, batik is created by applying hot wax onto the fabric using tools such as *canting tulis* and *canting cap* [3]. Based on the production technique, batik is generally classified into two types: hand-drawn batik (*batik tulis*), which is produced manually with *canting*, and stamped batik (*batik cap*), which is made using stamping machines for mass production [2]. According to UNESCO, authentic Indonesian batik is defined as cloth whose motifs are created manually using hot wax and applied with the hand using a *canting* [2].

The batik-making process typically consists of four main stages: (1) designing the motif, either based on traditional heritage or modern innovation [7], (2) applying hot wax in the *klowong* process, which is the most time-consuming and accuracy-demanding phase [8], (3) initial coloring, and (4) the final *nembok* process that ensures color separation and motif clarity [7], [8]. Among these, the *klowong* process is particularly labor-intensive and requires high precision in following the pattern lines, which often poses a significant bottleneck in production time [9].

The *klowong* is the process of applying wax to the patterns that have been previously drawn with pencil on the fabric. The *klowong* process constitutes a pivotal phase in traditional batik craftsmanship, characterized by the selective application of molten wax onto designated fabric regions that have been previously outlined with preliminary pencil markings, serving as a resist medium to prevent dye penetration in subsequent coloring stages [10]. The selection of appropriate wax compositions is paramount, typically involving a blend of beeswax, paraffin wax, and dammar resin, each contributing unique properties such as melting point, flexibility, and adhesion to the fabric [11]. One example of the *klowong* process is shown in Figure 1.

Figure 1. *Klowong* Process

One of the most critical challenges in hand-drawn batik production is the manual defect inspection, especially during the *klowong* stage [1]. In this context, a defect refers to an unwanted anomaly or irregularity that occurs during the batik-making process, potentially degrading the visual quality or integrity of the motif. These defects may stem from human error, equipment issues, or material inconsistencies.

In particular, nine types of *klowong* defects are frequently encountered during inspection such as defect 1 (oil stain), defect 2 (hole), defect 3 (broke yarn), defect 4 (fabric penetration), defect 5 (line continuity), defect 6 (pattern corner quality), defect 7 (line thickness continuity), defect 8 (line deviation), and defect 9 (droplet) [1]. These defects disrupt motif quality and can significantly impact the aesthetic and commercial value of the final product. Therefore, detecting these flaws accurately during production is essential to maintaining quality standards.

Furthermore, this process is not only time-consuming but also prone to subjective human judgment, leading to inconsistencies and potential quality issues [1]. Furthermore, errors occurring at this stage often result in rework or fabric rejection, thereby reducing overall production efficiency [7], [9]. To address this, some batik manufacturers have begun implementing CNC technology to accelerate the *klowong* process while preserving the essence of traditional batik-making [12].

The traditional batik industry continues to face broader challenges in balancing production efficiency, product quality, cultural preservation, and market responsiveness [13]. These include the need for effective quality control systems, cost-efficient production methods, and sustainable integration of modern technology. In this context, the development of automated inspection systems is essential to ensure consistent quality and minimize waste. Automatic detection can serve as an early warning mechanism, identifying defects before the *klowong* process is completed, thus reducing the likelihood of production failure [1].

Recents advancements in textile inspection technologies have introduced high-precision fabric defect detection systems for complex and high-variation fabric textures [14]. In response to this, the present study proposes the development of an automated detection model with high accuracy and efficiency, specifically targeted for implementation at the Batik Butimo center. This model is integrated into a real-time defect detection system designed to monitor the *klowong* process, providing immediate feedback to production operators [15], [16]. By incorporating this system, the batik production process at Butimo is expected to become more standardized, productive, and reliable.

In the study presented by [17], the researchers addressed the challenge of detecting fabric defects, highlighting a key issue such as methods that perform well for large defects often fail with small ones, and vice versa. This reciprocal contradiction is particularly difficult to resolve using a single detection method, especially when dealing with colored fabrics [17]. To tackle this, the study introduced a hybrid approach that combines two architectures—Squeeze and Excitation Network (SE-Net) and Single Shot Multibox Detector (SSD) [17]. The SE module is applied to enhance the model's focus on feature channels rich in defect-related information, while the SSD architecture retains its original function of extracting multi-scale feature maps [17]. Experimental results demonstrated that this combined method is capable of identifying six different defect types in colored fabrics, effectively meeting practical application requirements [17].

Subsequent research by [18] explores an efficient and rapid approach for fabric defect detection. The YOLO (You Only Look Once) family of models continues to hold a prominent role in object detection tasks [18]. To enhance performance, the study incorporates several improvements, including a combined data augmentation strategy to increase dataset diversity and enhance model robustness [18]. Additionally, the researchers applied the k-means clustering algorithm to the ground truth bounding boxes in order to generate optimal anchor boxes tailored for fabric defect detection [18]. A new prediction layer was also added to the *yolo_head* component, aiming to improve the detection of small-sized defects [19]. Moreover, the study [19] integrated the Convolutional Block Attention Module (CBAM) into the model's backbone to refine feature extraction and emphasize defect-relevant information [19]. The experimental results show that compared with the original YOLOv4 algorithm, the detection accuracy of the improved YOLOv4 algorithm for small targets is significantly improved, with the Average Precision (AP) value for small target detection increased by 12% and the overall average precision (mAP) increased by 3%. The prediction results of the proposed algorithm can provide more accurate defect positions for enterprises, reduce the defect rate of fabric products, and improve their economic impact [19].

Detecting defects in fabrics or batik presents a significant challenge in vision-based quality inspection systems, particularly when the defects are small and embedded within complex backgrounds such as batik patterns. Various YOLO-based object detection models have been applied in this domain, but their effectiveness in detecting tiny objects heavily depends on architectural enhancements and optimization strategies. YOLOv8 has shown strong performance in general textile defect detection, offering improved energy efficiency, higher accuracy, and faster inference compared to its predecessors [20]. Recent studies have introduced significant improvements to YOLOv5n and YOLOv8n lightweight variants designed for real-time applications with limited computational resources. For instance, YOLOv8n incorporates a new backbone architecture based on C2f modules, streamlined model heads, and enhanced label assignment strategies, all contributing to superior performance in both speed and detection accuracy. YOLOv5n, while earlier, introduced anchor-free detection and optimized training strategies that improved model robustness in constrained environments [18].

While numerous lightweight object detectors exist such as YOLOv7-Tiny, YOLOv6-Nano, and MobileNet-SSD, the current study focuses on benchmarking YOLOv5n and YOLOv8n due to their balance between simplicity, accessibility, and proven effectiveness in prior industrial applications. These models are also supported by active development communities and integrated into widely used open-source frameworks, which ensures reproducibility and maintainability for industrial deployment. The decision to exclude other lightweight detectors was made based on preliminary assessments, where YOLOv5n and YOLOv8n showed superior inference speeds and accuracy trade-offs in textile, relevant tasks while maintaining model compactness [21], [22]. Although models like YOLOv7-Tiny and MobileNet-SSD were initially considered, they were ultimately excluded due to either lower detection consistency in early tests or limited support for streamlined integration into the batik defect detection pipeline.

Despite advancements in YOLO-based detection, no previous studies have explicity benchmarked lightweight YOLO architectures within the specific visual context of *Klowong* batik defects, characterized by intricate, highly variable patterns and subtle color gradients. In response to this gap, this study presents the first direct architectural benchmark between YOLOv5n and YOLOv8n for detecting *Klowong* batik defects. The goal is to evaluate detection performance in terms of precision, inference speed, and training efficiency. Unlike general textile studies, this work focuses on domain-specific implementation, where traditional batik motifs require fine-grained feature discrimination and strong generalization across diverse pattern complexities. This explicit comparison contributes novel insights into the applicability of modern object detection architectures for quality control in heritage textile production, bridging computer vision with cultural preservation efforts [23], [24], [25].

2. RESEARCH METHOD

2.1. Data Collection

The object used in this study is a collection of handcrafted batik products produced using CNC machines by artisans at Batik Butimo, located in Bantul, Yogyakarta. The dataset acquisition process utilized the following tools; Fujifilm camera with a 35 mm lens, tripod, a 2.5 m x 1.15 m batik frame, measuring tape, ,60 mm paper clips to secure the fabric during photography, and a Dell Latitude 7420 laptop. The dataset consisted of 44 handcrafted batik klowong motifs as the primary research material. These were photographed under controlled lighting and distance settings to ensure consistency across all samples. Annotation was conducted using the Labelimg software, while model training and testing were performed on Google Colab.

.

The process begins with the digital camera is positioned perpendicular to the fabric using a certain predetermined distance so that the image results are consistent between samples. Each batik motif is photographed from several angles and distances so that the variety of images can capture possible defects that are not visible from just one angle. The whole process is systematically carried out and controlled to ensure that each image meets the quality standards required for training the defect detection model. This data becomes the basis for labeling and annotation as input to the YOLO model training process.

The dataset consists of 3.138 images of batik fabrics that have been produced by Batik Butimo. Each image is acquired under customized lighting conditions, and then the types of defects such as pattern breaks, stains, or motif line mismatches are identified. Subsequently, defect identification was conducted, focusing on flaws in the *klowong* process, using a deep learning approach, particularly computer vision, by implementing the You Only Look Once (YOLO) architecture and Design of Experiments (DOE).

To prepare the dataset for training, each image was annotated manually using bounding boxes to mark the location and type of defect. The annotation process was performed manually, with 1 to 3 defects annotated per image, depending on the presence and clarity of visual anomalies. Rather than free-hand labeling, the annotations were guided by predefined pattern references (batik motif outlines), enabling consistent placement of bounding boxes in alignment with the *klowong* batik design. This structured annotation method reduced subjectivity and ensured that the defect labeling closely followed the intended motif contours. The use of pattern-based annotation contributed to generating a reliable and representative ground truth dataset for training and evaluating the detection model.

2.2. Preprocessing

After collecting the data, the preprocessing stage in this study began with the acquisition of raw image data. A total of 3,138 images of the *klowong* batik process were collected, each with a standardized resolution of 640 x 640 pixels to ensure uniformity in subsequent processing stages. One example of the collected image is shown in Figure 1.

Figure 1. The example of *Klowong* Process Image

The preprocessing stage involved several key steps. First, image standardization was applied to ensure consistent dimensions across the dataset. Next, a data cleaning process was performed to remove low-quality images such as those that were blurry, underexposed, or irrelevant. Following this, a crucial step of defect editing was carried out on the entire dataset. This editing process involved manually annotating defective areas such as oil stains, ink bleeds, or pattern irregularities by using bounding boxes to mark regions of interest. There are nine defect classes that have been created for the data augmentation process [1].

To enhance model generalization and improve robustness agains variations in image orientation and lighting, several augmentation techniques were applied. These include HSV color augmentation to adjust hue, saturation, and value; horizontal and vertical flipping to simulate mirrored defects; random rotation; saling; brightness adjustment; and copy-paste augmentation, which introduces synthetic defect instances by duplicating and repositioning defect regions within the same image. The application of these augmentations aims to simulate diverse real-word conditions and reduce overfitting by exposing the model to a wider range of defect appearances during training [1].

These nine defects commonly found in batik fabrics include defect 1 (oil stain), defect 2 (hole), defect 3 (broke yarn), defect 4 (fabric penetration), defect 5 (line continuity), defect 6 (pattern corner quality), defect 7 (line thickness continuity), defect 8 (line deviation), and defect 9 (droplet) [1]. Defect 1, also known as *mbleber,* refers to the spread of dye or batik ink beyond the intended boundaries of the motif, resulting in blurred and imprecise patterns that diminish the visual sharpness and aesthetic of the design. Defect 2 consists of small holes or tears on the fabric surface, typically caused by mechanical damage during handling or production, which disrupt the batik pattern and appear as empty or damaged spots. Defect 3, or broken yarn, occurs when threads within the woven fabric break, leading to uneven surfaces and distortion of the batik pattern, which may compromise both the strength and appearance of the fabric. Defect 4, referred to as fabric penetration, involves unwanted transparency or light passing through due to thin or weakened areas in the cloth, which lowers both the visual and structural quality. Defect 5 pertains to discontinuity in batik lines, where broken or inconsistent line flow disrupts the pattern's integrity and compromises the design's harmony.

Defect 6, or poor pattern corner quality, describes imprecise or untidy motif corners that lack sharpness or become overly rounded, thereby reducing the visual definition of the pattern. Defect 7 involves inconsistency in line thickness across the motif, where irregularities such as lines being too thick or thin can create an unbalanced appearance and reduce the overall aesthetic quality. Defect 8, known as line deviation, occurs when motif lines deviate from their intended path, resulting in asymmetrical patterns that impair the design's precision. Lastly, defect 9, called droplet or *netes*, appears as unintended small spots or ink droplets that stain the fabric, disrupting the motif and making the cloth appear dirty or flawed.

Furthermore, visual improvements like enhancing contrast in defect regions were implemented to help the object detection model better identify these areas. To expand and diversify the dataset, additional augmentation methods such as rotation, flipping, zooming, and brightness modification were utilized. These techniques were intended to boost the model's robustness by providing a wider range and greater volume of training data.

2.3. Labeling Process (Bounding Box)

The labeling process in this study involved annotating each defect in the batik images using bounding boxes to specify their position and class. This labeling stage is essential for training object detection algorithms like YOLO (You Only Look Once). The labeling was performed using annotation software like 'Labeling', which generates a .txt file for each image. The labeling was carried out manually because approximately half of the defect classes are identified based on pattern characteristic rather than distinct shapes, requiring human judgement to ensure accurate annotation.

These files contain bounding box information in YOLO format, structured as: class_id; x_center; y_center; width, and height. Here, class_id refers to the class index of the defect based on the training class list. The x_center and y_center represent the normalized coordinates of the box center, divided by the image width and height respectively, producing values between 0 and 1. Similarly, the width and height of the bounding box are also normalized. The following shows examples of the nine defects that have undergone the labeling process.

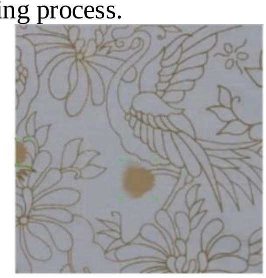

Figure 2(a). Defect 1

Figure 2(b). Defect 2

Figure 2(c). Defect 3

Figure 2(d). Defect 4

Figure 2(e). Defect 5

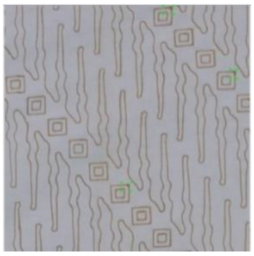

Figure 2(f). Defect 6

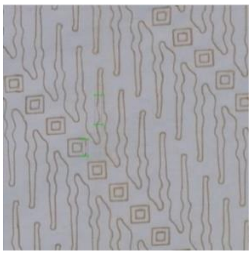

Figure 2(g). Defect 7

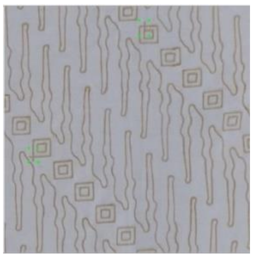

Figure 2(h). Defect 8

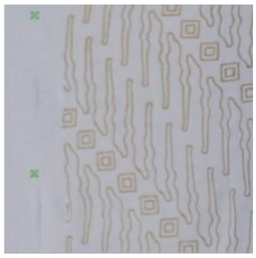

Figure 2(i). Defect 9

At this stage, the annotated dataset is divided into three parts, namely taining data, validation data, and testing data. The total dataset consisted of 3.138 batik images. From this, a subset of 2.217 images was allocated for training and validation purposes using an 80 : 20 split. Specifically, 80% of the 2.217 images were used for model training, while the remaining 20% were used for validation. Training data is used to train the model to recognize patterns and important features of defects in batik cloth, validation data is used to monitor features of defects in batik cloth, validation data is used to monitor the performance of the model during training to prevent overfitting. On the other hand, the test dataset was reserved for the final evaluation phase after model training was completed. A total of 921 images were specifically allocated for testing purposes to measure how well the model generalizes to new data. This separation guarantees that the test images remain entirely unseen during both training and validation, enabling an unbiased evaluation of the model's performance in real-world conditions.

2.4. Model Architecture

The structure of an object detection model significantly influences its performance in terms of accuracy, speed, and adaptability to different use cases. This research explores two cutting-edge models such as YOLOv5n and YOLOv8n selected for their notable improvements in both detection efficiency and precision. Both models employ convolutional neural network backbones optimized for feature extraction, but they differ in architectural design and optimization strategies. Understanding the structural differences and the components of these models is essential to analyze their comparative performance effectively.

Furthermore, while the section on model training and architecture provides a clear description of the differences between YOLOv5n and YOLOv8n, it is also important to elaborate on the rationale behind selecting these particular models [26], [27]. YOLOv5n and YOLOv8n represent lightweight yet high-performing architectures that are well-suited for real-time detection tasks, making them highly relevant for detecting fine-grained defects in batik fabrics, which often contain intricate patterns and subtle imperfections. Their balance between model size, inference speed, and detection accuracy aligns well with the operational constraint typically found in textile manufacturing environments, such as limited hardware resources and the need for rapid inspection [28]. By evaluatin both models, this study aims to identify the most effective solution for automated batik defect detection in practical applications.

2.5. Model Training and Architecture

The initial training process of the YOLO (You Only Look Once) models, specifically YOLOv5n and YOLOv8n, was conducted to evaluate their performance under different optimization schemes. Both models were trained on the same dataset with consistent hyperparameter settings to ensure a fair comparison. The input image size (mg) was set to 640 pixels, batch size to 32, and the number of training epochs to 100.

2.6. Optimizer Configuration

A notable distinction between the two models lies in their respective optimizers, which were not manually modified but rather retained as part of each model's default configuration. YOLOv5 utilizes the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01, while YOLOv8n adopts AdamW with a learning rate of 0.000768. This difference arises from the intention to evaluate each model in its original, unaltered state, thereby reflecting the design choices made by the respective developers. YOLOv5's use of SGD, a classical optimizer that leverages momentum and learning rate schedules, contrast with YOLOv8n's adoption of AdamW, an adaptive optimizer known for its ability to decouple weight decay from the gradient update and improve generalization. By maintaining the default settings, this study aims to compare the performance of both models as they are typically deployed out-of-the-box, offering a fair and practical benchmark for real-world applications.

The rationale for retaining the default optimizer settings is to ensure a fair and practical benchmark that reflects the standard behavior of each model when deployed out-of-the-box. These default configurations are typically selected and fine-tuned by the developers to suit the underlying architecture. Additionally, by maintaining these defaults, this study avoids the introduction of bias from manual hyperparameter tuning, allowing for a clearer assessment of how each model performs under its intended optimization strategy. This distinction in optimizer design can influence convergence speed, stability during training, and ultimately the generalization performance on unseen data.

2.7. Backbone Architecture

The backbone network, responsible for feature extraction in both models, differs significantly in design and complexity. YOLOv8n utilizes a lightweight and efficient backbone based on the Cross Stage Partial Network (CSPDarknet) architecture. CSPDarknet is designed to reduce computational cost while maintaining strong representational power, optimizing the balance between inference speed and detection accuracy. YOLOv5n, on the other hand, employs CSPDarknet53, an extension of the original Darknet53 architecture used in YOLOv3. CSPDarknet53 introduces cross-stage partial connections to enhance gradient flow and reduce computational redundancy, resulting in a deeper and more complex network that is capable of extracting more detailed feature representations.

2.8. Evaluation Metrics

The evaluation of model performance is based on several key metrics such as Precision, Recall, mAP50, and mAP50-95. Precision indicates how accurately the model identifies true positives among all predicted detections, while Recall measures the model's capability to detect all actual objects within the dataset. The mAP50 metric, or mean Average Precision at an IoU threshold of 0.5, reflects the average detection accuracy when predictions overlap the ground truth by at least 50%. Meanwhile, mAP50-95 extends this by averaging precision across IoU thresholds ranging from 0.5 to 0.95, offering a more thorough and rigorous assessment. These metrics are essential for evaluating and comparing the detection performance of YOLOv5n and YOLOv8n models.

3. RESULTS AND DISCUSSION

3.1. Base Model Validation of YOLOv5n

The validation of the base YOLOv5n model was performed to assess its effectiveness in detecting multiple defect classes within the test dataset. The dataset comprised 443 images containing a total of 728 instances of nine distinct defect types. Model evaluation metrics including Precision, Recall, mean Average Precision at IoU threshold 0.5 (mAP50), and mean Average Precision across IoU thresholds 0.5 to 0.95 (mAP50-95) were computed for each defect class as well as overall. These metrics provide insight into the model's detection accuracy, sensitivity, and localization performance across varying degrees of overlap between predicted and ground truth bounding boxes. Figure 3 shows the detailed validation results for each defect class, highlighting both strengths and weaknesses of the model's detection capability.

Class	Images	Instances	P	R	mAP50	mAP50-95: 1
all	443	728	0.89	0.837	0.872	0.493
Defect1	443	81	0.994	1	0.995	0.853
Defect2	443	59	0.992	1	0.995	0.757
Defect3	443	72	0.948	0.875	0.931	0.593
Defect4	443	59	0.85	0.78	0.823	0.51
Defect5	443	83	0.84	0.904	0.848	0.406
Defect6	443	85	0.877	0.765	0.863	0.322
Defect7	443	104	0.909	0.75	0.857	0.336
Defect8	443	86	0.789	0.593	0.662	0.24
Defect9	443	99	0.811	0.869	0.871	0.422

Figure 3. Base Model Validation of YOLOv5n

Among the individual classes, Defect1 and Defect2 exhibited the highest detection performance, both achieving near-perfect Recall (1.0) and Precision values above 0.99, with mAP50 scores of 0.995. This suggests that the model is highly effective at detecting these defect types with minimal error.

In contrast, Defect 8 stands out as the class with the lowest performance metrics, having a Precision of 0.789, Recall of 0.593, and an mAP50 of 0.662. This indicates that the model struggled to accurately detect Defect8 instances, likely due to more subtle or complex visual characteristics. Additionally, Defect 8 records the lowest mAP50-95 score of 0.24 among all classes, highlighting the particular difficulty in achieving precise localization and classification for this defect type.

The F1-score, representing the harmonic mean of Precision and Recall, averaged across all classes, reached its highest value of 0.86 at a confidence threshold of 0.460. This indicates a balanced trade-off between precision and recall for the overall model performance.

In summary, the base YOLOv5n model demonstrates strong overall detection capabilities with excellent performance on some defect classes, while indicating potential areas for enhancement in detecting certain defects, especially Defect 8, which exhibited the lowest precision and recall values.

3.2. Test Base Model of YOLOv5n

The test results of the base YOLOv5n model were obtained on an extended dataset comprising 921 images and 1295 defect instances across the same nine defect classes. Figure 4 presents the performance metrics including Precision (P), Recall (R), mAP50, and mAP50-95 for each defect class and overall.

Class	Images	Instances	P	R	mAP50	mAP50-95:
all	921	1295	0.855	0.757	0.793	0.44
Defect1	921	135	0.994	0.993	0.995	0.811
Defect2	921	143	0.973	0.999	0.992	0.734
Defect3	921	136	0.925	0.813	0.866	0.54
Defect4	921	139	0.89	0.806	0.855	0.502
Defect5	921	122	0.876	0.934	0.941	0.449
Defect6	921	149	0.786	0.395	0.468	0.15
Defect7	921	153	0.84	0.651	0.719	0.251
Defect8	921	110	0.65	0.545	0.566	0.177
Defect9	921	208	0.763	0.679	0.735	0.345

Speed: 0.3ms pre-process, 7.4ms inference, 2.2ms NMS per image at shape (8, 3, 640, 640)

Figure 4. Test Base Model of YOLOv5n

Comparing the test results with the previous validation outcomes reveals several notable changes. The overall Precision decreased slightly from 0.89 to 0.855, while Recall experienced a more pronounced drop from 0.837 to 0.757. This reduction in Recall indicates a decrease in the model’s ability to detect all relevant defect instances on the larger test set, potentially due to increased data variability or more challenging samples. The mAP50 value also dropped from 0.872 in validation to 0.793 during testing, reflecting a decline in the model’s average detection accuracy at the 0.5 IoU threshold. Similarly, mAP50-95 decreased from 0.493 to 0.44, indicating less robust performance across stricter localization thresholds.

At the class level, several defect types showed significant performance shifts. For example, Defect 6 exhibited a dramatic decrease in Precision (from 0.877 to 0.786) and Recall (from 0.765 to 0.395), leading to a sharp drop in mAP50 from 0.863 to 0.468. This suggests the model struggles considerably with this defect under testing conditions. Defect 8 also showed further degradation, with Precision dropping to 0.65 and Recall to 0.545, reaffirming its status as one of the most challenging defect categories to detect.

These performance variations underscore the importance of evaluating models on sufficiently diverse and comprehensive datasets to capture real-world complexities. The test results provide valuable insights for further model improvements, particularly targeting defect classes with significant declines in detection metrics.

Figure 5 and Figure 6 shows the defect detection results obtained from the proposed model compared to the original labeled data. These visualizations demonstrate the model’s capability in accurately identifying and localizing various defect types on batik fabric samples. The predicted bounding boxes and confidence scores are presented alongside the ground truth annotations for direct comparison.

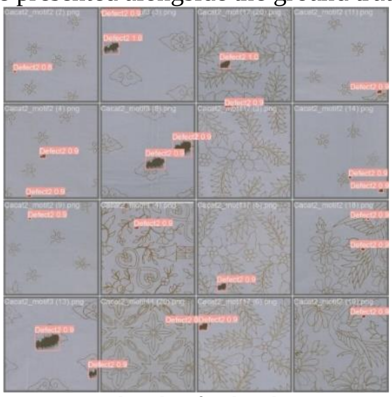

Figure 5. Predicted Defect by The YOLOv5n

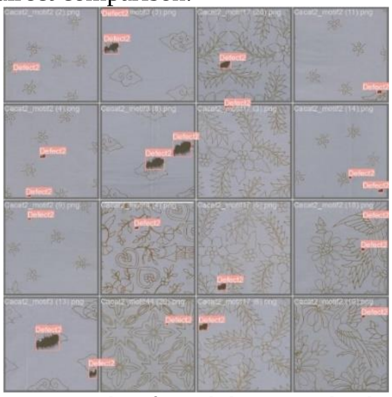

Figure 6. Truth Defect Labeling on Batik Fabric

As shown in Figure 5, the model successfully detected multiple defect instances with varying confidence levels, demonstrating its ability to handle different defect classes and complexities. The predicted bounding boxes closely align with the ground truth labels displayed in Figure 6, indicating reliable localization accuracy. Some minor discrepancies in bounding box size and positioning are observed, which may be attributed to the inherent variability in defect shapes and patterns. Overall, these qualitative results confirm that the model effectively generalizes to real batik fabric images and is capable of precise defect detection, supporting its practical application for automated quality control.

3.3. Base Model Validation of YOLOv8n

The baseline YOLOv8n model was validated to evaluate its ability to accurately detect various defect classes within the test dataset. This dataset included 443 images, encompassing a total of 728 annotated instances across nine different defect categories. Performance metrics such as Precision, Recall, mean Average Precision at an IoU threshold of 0.5 (mAP50), and mean Average Precision across a range of IoU thresholds from 0.5 to 0.95 (mAP50-95)—were calculated for each defect type individually as well as in aggregate. These measurements offer a comprehensive understanding of the model’s accuracy, sensitivity, and effectiveness in localizing defects at different levels of prediction-ground truth overlap.

For instance, while YOLOv8n demonstrates improved F1 scores and faster inference time compared to YOLOv5n, it is important to consider the trade-offs between precision, recall and inference speed. A higher precision may lead to fewer false positives but could reduce recall if the model becomes too conservative in making predictions. Conversely, optimizing for recall might increase false positives. In real world applications, such as real-time defect detection in textile production, these trade-offs must be carefully balanced to ensure optimal performance under operational constraint. Figure 7 presents the complete validation results by defect type, emphasizing the model’s strengths and areas that require improvement.

Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	443	728	0.868	0.867	0.888	0.533
Defect1	59	81	0.989	1	0.995	0.867
Defect2	46	59	0.943	1	0.989	0.785
Defect3	51	72	0.895	0.931	0.959	0.645
Defect4	32	59	0.773	0.864	0.878	0.604
Defect5	42	83	0.854	0.904	0.873	0.46
Defect6	41	85	0.887	0.835	0.887	0.379
Defect7	54	104	0.821	0.837	0.833	0.386
Defect8	41	86	0.804	0.698	0.766	0.27
Defect9	55	99	0.849	0.736	0.812	0.401

Speed: 0.2ms preprocess, 1.2ms inference, 0.0ms loss, 1.8ms postprocess per image

Figure 7. Base Model Validation of YOLOv8n

Among all defect classes, Defect 1 and Defect 2 exhibited the highest detection performance. Both classes achieved Recall values of 1.0 and Precision values above 0.94, with mAP50 scores of 0.995 and 0.989, respectively. This indicates that YOLOv8n is highly reliable in identifying these types of defects with minimal misclassification. Conversely, Defect 4 showed the lowest overall performance, with a Precision of 0.773 and a Recall of 0.864. Although the Recall remains reasonably high, the mAP50 dropped to 0.878 and the mAP50-95 to 0.604, making it the weakest among the higher-performing classes. This suggests that while the model detects most instances of Defect 4, the localization or bounding box accuracy might still require refinement.

Additionally, Defect8 also displayed relatively weaker results compared to other classes, with a lower Recall of 0.698 and the lowest mAP50-95 score of 0.270. This further highlights the need for improvement in detecting more visually complex or ambiguous defect types. The F1-score, calculated as the harmonic mean of Precision and Recall across all classes, achieved a maximum value of 0.87 at a confidence threshold of 0.297. This reflects the optimal balance point for classification performance, affirming that the model performs most effectively around this confidence setting.

In conclusion, the YOLOv8n base model demonstrates strong and efficient detection capabilities, particularly for certain defect types, while also revealing areas for improvement—most notably in the detection precision and bounding box localization of Defect 4 and Defect 8. These two defect types present specific challenges for the model due to their subtle visual characteristics and high variability in shape, size, and texture. For instance, Defect 4 often appears with blurred or irregular boundaries that resemble natural batik motifs, making it difficult to distinguish from non-defective areas. Meanwhile, Defect 8 tends to have small sizes and low contrast against the surrounding pattern, causing the model to miss or inaccurately localize the defect. This explains why both models struggle with these classes, despite overall strong performance.

3.4. Test Base Model of YOLOv8n

The test results of the base YOLOv8n model were obtained on an extended dataset comprising 921 images and 1295 defect instances across the same nine defect classes. Figure 8 presents the performance metrics including Precision (P), Recall (R), mAP50, and mAP50-95 for each defect class and overall.

Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 1
all	921	1295	0.879	0.82	0.854	0.508
Defect1	108	135	1	1	0.995	0.844
Defect2	117	143	0.987	1	0.995	0.792
Defect3	108	136	0.921	0.985	0.975	0.657
Defect4	91	139	0.908	0.785	0.857	0.501
Defect5	80	122	0.866	0.902	0.907	0.513
Defect6	98	149	0.833	0.604	0.711	0.273
Defect7	94	153	0.831	0.676	0.759	0.337
Defect8	72	110	0.816	0.773	0.746	0.283
Defect9	121	208	0.749	0.654	0.744	0.377

Speed: 0.4ms preprocess, 1.6ms inference, 0.0ms loss, 0.9ms postprocess per image
Results saved to **runs/detect/val2**
Fold 3 - mAP@0.5 : 0.8544
Fold 3 - mAP@0.5:0.95 : 0.5083
Fold 3 - Precision : 0.8791
Fold 3 - Recall : 0.8198

Figure 8. Test Base Model of YOLOv8n

Compared to the validation stage, the overall Precision slightly improved from 0.868 to 0.879, suggesting that the model produced fewer false positives on the larger test set. However, overall Recall dropped from 0.867 to 0.82, indicating a slight decline in the model's ability to detect all true defect instances. Similarly, mAP50 decreased from 0.888 to 0.854, and mAP50-95 fell from 0.533 to 0.508, highlighting a modest decline in both standard and stringent localization accuracy.

At the class level, Defect 1 and Defect 2 maintained high performance, with near-perfect Recall and mAP50 scores consistent across both validation and testing, confirming the model's robustness in detecting these defect types. However, Defect 6 and Defect 8 continued to demonstrate limited performance, with mAP50-95 values of 0.273 and 0.283, respectively—consistent with their lower validation scores, which suggests persistent difficulty in detecting these classes due to complexity or visual ambiguity.

Notably, Defect 4 showed improved performance, with Recall rising from 0.864 to 0.785 and mAP50 increasing from 0.878 to 0.857. Although slight, this improvement may reflect better generalization on new

.

test data. However, Defect 9 exhibited a drop in both Precision and Recall, leading to decreased mAP50 and mAP50-95 values, indicating increased misclassifications in this category.

In summary, the YOLOv8n model maintained strong performance in key classes but experienced slight degradation in overall Recall and localization precision during testing. These findings emphasize the importance of evaluating model robustness across larger and more diverse datasets to uncover detection weaknesses and guide future optimization.

Figure 9 and Figure 10 shows the defect detection results obtained from the proposed model compared to the original labeled data. These visualizations demonstrate the model's capability in accurately identifying and localizing various defect types on batik fabric samples. The predicted bounding boxes and confidence scores are presented alongside the ground truth annotations for direct comparison.

Figure 9. Predicted Defect by The YOLOv8n

Figure 10. Truth Defect Labeling on Batik Fabric

As shown in Figure 9, the model successfully detected multiple defect instances with varying confidence levels, demonstrating its ability to handle different defect classes and complexities. The predicted bounding boxes closely align with the ground truth labels displayed in Figure 10, indicating reliable localization accuracy. Some minor discrepancies in bounding box size and positioning are observed, which may be attributed to the inherent variability in defect shapes and patterns. Overall, these qualitative results confirm that the model effectively generalizes to real batik fabric images and is capable of precise defect detection, supporting its practical application for automated quality control.

3.5. Comparative Analysis of The Architecture

To further evaluate the relative performance and efficiency of the two object detection models, YOLOv5n and YOLOv8n, a comparative analysis was conducted. This analysis focuses on three key aspects: the F1-score at the optimal confidence threshold, total computation time during training, and average inference time per image. These metrics are crucial for understanding not only the detection accuracy but also the practical deployment efficiency of each architecture in real-world scenarios. Furthermore, Figure 11 provides a visual comparison of the detection results between YOLOv5n and YOLOv8n, highlighting the qualitative differences in identifying defects across various test images.

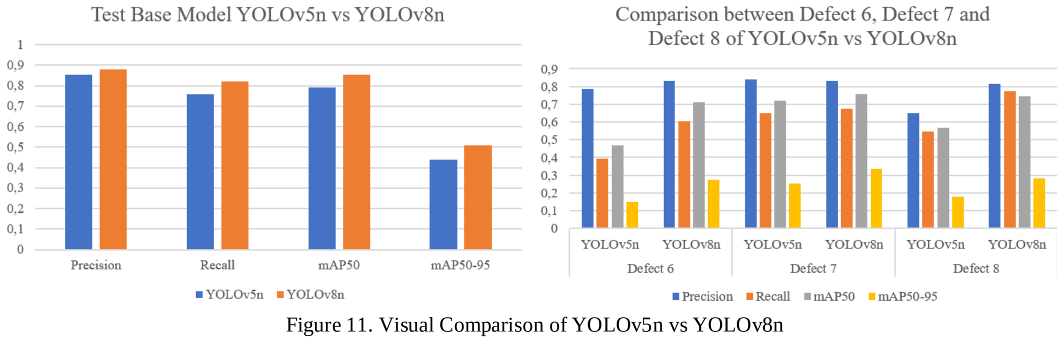

Figure 11. Visual Comparison of YOLOv5n vs YOLOv8n

Overall, YOLOv8n demonstrates better performance than YOLO5n, particularly in detecting more complex defect types. This is especially evident in the case of Defect 6, Defect 7, and Defect 8, where YOLOv5n consistenly struggles to accurately identify these classes. The challenges faced by YOLOv5n in detecting these defects suggest limitations in capturing the subtle or irregular patterns associated with them. In contrast, YOLOv8n shows a higher detection capability and robustness, making it more effective for real-world deployment where precision in identifying diverse defect types is crucial. Table 1 summarizes the results of this architectural comparison.

Table 1. The Results of This Architectural Comparison

Architecture	F1	Computation Time	Inference
YOLOv5n	0.86 (Confidence 0.46)	0.868 hours	9.7 ms/images
YOLOv8n	0.87 (Confidence 0.297)	0.320 hours	2.9 ms/images

The results clearly show that YOLOv8n outperforms YOLOv5n in terms of both computational efficiency and detection performance. YOLOv8n achieved a slightly higher F1-score of 0.87 compared to 0.86 for YOLOv5n, indicating a marginal improvement in the balance between precision and recall. More significantly, YOLOv8n required substantially less training time—only 0.320 hours versus 0.868 hours for YOLOv5n—demonstrating better optimization and convergence speed.

Additionally, the inference time per image for YOLOv8n was approximately three times faster (2.9 ms/image) than YOLOv5n (9.7 ms/image). It is important to note that the inference time reported in Table 1 reflects the total end-to-end processing time per image, which includes the cumulative duration of preprocessing, forward inference, loss calculation, and postprocessing. This considerable reduction in inference time makes YOLOv8n more suitable for real-time applications and deployment on devices with limited computational resources.

However, despite these clear advantages, it is important to acknowledge potential limitations of YOLOv8n. For instance, the model may still struggle in scenarios involving extreme occlusion, highly cluttered backgrounds, or defect types with very subtle visual cues, where the model's feature extraction capability might be insufficient for precise localization. Additionally, as a lightweight variant, YOLOv8n may exhibit reduced robustness compared to larger models when generalized across vastly different fabric textures or unseen defect types. Therefore, while YOLOv8n offers superior speed and comparable accuracy, careful consideration should be given to the complexity of the visual domain and defect characteristic when deploying the model in production environments.

In summary, YOLOv8n provides improved detection speed and efficiency while maintaining comparable or slightly better detection accuracy, making it a more favorable architecture for small object detection in time-sensitive environments.

4. CONCLUSION

This study aimed to evaluate and compare the performance of YOLOv5n and YOLOv8n architectures for small object detection tasks, particularly in terms of detection accuracy, computational efficiency, and suitability for real-time applications. Based on a comprehensive series of validation and testing procedures, the findings clearly indicate that YOLOv8n offers superior performance over YOLOv5n across multiple metrics.

YOLOv8n achieved a higher average F1-score of 0.87 at an optimal confidence threshold of 0.297, compared to 0.86 for YOLOv5n at a threshold of 0.46. More significantly, YOLOv8n demonstrated remarkable efficiency with a training time of only 0.320 hours, which is more than twice as fast as YOLOv5n (0.868 hours). Furthermore, the model's inference time per image was approximately three times faster, recorded at 2.9 ms/image compared to 9.7 ms/image for YOLOv5n. These results confirm that YOLOv8n is better optimized for both speed and accuracy.

From the parameter analysis, YOLOv8n consistently outperformed YOLOv5n in detection tasks across various defect classes, particularly in challenging or time-sensitive scenarios. While both models showed strong detection capability in certain classes, YOLOv8n offered more balanced and reliable performance across broader testing conditions. YOLOv8n is a more effective and efficient model for small object detection and is particularly well-suited for real-time detection applications, where high speed and low latency are essential. These results support the adoption of YOLOv8n in practical deployments that require fast and accurate visual analysis, such as industrial inspection, autonomous systems, and embedded devices.

In the context of the batik industry, these findings support real-time quality control improvements, potentially reducing defects and increasing productivity. Moreover, the benefits of this approach may extend to other sectors that require fine-grained defect detection, such as textile manufacturing, electronics inspection, and precision engineering. For future work, further optimizations of the YOLOv8n model can be explored, including lightweight architectural modifications or quantization for deployment on edge devices. Additionally, investigating alternative model architectures that may offer even better trade-offs between speed, accuracy, and resource efficiency would be a promising direction.

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

.

REFERENCES

[1] D. S. E. Atmaja, S. Wibirama, M. K. Herliansyah, and A. Sudiarso, "Comparative study of integral image and normalized cross-correlation methods for defect detection on Batik klowong fabric," *Results in Engineering*, vol. 25, Mar. 2025, doi: 10.1016/j.rineng.2025.104124.

[2] F. A. Putra *et al.*, "Classification of Batik Authenticity Using Convolutional Neural Network Algorithm with Transfer Learning Method," in *2021 6th International Conference on Informatics and Computing, ICIC 2021*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ICIC54025.2021.9632937.

[3] N. W. Parwati Septiani *et al.*, "Convolutional Neural Network (CNN) Algorithm for Geometrical Batik Sade' Motifs," in *ICCoSITE 2023 - International Conference on Computer Science, Information Technology and Engineering: Digital Transformation Strategy in Facing the VUCA and TUNA Era*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 597–602. doi: 10.1109/ICCoSITE57641.2023.10127829.

[4] S. Akhmad, A. Arendra, Mu'Alim, K. Winarso, and R. Hidayat, "Design of the mBatik, textile hot wax applicator to emulate hand drawn batik using CNC plotter machine and characterization of wax plotting parameters," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jul. 2020. doi: 10.1088/1742-6596/1569/3/032026.

[5] W. Wesnina, M. Prabawati, and M. Noerharyono, "Integrating traditional and contemporary in digital techniques: the analysis of Indonesian batik motifs evolution," *Cogent Arts Humanit*, vol. 12, no. 1, 2025, doi: 10.1080/23311983.2025.2474845.

[6] D. A. Mardani, Pranowo, and A. J. Santoso, "Deep learning for recognition of Javanese batik patterns," 2020, p. 030012. doi: 10.1063/5.0000686.

[7] S. P. D. Kristiana, A. M. S. Asih, and A. Sudiarso, "Designing Simulation to Improve Production Efficiency of Batik Industry," *Simul Gaming*, vol. 54, no. 6, pp. 730–759, Dec. 2023, doi: 10.1177/10468781231205667.

[8] N. A. M. Asri, A. M. A. Hamid, Norhashimahshaffiar, N. A. Sukindar, S. I. Syedshaharuddin, and F. S. Hassan, "APPLICATION OF HOUSE OF QUALITY IN THE CONCEPTUAL DESIGN OF BATIK WAX EXTRUDER AND PRINTER," *IIUM Engineering Journal*, vol. 23, no. 1, pp. 310–328, 2022, doi: 10.31436/IIUMEJ.V23I1.1842.

[9] M. Naufal Daffa'ulhaq and A. Sudiarso, "Parameter Optimization of Writing Batik Machine with Synthetic & Natural Fabric *Correspondence," 2023.

[10] K. Kudiya, B. Sumintono, S. Sabana, and A. Sachari, "Batik Artisans' Judgment of Batik Wax Quality and Its Criteria: An Application of the Many-Facets Rasch Model," in *Pacific Rim Objective Measurement Symposium (PROMS) 2016 Conference Proceedings*, Singapore: Springer Singapore, 2018, pp. 27–37. doi: 10.1007/978-981-10-8138-5_3.

[11] A. Z. Khoirunnisa and I. K. Sunarya, "Analysis of the Mawar Gentong Surakarta Batik Motif in A Charles Sanders Peirce's Semiotic Study," in *Proceedings of the 3rd International Conference on Arts and Arts Education (ICAAE 2019)*, Paris, France: Atlantis Press, 2020. doi: 10.2991/assehr.k.200703.012.

[12] A. Fitri Mustafida, A. Sudiarso, and M. Kusumawan Herliansyah, "CNC Parameter Optimization for Leather Batik Production," 2024.

[13] R. Syamwil, "Conservation of batik: Conseptual framework of design and process development," 2018, p. 020043. doi: 10.1063/1.5028101.

[14] X. Jun, J. Wang, J. Zhou, S. Meng, R. Pan, and W. Gao, "Fabric defect detection based on a deep convolutional neural network using a two-stage strategy," *Textile Research Journal*, vol. 91, no. 1–2, pp. 130–142, Jan. 2021, doi: 10.1177/0040517520935984.

[15] K. Yu, W. Lyu, X. Yu, Q. Guo, W. Xu, and L. Zhang, "FA-YOLO: A High-Precision and Efficient Method for Fabric Defect Detection in Textile Industry," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E107.A, no. 6, pp. 890–898, Jun. 2024, doi: 10.1587/transfun.2023EAP1030.

[16] W. Pebrianto, P. Mudjirahardjo, and S. H. Pramono, "YOLO Method Analysis and Comparison for Real-Time Human Face Detection," in *Proceedings - 11th Electrical Power, Electronics, Communications, Control, and Informatics Seminar, EECCIS 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 333–338. doi: 10.1109/EECCIS54468.2022.9902919.

[17] H. Zhao and T. Zhang, "Fabric Surface Defect Detection Using SE-SSDNet," *Symmetry (Basel)*, vol. 14, no. 11, Nov. 2022, doi: 10.3390/sym14112373.

[18] S. Sahani, P. Sumbre, and M. Biradar, "Fabric Defect Detection Using YOLO V8," in *2024 8th International Conference on Computing, Communication, Control and Automation, ICCUBEA 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ICCUBEA61740.2024.10775212.

[19] X. Yue, Q. Wang, L. He, Y. Li, and D. Tang, "Research on Tiny Target Detection Technology of Fabric Defects Based on Improved YOLO," *Applied Sciences (Switzerland)*, vol. 12, no. 13, Jul. 2022, doi: 10.3390/app12136823.

[20] W. Song, D. Lang, J. Zhang, M. Zheng, and X. Li, "Textile Defect Detection Algorithm Based on the Improved YOLOv8," *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3528771.

[21] Z. Yaohui, R. Jia, and L. Yu, "Yolov7-Tinier: Towards High-Precision and Lightweight Detection of Fabric Defects in Textile Plant," *Fibers and Polymers*, Sep. 2024, doi: 10.1007/s12221-024-00662-w.

[22] Y. Chen *et al.*, "Research on Fabric Defect Detection Algorithm Based on YOLO-V8," in *2025 Asia-Europe Conference on Cybersecurity, Internet of Things and Soft Computing (CITSC)*, IEEE, Jan. 2025, pp. 1018–1022. doi: 10.1109/CITSC64390.2025.00189.

[23] V. A. Adibhatla, H. C. Chih, C. C. Hsu, J. Cheng, M. F. Abbod, and J. S. Shieh, "Applying deep learning to defect detection in printed circuit boards via a newest model of you-only-look-once," *Mathematical Biosciences and Engineering*, vol. 18, no. 4, pp. 4411–4428, 2021, doi: 10.3934/mbe.2021223.

[24] Z. Qi, L. Ding, X. Li, J. Hu, B. Lyu, and A. Xiang, "Detecting and Classifying Defective Products in Images Using YOLO."

[25] M. Hussain, "YOLOv5, YOLOv8 and YOLOv10: The Go-To Detectors for Real-time Vision," Jul. 2024, [Online]. Available: http://arxiv.org/abs/2407.02988

[26] B. Mahaur and K. K. Mishra, "Small-object detection based on YOLOv5 in autonomous driving systems," *Pattern Recognit Lett*, vol. 168, pp. 115–122, Apr. 2023, doi: 10.1016/j.patrec.2023.03.009.

[27] S. Rahman, J. H. Rony, J. Uddin, and M. A. Samad, "Real-Time Obstacle Detection with YOLOv8 in a WSN Using UAV Aerial Photography," *J Imaging*, vol. 9, no. 10, Oct. 2023, doi: 10.3390/jimaging9100216.

[28] M. Mao and M. Hong, "YOLO Object Detection for Real-Time Fabric Defect Inspection in the Textile Industry: A Review of YOLOv1 to YOLOv11," *Sensors*, vol. 25, no. 7, p. 2270, Apr. 2025, doi: 10.3390/s25072270.

BIOGRAPHIES OF AUTHORS

Rifqi Restu Hamidi is currently pursuing his master studies in Industrial Engineering at Universitas Gadjah Mada (UGM). His bachelor's degree in Industrial Engineering from Universitas Islam Indonesia (UII). During his postgraduate studies, he served as an assistant lecture of Programming and Computing at UGM. His academic interests focus on data science, artificial intelligence in manufacturing and quality control systems, image processing, industrial engineering's role in optimizing the processes through innovation, and community service programs related to industrial engineering applications. He can be contacted at email: rifqirestuhamidi@mail.ugm.ac.id

Muhammad Kusumawan Herliansyah received Bachelor's degree (S.T.) in Mechanical Engineering from Universitas Gadjah Mada, Yogyakarta, Indonesia in 1996 and Master degree (M.T.) in manufacturing system from Institut Teknologi Bandung, Indonesia, in 2002. He receives doctoral double degree in advance material processing from University of Malaya, Malaysia, and Graduate School of Engineering Kyoto Unversity. He is currently a lecturer and researcher at UGM in the Department of Mechanical and Industrial Engineering. His research interests are advanced material engineering, biomaterials, and automation. He received certificate as a professional engineer from the Institution of Engineers Indonesia (PII) and ASEAN Engineer from ASEAN Fderation of Engineering Organisations (AFEO). He can be contacted at email: herliansyah@ugm.ac.id

Denny Sukma Eka Atmaja received the Bachelor degree (S.T.) in Industrial Engineering from Institut Teknologi Telkom, Indonesia, and the Master of Science degree (M.Sc.) in Industrial Engineering fro Universitas Gadjah Mada, Indonesia. His master thesis research was the optimization of flaw detection in ceramic tiles. He is currently a doctoral student at the Department of Mechanical and Industrial Engineering Universitas Gadjah Mada with a research focus in the field of artificial intelligence, especially developing programming for quality inspection of the *Batik klowong* process in real time. From 2015 until now, he is a lecturer in Telkom University, Bandung, West Java, Indonesia in School of Industrial and System Engineering. His research interest are in industrial automation, optimization, image processing, and artificial intelligence. He can be contacted at email: denny.sukma.e@mail.ugm.ac.id and dennysukma@telkomuniversity.ac.id

Andi Sudiarso received Bachelor's degree (S.T.) and Master degree (M.T.) in Electrical Engineering from Universitas Gadjah Mada (UGM), Indonesia, master degree (M.Sc) in Manufacturing Engineering from UMIST, UK, and Ph.D. in Mechanical Engineering from The University of Manchester, UK. He is currently a lecturer and researcher at the Department of Mechanical and Industrial Engineering, UGM. His research interest are manufacturing engineering, production system, and artificial intelligence. He's the owner of Butimo *Batik*, which has more than 25 IPRs, and the founder of the Yayasan Rumah Riset Indonesia (YRRI) foundation. He can be contacted at email: a.sudiarso@ugm.ac.id

.