Volume XI Issue 3 Year 2026 | Page 829-840 | ISSN: 2527-9866
Received: 13-06-2026 | *Revised: 21-07-2026* | *Accepted: 03-08-2026*

Implementation of YOLO26 for Mold Detection on White Bread Based on Digital Imagery

Malvin Hendrawan[1], Yoannita[2]
[1, 2]Multi Data Palembang University, Palembang, Indonesia, 30113
E-mail: malvinhendrawan_2226250038@mhs.mdp.ac.id[1],yoannita@mdp.ac.id[2]
*Correspondence: malvinhendrawan_2226250038@mhs.mdp.ac.id

Abstract *White bread is highly susceptible to visible mold contamination, which causes physical deterioration and potential health risks. Conventional manual visual inspection is slow, subjective, and inconsistent, necessitating an automated detection system. This study implemented the YOLO26n algorithm for mold contamination detection on white bread based on digital imagery. A primary dataset of 300 images (150 fresh bread and 150 moldy bread) was collected independently, annotated via Roboflow, and split into 70% training, 20% validation, and 10% testing. The model was trained on Google Colab using the MuSGD optimizer with 200 epochs. The YOLO26n model achieved an overall precision of 0.827, recall of 0.734, and mAP50 of 0.711, with an inference speed of 8.1 ms per image, demonstrating its potential as a fast and lightweight solution for automated mold inspection, though further improvement in moldy bread detection performance is required before reliable deployment in bakery production lines.*

Keywords: *Mold Detection, White Bread, YOLO26.*

1. Introduction

Bread consumption in Indonesia has shown a positive growth trend, increasing by approximately 14–15% annually, driven by changing lifestyles and increasing demand for practical food products [1] and it is commonly consumed as a morning staple or a substitute for rice [2]. Despite its practicality, white bread has a limited shelf life of approximately 3–4 days and is highly susceptible to visible mold contamination, leading to physical deterioration, including discoloration, unpleasant odor, and a hardened texture that renders the product unfit for consumption[3], [4]. Such contamination, if caused by toxigenic fungi such as Aspergillus sp., may also pose potential health risks including aflatoxin-related hazards, although microbiological identification was beyond the scope of this study [5]. On a broader scale, fungal contamination poses a serious threat to food supply chain safety and human health, as exposure to fungi not only leads to substantial economic losses but also risks triggering dangerous illnesses including allergies, asthma, and respiratory disorders such as *Aspergillus* [6].

To date, conventional methods such as manual visual inspection remain widely used; however, this approach is often slow, labor-intensive, and insensitive to early-stage contamination. As a solution, *computer vision* technology based on *deep learning*, particularly the YOLO (*You Only Look Once*) algorithm family, offers non-destructive real-time detection capabilities. The use of recent variants such as YOLOv8n, YOLOv10n, YOLO11n, and YOLOv12n has become highly relevant, as these "nano" architectures are specifically designed to operate on resource-constrained *edge devices* in production environments [7]. However, prior YOLO variants still exhibit limitations in detecting very small objects such as fungal spores due to constraints in pixel representation, presenting a gap that necessitates the exploration of a more advanced algorithm.

YOLO26 represents the latest evolution in the YOLO series, developed with a focus on architectural simplification and efficiency for *edge devices*. Key technical innovations include an end-to-end inference system without Non-Maximum Suppression (NMS) to reduce latency, the MuSGD optimizer, a hybrid of SGD and Muon inspired by LLM training advances—for faster convergence, and the use of ProgLoss and STAL loss functions that improve accuracy on small objects. These optimizations enable YOLO26 to run up to 43% faster on CPU compared to its predecessors [8], making it highly suitable for detecting tiny objects such as fungal spores and for integration into low-power *edge devices* on bread production lines [9]. To the best of the authors' knowledge, no prior study has specifically applied YOLO26 for fungal contamination detection on bread, marking this work as a novel contribution at the intersection of *computer vision* and food quality control.

This study aims to implement the YOLO26 algorithm as an automated fungal contamination detection system on white bread to replace manual inspection methods that are slow and subjective. The model is trained and evaluated on a self-collected dataset comprising two categories *Fresh Bread* and *Moldy Bread,* split into 70% training, 20% validation, and 10% testing, with performance measured using precision, recall, *Intersection over Union* (IoU), and *mean Average Precision* (mAP50) [9][10]. The findings are expected to serve as a practical reference for the application of *deep learning* in food quality control, particularly for the bakery industry.

2. Literature Review

Several prior studies have explored the use of deep learning and computer vision for mold and fungal detection on food products. Jubayer [4] applied YOLOv5 to detect mold on food surfaces using a dataset of 2,050 augmented images, achieving a precision of 98.10%, a recall of 100%, and an AP of 99.60%, demonstrating the strong capability of one-stage detectors for non-destructive contamination assessment. Building on this foundation, Harahap and Syahputra [11] developed a real-time mold detection system for white bread using MobileNetV2 Transfer Learning on 640 primary images, correctly classifying 30 out of 32 moldy bread samples and all 32 clean bread samples. However, their approach relied on image-level classification rather than object detection, meaning the precise location of contamination on the bread surface could not be identified—a critical limitation for automated quality control applications.

More recent work has focused on comparing YOLO nano variants specifically for mold detection on bread. Siwi [5] conducted a comparative analysis of YOLOv8n, YOLOv10n, YOLO11n, and YOLOv12n for detecting *Rhizopus stolonifer* mold, finding that YOLO11n achieved the highest mAP50 of 0.504 while YOLOv12n obtained the best mAP50:95 of 0.224. Although these nano architectures are well-suited for edge deployment, the relatively low mAP50 values across all variants indicate that detecting small mold colonies with high spatial precision remains a persistent challenge. Munthe [3] similarly demonstrated the limitation of non-localization approaches, where a KNN classifier with ORB feature extraction achieved 99% overall accuracy on 3,000 images but produced no bounding box output, making it unsuitable for pinpointing contamination zones on a production line.

In the broader context of YOLO-based object detection for food quality and related domains, multiple studies confirm the reliability of this algorithm family across varying conditions. Dewi et al [12] achieved a mAP50 of 96.51% using YOLOv8m for Bali MSME staple food quality detection, while Maulana et al. [13] reported a mAP of 91.3% for fish freshness identification with YOLOv8. Despite these strong results, the aforementioned studies all rely on YOLOv8 or earlier variants that include Non-Maximum Suppression (NMS) post-processing, which introduces additional inference latency. To the best of the authors' knowledge, no prior study has applied YOLO26, the latest Ultralytics model featuring NMS-free inference, MuSGD optimizer, and ProgLoss/STAL loss functions designed to improve small-object accuracy for mold detection on white bread. This research gap motivates the present study.

3.Methods

This study was conducted through several systematically arranged stages to ensure that the data collection and analysis processes were carried out accurately. The research workflow consists of five sequential stages, beginning with a literature study and concluding with model testing. The overall research stages are illustrated in Figure 1.

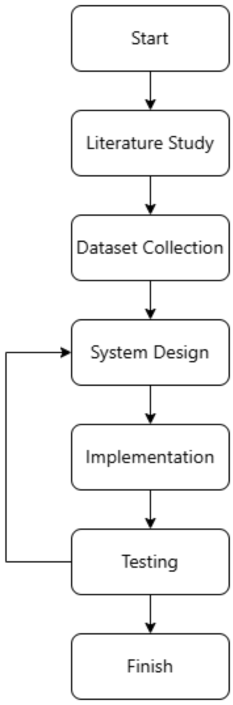

Figure 1. Research Flow

A. Dataset Collection

The dataset used in this research is primary data collected independently through direct image acquisition of white bread samples obtained from various local bakeries and minimarkets to ensure representative texture variation. The bread samples include variants with and without crust, as well as wheat, chocolate, and pandan bread types. Image acquisition was performed one by one using a smartphone camera assisted by a portable *light tent* (photobox) equipped with an integrated LED lighting system. The controlled illumination provided by the photobox was intended to standardize image acquisition conditions and reduce variability caused by external lighting, thereby improving image consistency for subsequent model training[14]. Varied backgrounds (white, orange, cream, and black) and additional props such as plates and cutting boards were used to minimize *ambient light* interference and produce consistent object contrast.

To stimulate natural fungal growth, the surface of each bread sample was rubbed against the palm, then sprayed with water until moist, sealed in a 25 × 16 cm zip-lock bag, and stored at room temperature for 4–7 days until visible fungal growth appeared. Images were then re-acquired from each sample that had shown fungal growth. The final dataset consists of 150 images of fresh bread and 150 images of moldy bread, totaling 300 images, which were subsequently compiled and subjected to augmentation to increase data variation. Sample images of both categories are presented in Figure 2.

Figure 2. Fresh & Moldy Bread

B. System Design

This stage involved designing the YOLO26-based fungal detection system for white bread. All bread images were uploaded to the Roboflow platform for annotation. For the *moldy bread* category, bounding boxes were drawn over each fungal contamination area and labeled as the *moldy_bread* class, while for the *fresh bread* category, a bounding box was drawn over the entire bread surface and labeled as the *fresh_bread* class.

Following annotation, the dataset split was performed at the image level within the Roboflow platform using a 70/20/10 ratio. Following the split and augmentation process, the training subset expanded to 951 images, the validation subset contained 91 images, and the testing subset contained 45 images. The testing subset was not subjected to any augmentation, and the increase from the expected 30 images (10% of 300) to 45 images is attributable to Roboflow's internal splitting mechanism, which distributes images based on bounding box instance distribution rather than strict image count proportions to ensure balanced class representation across subsets. Preprocessing was then applied to all three subsets using the Auto-Orient feature to standardize image orientation, followed by resizing all images to 640×640 pixels. Data augmentation was applied exclusively to the training subset to improve model generalization and reduce the risk of overfitting, as data augmentation has been widely recognized as an effective strategy for increasing training data diversity in deep learning models [15]. The augmentation techniques consisted of horizontal flip, 90° clockwise rotation, grayscale conversion (10%), brightness adjustment (−15% and +15%), blur (0.2 px), and hue shift (−15° and +15°). These transformations were designed to simulate variations in image orientation, illumination, color characteristics, and acquisition conditions that may occur in practical scenarios, thereby enhancing the model's robustness and generalization capability [16]. The validation and testing subsets were left unaugmented to ensure objective evaluation. The dataset then proceeded to the training and validation stages to produce the final model, which was subsequently evaluated using the testing subset to obtain the final detection performance results. The overall system design scheme is illustrated in Figure 3.

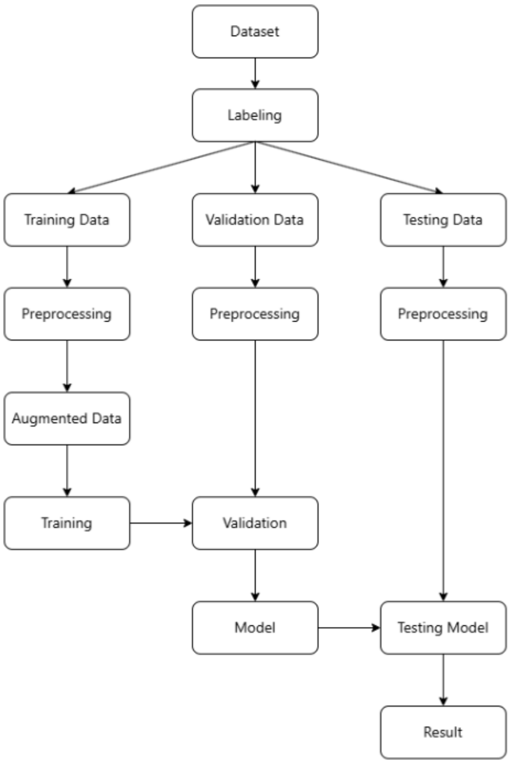

Figure 3. Model Design YOLO26

C. Implementation

Model implementation was performed using the Python programming language with the Ultralytics library on the Google Colab platform to leverage GPU acceleration. The YOLO26 nano variant (YOLO26n) was selected to optimize inference speed without significantly sacrificing detection accuracy. The training parameters were configured as follows: 200 epochs to ensure model convergence, a batch size of 8, an image size of 640×640 pixels, and the MuSGD optimizer—a hybrid of Muon and SGD—to accelerate the training process and minimize the loss function. The learning rate was dynamically managed using a cosine scheduler with an initial value of 0.001, and early stopping was applied with a patience of 50 consecutive epochs to prevent overfitting.

D. Testing

At this stage, the trained and validated model was evaluated using the testing dataset to measure its detection and classification accuracy. The results are presented in the form of images annotated with bounding boxes, class labels, and *confidence scores* as indicators of the model's certainty toward detected objects. System performance was further assessed using a *Confusion Matrix* to compare the correspondence between model predictions and actual labels. In this context, TP represents correctly detected mold instances, FP represents false detections, and FN represents missed mold instances.

The determination of TP, FP, and FN was based on a one-to-one matching mechanism between predicted and ground truth bounding boxes, using an IoU threshold ≥ 0.5 and a confidence score ≥ 0.25. A predicted bounding box is classified as TP if it meets both thresholds against an unmatched ground truth; as FP if the confidence score ≥ 0.25 but IoU < 0.5, has no matching ground truth, or maps to an already-matched ground truth; and as FN if a ground truth bounding box has no matching prediction with IoU ≥ 0.5 [17]. In object detection tasks, Mean Average Precision (mAP) is widely recognized as the standard evaluation metric because it simultaneously measures localization accuracy and classification performance by considering the overlap between predicted and ground truth bounding boxes across different object classes

[18]. Model performance was evaluated using five metrics defined in equations (1) through (5) as follows:

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (2)$$

$$IoU = \frac{A \cap B}{A \cup B} \quad (3)$$

$$mAP50 = \frac{1}{N}\sum_{i=1}^{N} AP_i^{0.5} \quad (4)$$

$$mAP50-95 = \frac{1}{10}\sum_{t=\{0.5,0.55,..0,95\}} mAPt \quad (5)$$

where:
TP : the number of correctly detected defect instances
TN : the number of correctly identified non-defect instances
FP : the number of non-defect instances incorrectly classified as defect
FN : the number of defect instances that were missed
N : the number of classes in the dataset
A : predicted bounding box
B : ground truth bounding box
i : class category in the dataset
AP*i* : *Average Precision* for class *i*

3. Results and Discussion

A. Detection Performance

The trained YOLO26n model was evaluated on the testing dataset consisting of 45 images with 78 total instances across two classes. The term *instances* refers to the total number of individual annotated objects present in the testing images. Each bounding box annotation counts as one instance, regardless of how many objects appear per image. The overall detection performance is presented in Table I.

Table 1. YOLO26n Detection Performance on Testing Data

Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
All	45	78	0.827	0.734	0.711	0.482
Fresh Bread	20	20	0.905	0.950	0.943	0.811
Moldy Bread	25	58	0.750	0.517	0.480	0.152

Table I presents the per-class detection performance of the YOLO26n model. The *fresh bread* class achieved excellent results with precision of 0.905, recall of 0.950, and mAP50 of 0.943, indicating that the model is highly capable of identifying uncontaminated bread. The *moldy bread* class achieved precision of 0.750 and mAP50 of 0.480, with a recall of 0.517, meaning that approximately half of the fungal contamination instances present in the testing images were not detected by the model. The relatively high mAP50 for fresh bread (0.943) compared to moldy bread (0.480) reflects the challenge of detecting mold colonies, which vary significantly in color, size, and pattern across samples. The inference speed of 8.1 ms per image demonstrates the real-time processing potential of YOLO26n. However, given the relatively low recall for the moldy bread class (0.517), further improvement in detection accuracy is required before the system can be reliably deployed in a production environment where missed detections carry significant food safety implications.

The relatively low recall for the moldy bread class (0.517) indicates that approximately half of the fungal contamination instances present in the testing images were not detected by the model. This limitation is primarily attributed to two factors. First, the dataset used in this study is relatively small, consisting of only 150 moldy bread images, which limits the model's ability to learn the full range of real-world mold variations, including different colony sizes, colors, growth stages, and lighting conditions. Although data augmentation was applied to increase training data diversity, augmentation cannot fully replace the genuine variation that would be obtained from a larger original dataset. Second, the high visual variability of visible mold contamination—ranging from early-stage pale yellow colonies to more advanced green and black patches—poses an inherent detection challenge, particularly for small or low-contrast mold regions that are visually similar to the natural bread surface texture.

B. Confusion Matrix Analysis

The confusion matrix of the YOLO26n model on the testing dataset is presented in Figure 4.

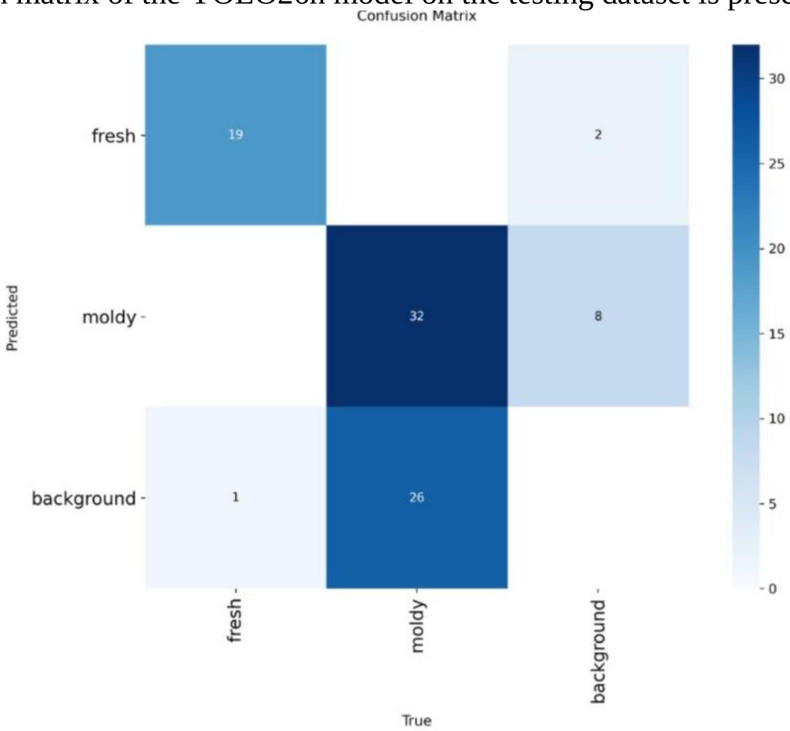

Figure 4. Confusion Matrix of YOLO26n Model

Figure 4 illustrates the classification behavior of the YOLO26n model across both classes and the background category. The model correctly detected 19 out of 20 *fresh bread* instances (TP = 19) and 32 out of 58 *moldy bread* instances (TP = 32). A total of 26 *moldy bread* instances were misclassified as background, representing the primary source of false negatives in the model. Additionally, 1 *fresh bread* instance was misclassified as background, and 8 background instances were incorrectly predicted as *moldy bread*. These results indicate that the model has a tendency to miss moldy regions that are small or exhibit low contrast against the bread surface, which is consistent with the relatively low recall value (0.517) observed for the *moldy* class.

These findings further confirm that the current model, while demonstrating strong performance for the fresh bread class, requires additional training data and potentially more targeted augmentation strategies to improve its reliability in detecting the full spectrum of mold contamination patterns on white bread.

C. Evaluation Curve Analysis

To further characterize the model's behavior across varying confidence thresholds, four evaluation curves were analyzed: the Precision-Confidence Curve (Figure 6), Recall-Confidence Curve (Figure 7), F1-Confidence Curve (Figure 8), and Precision-Recall Curve (Figure 9).

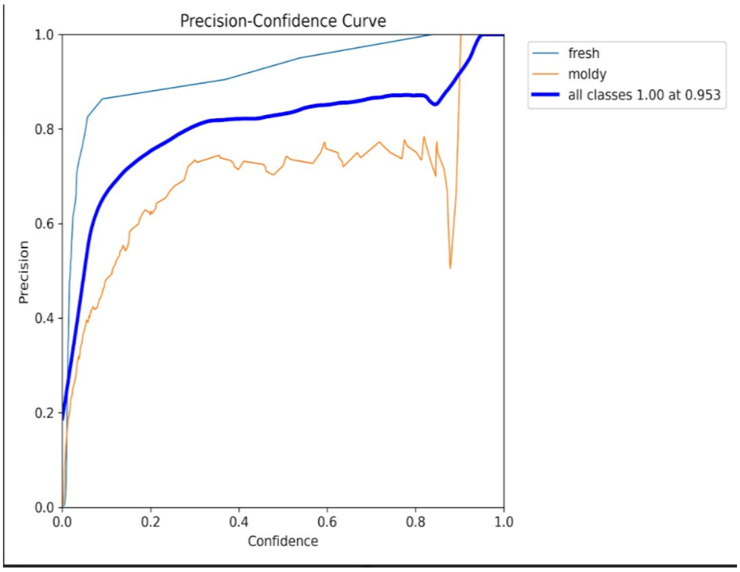

Figure 5. Precision-Confidence Curve YOLO26n Model

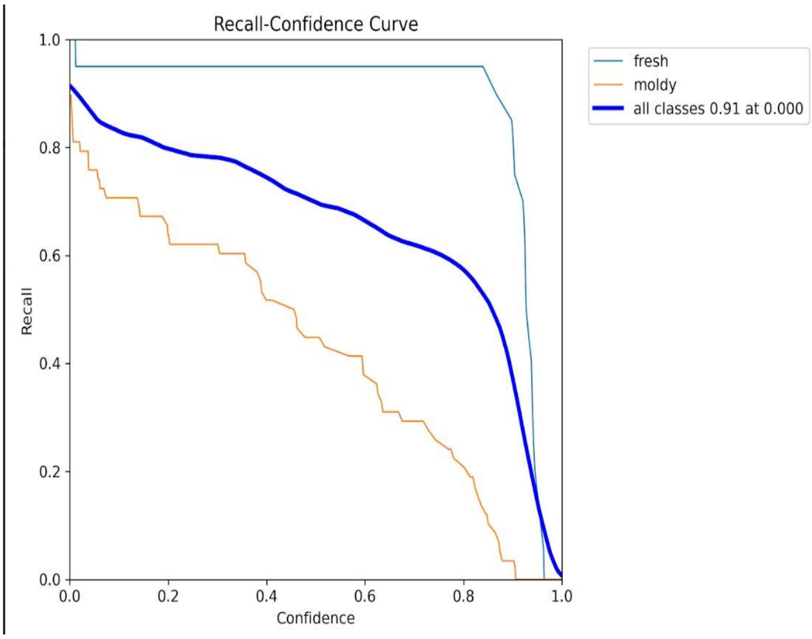

Figure 6. Recall-Confidence Curve of YOLO26n Model

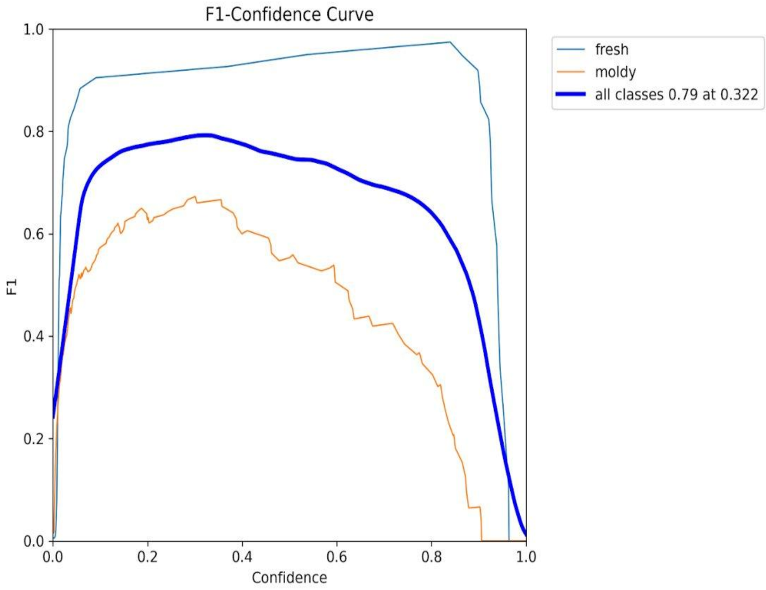

Figure 7. F1-Confidence Curve of YOLO26n Model

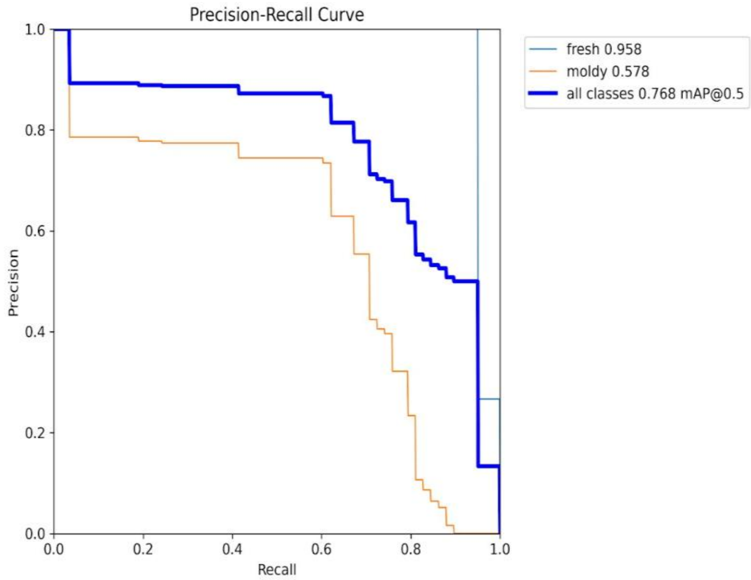

Figure 8. Precision-Recall Curve of YOLO26n Model

The Precision-Confidence Curve (Figure 6) shows that the fresh bread class maintains high and stable precision above 0.85 across nearly the entire confidence range, while the moldy bread class exhibits lower and less stable precision, oscillating between 0.70 and 0.78 across confidence values of 0.1–0.8. This indicates that while the model's mold predictions are largely correct when made, its confidence in those predictions is comparatively lower. The Recall-Confidence Curve (Figure 7) reveals that the fresh bread class sustains recall near 0.95 up to a confidence threshold of approximately 0.9, whereas the moldy bread class shows a steeper decline as the threshold increases. The maximum recall across all classes reaches 0.91 at confidence = 0.000, reflecting the model's theoretical upper bound of detection coverage at the lowest threshold. The F1-Confidence Curve (Figure 8) shows that the optimal F1 score across all classes is 0.79, achieved at a confidence threshold of 0.322. The fresh bread class achieves a near-peak F1 of approximately 0.97 in the confidence range of 0.8–0.9, whereas the moldy bread class peaks at only 0.67 near confidence = 0.3, reflecting the disparity in detection difficulty between the two classes. Finally, the Precision-Recall Curve (Figure 9) confirms the per-class performance gap: the fresh bread class achieves an AP of 0.958 with a consistently high curve, while the moldy bread class achieves an AP of only 0.578. The overall mAP@0.5 of 0.768 reported on the PR curve differs slightly from the 0.711 in Table I due to differences in interpolation methodology and confidence filtering. Taken together, these curves confirm that while YOLO26n performs reliably for fresh bread detection, the moldy bread class requires further improvement through dataset expansion and more targeted augmentation strategies.

C. Misclassification Analysis

To better understand the source of detection failures, representative misclassified images were examined visually. Figure 5 shows examples of *moldy bread* instances that were not detected by the model (false negatives).

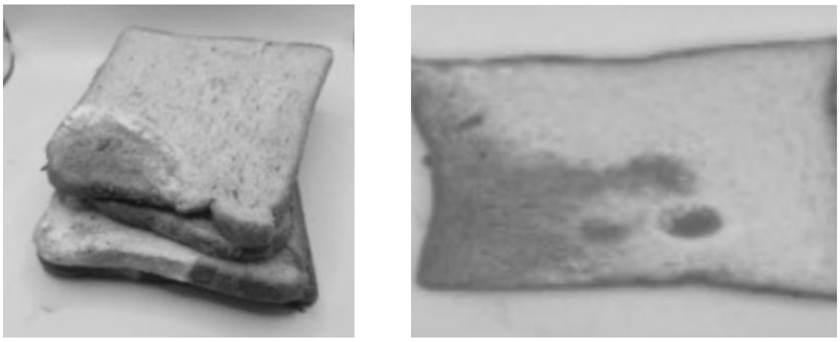

Figure 9. Examples of Misclassified Moldy Bread Instances

Visual inspection of the misclassified samples reveals three recurring patterns. First, several missed instances involved mold colonies that were very small relative to the total bread surface area, making them difficult to distinguish from natural surface texture variations. Second, images captured under high ambient brightness resulted in overexposed regions where mold discoloration became indistinct from the surrounding bread surface. Third, mold colonies exhibiting colors close to the natural bread crust color, such as light brown or pale yellow early-stage growth, showed the lowest detection rates, as the color contrast between contaminated and uncontaminated regions was insufficient for the model to form confident bounding box predictions. These findings suggest that the low mAP50 of 0.480 for the *moldy bread* class is primarily driven by cases where mold colonies are small, low-contrast, or overlap with visually similar non-mold regions, rather than a systematic failure of the model architecture.

4. Conclusions

This study implemented the YOLO26n algorithm for automated fungal contamination detection on white bread based on digital imagery. The model achieved an overall precision of 0.827, recall of 0.734, and mAP50 of 0.711 on the testing dataset. For the *fresh bread* class, the model demonstrated strong performance with a mAP50 of 0.943 and recall of 0.950, indicating reliable identification of uncontaminated bread. For the *moldy bread* class, however, the mAP50 remained relatively low at 0.480, reflecting the inherent difficulty of detecting mold colonies that vary widely in size, color, and surface contrast, particularly small, low-contrast, or early-stage colonies that are visually similar to the surrounding bread surface.

These results demonstrate that YOLO26n shows potential as a foundation for an automated mold inspection tool on white bread, offering a faster and more objective alternative to manual visual inspection. However, the current recall for the moldy bread class (0.517) indicates that the system is not yet ready for reliable deployment in a real production environment, as missed detections in food safety applications carry significant risks. Further improvement in detection performance, particularly for the moldy bread class, is necessary before practical deployment can be considered.

It is acknowledged that the relatively small dataset size, particularly for the moldy bread class (150 images), represents the primary limitation of this study. A larger and more diverse dataset capturing early-stage mold growth, small colonies, and varied lighting conditions would be expected to substantially improve recall performance for the moldy bread class.

For future work, several directions are recommended to address the identified limitations. First, expanding the dataset, particularly for the moldy bread class with more samples that capture early-stage, small-colony, and low-contrast mold growth would directly improve recall. Second, standardizing image acquisition conditions, including controlled lighting to avoid overexposed or underexposed regions, would reduce the number of misclassified instances caused by brightness variation. Third, applying targeted augmentation strategies that simulate difficult lighting conditions during training may further improve model robustness. Furthermore, future research may explore specialized small-object detection techniques or enhanced multi-scale feature extraction architectures to improve the detection of early-stage fungal colonies, which typically occupy only a small region of the image and exhibit low visual contrast against the bread surface [19]**.** Finally, a systematic analysis of misclassified images and their underlying visual characteristics should be conducted to inform future improvements to both the dataset composition and the model training configuration.

References

[1] A. Makmur, "Alternative Demand Forecasting Methods for FMCG in Indonesia," 2021.

[2] R. Saleh, H. Ufiyana, S. Thahir, A. Dwiyana, and I. Artikel, "IDENTIFIKASI JAMUR Aspergillus sp PADA ROTI TAWAR YANG DIPERJUALBELIKAN DI DESA BUNGA EJAYA KECAMATAN PALLANGGA KABUPATEN GOWA," vol. 13, no. 1, pp. 81–90, 2025.

[3] I. G. A. Putu Mahendra, Muhammad Ikhsan Wibowo, and Zuliar Efendi, "Peningkatan Akurasi Klasifikasi Ikan kepe-kepe (Famili Chaetodontidae) dengan EfficientNetV2 dan Bayesian Hyperparameter Tuning," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 7, no. 2, pp. 260–270, Dec. 2025, doi: 10.30865/json.v7i2.9028.

[4] I Gusti Agung Putu Mahendra, A. Julianto, and A. Tedyyana, "Klasifikasi Citra Topeng Bali Berdasarkan Karakteristik Visual Menggunakan Arsitektur EfficientNetV2," *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, vol. 8, no. 3, pp. 122–132, Dec. 2025, doi: 10.31598/jurnalresistor.v8i3.1926.

[5] S. br Munthe, "IMPLEMENTASI ALGORITMA K-NEAREST NEIGHBOR TAWAR MENGGUNAKAN EKSTRAKSI FITUR ORB," 2023.

[6] F. Jubayer *et al.*, "Current Research in Food Science Detection of mold on the food surface using YOLOv5," *Current Research in Food Science*, vol. 4, pp. 724–728, 2021, doi: 10.1016/j.crfs.2021.10.003.

[7] V. H. Siwi, J. Wuntu, N. L. Wuntu, and A. D. Wuntu, "Jamur pada Produk Pangan : Analisis Komparatif Varian YOLO untuk Deteksi Rhizopus stolonifera pada Roti Mold on Food Product : Comparative Analysis of YOLO Variants for Detecting Rhizopus stolonifer on Bread," *Jurnal Ilmiah Sains*, vol. 25, no. October, pp. 173–186, 2025, doi: https://doi.org/10.35799/jis.v25i2.64398%0A.

[8] Ultralytics, "Ultralytics YOLO26." [Online]. Available: https://docs.ultralytics.com/models/yolo26/

[9] R. Sapkota, R. Harsha, and C. Ajay, "YOLO26: Key Architectural Enhancements and Performance Benchmarking for Real-Time Object Detection," no. September 2025, pp. 1–15, 2025, doi: https://doi.org/10.48550/arXiv.2509.25164.

[10] M. Larasati, "IDENTIFIKASI JAMUR Aspergillus sp. PADA ROTI TAWAR YANG DIGUNAKAN OLEH PEDAGANG ROTI BAKAR DI ALUN-ALUN KARANGANY.," *Sekolah Tinggi Ilmu Kesehatan Nasionalsurakarta*, 2021.

[11] M. L. Harahap and H. Syahputra, "DETEKSI OBJEK JAMUR PADA ROTI TAWAR SECARA REAL-TIME," *JATI(Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 2, pp. 2109–2114, 2025, doi: https://doi.org/10.36040/jati.v9i2.12999.

[12] N. Putu, D. Ariani, S. Dewi, J. Aji, D. Aryasa, and I. G. Hendrayana, "YOLOv8-Based Quality Detection of Bali MSMEs Staple Food," *TIERS Information Technology Journal*, vol. 6, no. 2, 2026, doi: https://doi.org/10.38043/tiers.v6i2.7144.

[13] A. A. Rasjid, B. Rahmat, and A. N. Sihananto, "Implementasi YOLOv8 Pada Robot Deteksi Objek," *Journal of Technology and System Information*, no. 3, pp. 1–9, 2024, doi: https://doi.org/10.47134/jtsi.v1i3.2969.

[14] R. Kaur, G. Karmakar, F. Xia, and M. Imran, *Deep learning : survey of environmental and camera impacts on internet of things images*, vol. 56, no. 9. Springer Netherlands, 2023. doi: 10.1007/s10462-023-10405-7.

[15] Z. Yang, R. O. Sinnott, J. Bailey, and Q. Ke, *A survey of automated data augmentation algorithms for deep learning-based image classification tasks*, vol. 65, no. 7. Springer London, 2023. doi: 10.1007/s10115-023-01853-2.

[16] W. Zeng, "Image data augmentation techniques based on deep learning : A survey," *Mathematical Biosciences and Engineering*, vol. 21, no. February, pp. 6190–6224, 2024, doi: 10.3934/mbe.2024272.

[17] A. Badithela, T. Wongpiromsarn, R. M. Murray, and R. O. Oct, "Evaluation Metrics for Object Detection for Autonomous Systems," 2022, doi: https://doi.org/10.48550/arXiv.2210.10298.

[18] R. Padilla, S. L. Netto, and E. A. B. Silva, "A Survey on Performance Metrics for Object-Detection Algorithms".

[19] Q. Feng, X. Xu, and Z. Wang, "Deep learning-based small object detection : A survey," *Mathematical Biosciences and Engineering*, vol. 20, no. December 2022, pp. 6551–6590, 2023, doi: 10.3934/mbe.2023282.