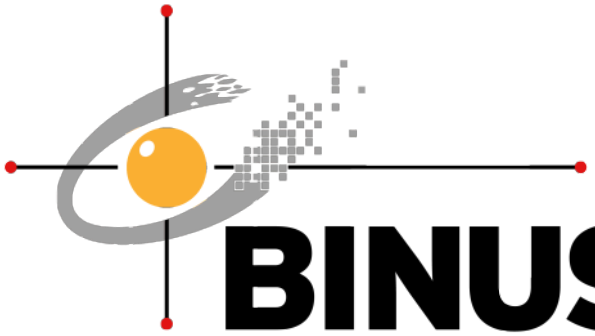


BINUS UNIVERSITY

SCHOOL OF COMPUTER SCIENCE

Deep Learning Final Report

Indonesian Food Object Detection Using YOLO11

Class:
LB01

Written by:

2702347250 - Nicholas Victorio
2702282490 - Kenneth Owen
2702253023 - Nixon Raine Vicsant

Lecturer :
Wawan Cenggoro, S.Kom, M.TI

Semester Ganjil
2025/2026

Abstract
This study successfully applied the YOLO11 object detection model to solve the complex problem of multi-class Indonesian food detection across 42 class of Indonesian Food. The model achieved a strong test-set performance with a mAP@0.5 of 0.795, despite significant data challenges identified in the exploratory analysis, such as severe class imbalance and dense object arrangements. Our findings confirm the viability of the YOLO framework for food recognition and establish a clear path for future work, focusing on improving dataset quality and integrating calorie and nutrient estimation capabilities.

I. Introduction

Indonesian cuisine is widely known for its remarkable diversity, influenced by more than 17,000 islands and hundreds of ethnic groups, resulting in thousands of unique traditional dishes[1]. This richness presents both challenges and opportunities for automated food recognition systems. On one hand, there are substantial visual variations within the same dish due to differences in ingredients, cooking styles, and presentation. On the other hand, many distinct dishes share highly similar visual characteristics, such as the subtle differences between Soto Ayam and Soto Banjar. These factors make Indonesian food recognition a challenging problem that requires advanced visual understanding.

Conventional image recognition methods often struggle to cope with the high intra-class variation and inter-class similarity commonly found in food images. In addition, real-world dining scenes frequently contain multiple dishes presented together, requiring not only accurate classification but also precise localization of each food item. Challenges such as occlusion, varying lighting conditions, inconsistent plating styles, and the complex textures of prepared foods further increase the difficulty of this task.

To address these challenges, this study adopts a deep learning–based object detection approach using YOLO11 (You Only Look Once version 11). YOLO11 is a modern single-stage Convolutional Neural Network (CNN) designed to achieve high accuracy while maintaining real-time performance. By training the model on a carefully curated dataset of Indonesian dishes, the proposed system is able to learn discriminative visual features directly from raw pixel data. This approach enables efficient and reliable detection of Indonesian food items.

II. Literature Review

Indonesian cuisine is widely recognized for its exceptional diversity, shaped by cultural, geographical, and regional influences across thousands of islands[1]. This diversity leads to significant intra-class variation, where the same dish can appear visually different due to variations in ingredients, cooking methods, and presentation styles[2]. At the same time, many Indonesian dishes share highly similar visual characteristics, resulting in high inter-class similarity[3]. Previous studies in food recognition emphasize that these characteristics make automated food detection particularly challenging, as models must distinguish subtle visual differences while remaining robust to changes in lighting, occlusion, and complex backgrounds[4].

To address such challenges, modern object detection algorithms have increasingly relied on deep learning approaches, particularly the YOLO (You Only Look Once) family of models[5]. YOLO is a single-stage object detection algorithm that performs object localization and classification simultaneously in a single forward pass of a convolutional neural network[6]. This design allows YOLO to achieve high detection speed while maintaining strong accuracy compared to two-stage detectors.Continuous architectural improvements across YOLO versions, including enhanced feature extraction and multi-scale detection, have made newer variants more capable of handling small, overlapping, and visually ambiguous objects such as food items.

In recent literature, YOLO-based models have been successfully applied to food detection tasks, demonstrating their effectiveness in identifying and localizing multiple food items within a single image[7]. Studies report that YOLO performs well in real-world food scenarios involving cluttered dining scenes, varied camera angles, and inconsistent lighting conditions. Its real-time inference capability makes it particularly suitable for applications such as dietary assessment systems and smart restaurant solutions. However, most existing research focuses on non-Indonesian cuisines, and the lack of large, diverse Indonesian food datasets remains a major limitation in current studies[8].

Performance evaluation of YOLO-based food detection systems commonly employs standard object detection metrics, including Precision, Recall, Intersection over Union (IoU), and Mean Average Precision (mAP)[9]. Precision

and Recall are used to assess classification reliability and detection completeness, while IoU measures the accuracy of bounding box localization. mAP, calculated over multiple IoU thresholds, is widely used as the primary indicator of overall detection performance[10]. These metrics are consistently adopted in the literature to ensure fair and comprehensive evaluation of YOLO models, enabling reliable comparison across different datasets and detection approaches[11].

III. Methodology

A. Source Dataset

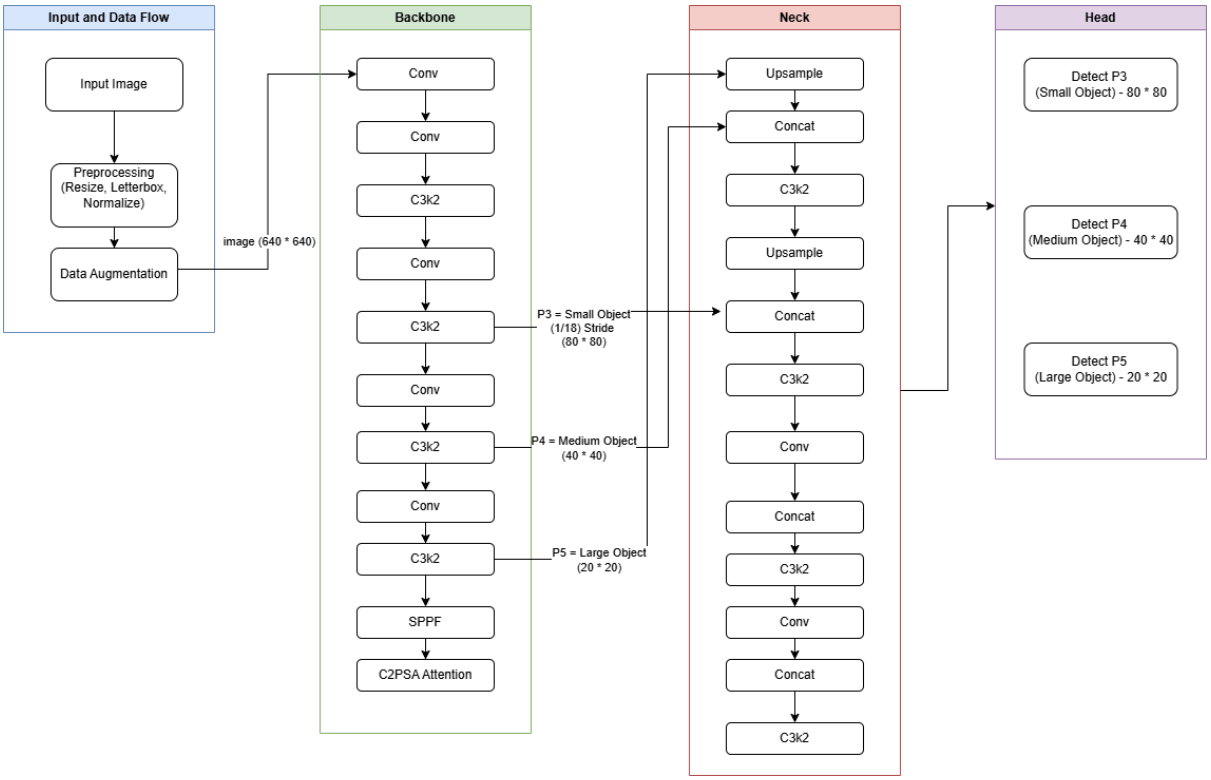
Accurate Indonesian food detection requires a diverse and well-annotated dataset to capture the cuisine's visual complexity. As no single comprehensive public dataset exists for this task, a multi-source aggregation strategy was adopted. The final dataset was constructed by combining fifteen public and community-contributed sources, primarily from Roboflow Universe and the UEC Food 256 dataset on Kaggle. This resulted in bounding box annotations for 42 Indonesian food classes, enabling robust generalization across variations in presentation, lighting, and regional styles. All data were standardized to the YOLO format, followed by rigorous label cleaning and quality filtering to ensure reliability for YOLO11 training.

B. Data Preprocessing

A structured preprocessing workflow was applied to ensure dataset consistency and compatibility with the YOLO11 framework. All collected images and annotations were first standardized into the YOLO format, including class indices and normalized bounding box coordinates. A duplicate and near-duplicate image filtering process was conducted to reduce redundancy and prevent data leakage, while images with incorrect labels or severely degraded visual quality were removed to maintain annotation reliability. The dataset was then divided into training, validation, and test sets using a stratified split to preserve class distribution across all 42 Indonesian food categories. Basic preprocessing steps such as image resizing and aspect ratio preservation were handled automatically by the YOLOv11 input pipeline. Data augmentation techniques, including

scaling, translation, rotation, horizontal flipping, and color adjustments, were applied implicitly through the YOLOv11 training and fine-tuning parameters. This built-in augmentation mechanism allows the model to learn robust visual representations under varying lighting conditions, viewpoints, and food presentation styles without requiring manual image manipulation, thereby improving generalization and reducing overfitting.

C. Model Architecture



D. Training Setup

The object detection model was trained using the YOLOv11 framework with pretrained weights to accelerate convergence and improve generalization. Training was conducted using a fixed input resolution of 640 × 640 pixels and optimized for a maximum of 150 epochs with early stopping enabled to prevent overfitting. Early stopping was triggered based on validation performance when no further improvement was observed over a predefined patience window. Model optimization employed the AdamW optimizer with a cosine learning rate

scheduling strategy to ensure smooth and stable convergence. A moderate batch size was selected to balance training stability and GPU memory constraints, while deterministic training settings and a fixed random seed were used to ensure reproducibility across experiments. Data loading was parallelized using multiple worker threads, and disk-based image caching was enabled to reduce input latency and improve training throughput. Throughout training, model checkpoints were saved automatically, with the best-performing checkpoint selected based on validation mean Average Precision. This checkpoint was subsequently used for final evaluation on the test set.

E. Evaluation Metrics

Model performance was evaluated using standard object detection metrics to assess both classification accuracy and localization quality. Precision and recall were used to measure the correctness and completeness of detected objects, while the F1-score provided a balanced summary of these two metrics. To evaluate localization accuracy, mean Average Precision at an IoU threshold of 0.5 (mAP@0.5) and mean Average Precision averaged across IoU thresholds from 0.5 to 0.95 (mAP@0.5–0.95) were reported. Evaluation was performed on both the validation and test sets to monitor training behavior and assess generalization on unseen data. In addition to aggregate metrics, class-wise performance trends were analyzed to identify variations in detection accuracy across different food categories. These metrics collectively provide a comprehensive assessment of the model's detection capability and robustness in multi-class Indonesian food recognition scenarios.

IV. Implementation & Result

A. System Implementation Details

The model was implemented with the Ultralytics YOLO framework using the YOLO11 model and trained in a PyTorch-based environment with GPU acceleration. The dataset was organized in a YOLO-compliant structure and split into training, validation, and test subsets using a 70/15/15 ratio with stratified sampling to preserve class distributions. Pretrained weights were used for

initialization, and training was conducted for a fixed number of epochs with early stopping based on validation performance. Custom YOLO data augmentation and optimization settings were employed. Model selection was based on the checkpoint achieving the highest validation mAP, ensuring a balance between localization accuracy and classification performance.

B. Experimental Setup

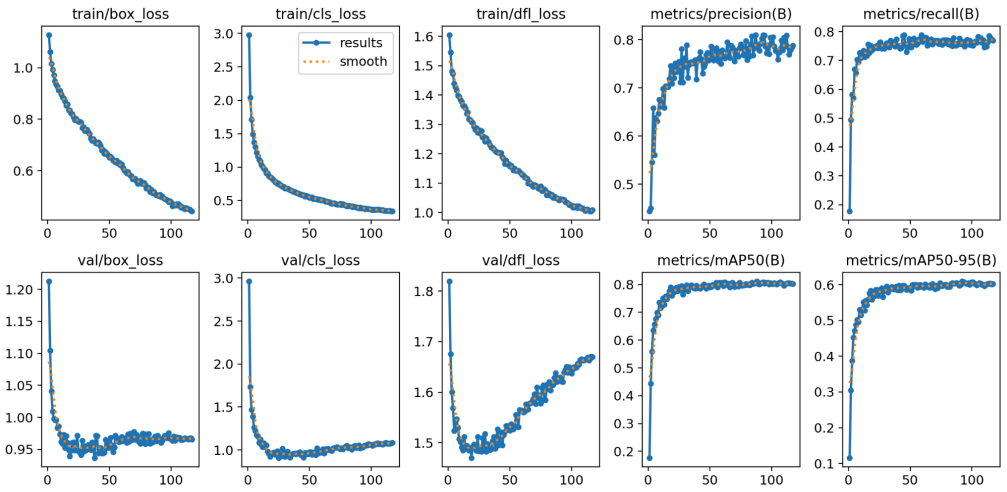
This study focuses on multi-class food object detection, where the model localizes and classifies multiple food items within a single image by predicting bounding boxes, class labels, and confidence scores. Performance is evaluated on unseen data using standard object detection metrics to assess generalization. The dataset follows the YOLO annotation format with normalized bounding box coordinates and zero-based class indexing, defined through a single data.yaml file. It contains 42 food classes, split into 4,989 training, 1,053 validation, and 1,109 test images, ensuring unbiased evaluation and preventing data leakage.

Experiments are conducted using the Ultralytics YOLO11 architecture with 640×640 input resolution, selected to balance accuracy and efficiency. Training is performed for up to 150 epochs with early stopping (patience = 25), using the AdamW optimizer with cosine learning rate decay, a batch size of 16, disk-based caching, deterministic training (seed = 42), and 12 data-loading workers. To enhance generalization, training incorporates HSV color jittering, geometric transformations (translation, scaling, horizontal flipping), and Mosaic augmentation, which is disabled after epoch 15. MixUp and Copy-Paste augmentations are excluded to preserve food appearance consistency.

All experiments are run on a system equipped with an NVIDIA RTX 3070 Ti Laptop GPU (8 GB VRAM), an AMD Ryzen 9 6900HX CPU (8C/16T), and 32 GB RAM, using PyTorch 2.4.1 with CUDA. Model checkpoints are saved automatically, with the best model selected based on validation mAP. Training outputs include loss curves, precision–recall metrics, F1-score analysis, and confusion matrices for subsequent performance evaluation.

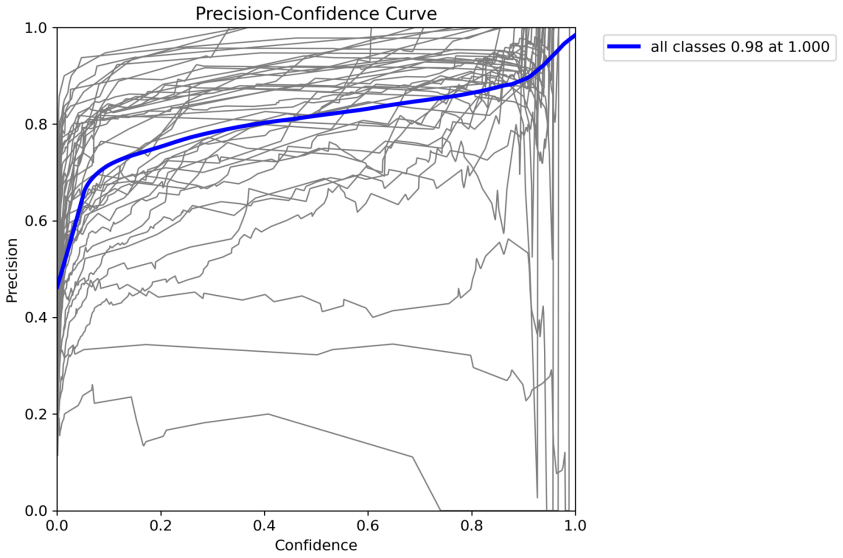
C. Visualizations

1. Training/Val Loss Curves



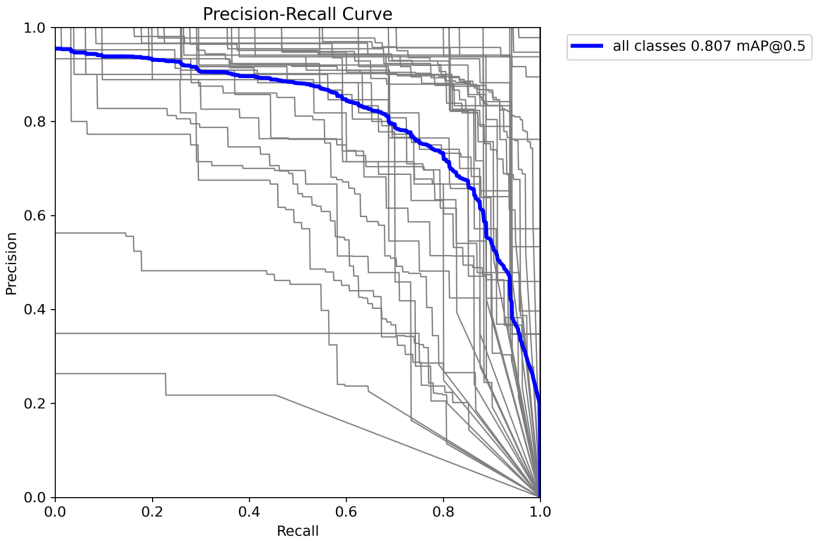
The training dynamics demonstrate stable convergence, effective generalization, and balanced detection behavior. While minor signs of localization overfitting are observed in later epochs, validation performance remains stable, indicating that the selected model is well-suited for final evaluation on the held-out test set.

2. Precision-Confidence Curve



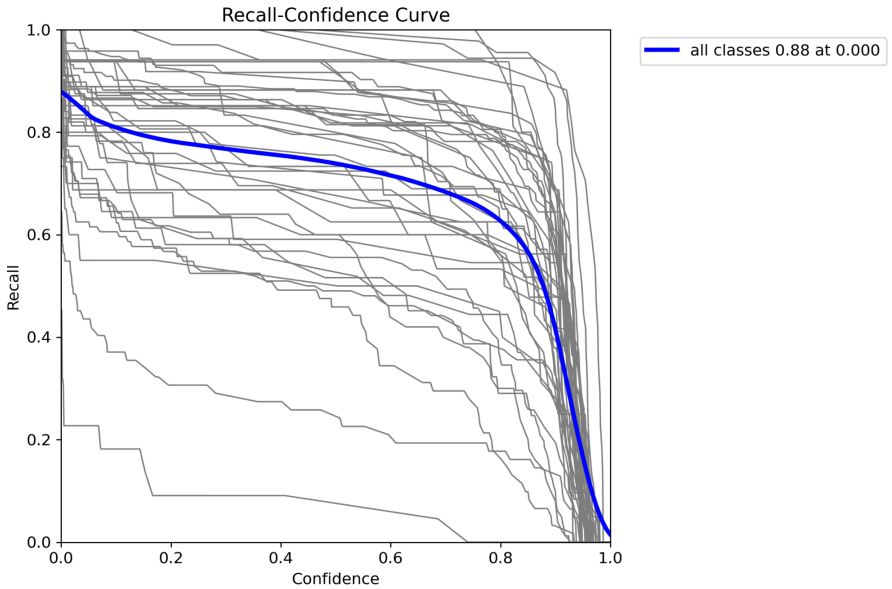
This curve confirms a clear precision–recall trade-off and supports the use of a moderate confidence threshold during inference to balance detection reliability and coverage.

3. Precision-Recall Curve



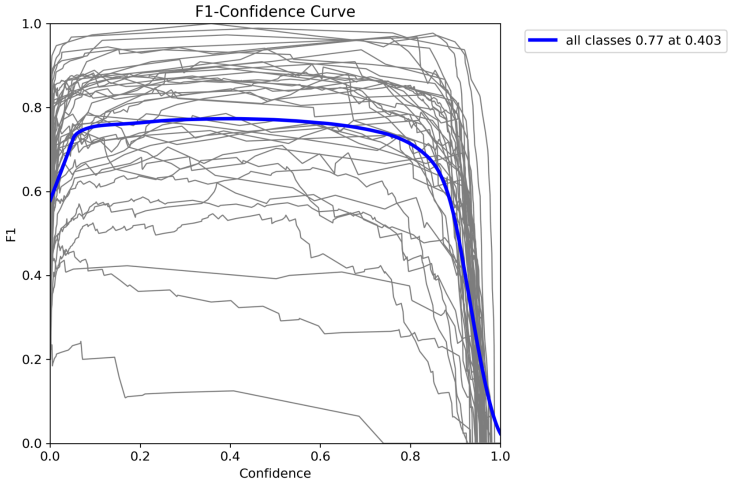
The PR curve confirms that the model maintains robust detection quality while scaling recall, consistent with the quantitative evaluation results reported earlier.

4. Recall-Confidence Curve



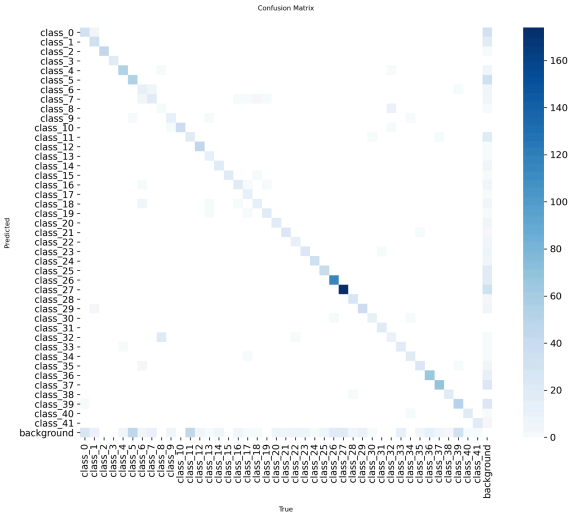
The aggregated curve (blue line) highlights a conservative detection behavior at higher confidence levels, where recall is sacrificed to improve precision. The variation among individual class curves (gray lines) further indicates class-dependent sensitivity to confidence thresholds, particularly for underrepresented or small-object categories.

5. F1-Confident Curve



The relatively smooth plateau of the F1 curve across mid-range confidence values reflects stable prediction behavior and robustness to threshold variation. While individual class curves exhibit variability, particularly for less frequent classes, the overall trend indicates consistent model performance across the dataset.

6. Confusion Matrix



The normalized confusion matrix further highlights that errors are generally sparse and localized, rather than systematic across many classes. Overall, the confusion matrix confirms effective class separation and supports the quantitative performance metrics reported earlier.

D. Result

1. Evaluation Metrics on Val Split Set

Evaluation Metrics	Results
Precision (mP)	0.8007
Recall (mR)	0.7602
mAP@0.5	0.8090
mAP@0.5:0.95	0.7799

The validation results indicate that the proposed model achieves strong and stable detection performance across most food categories, while performance degradation is primarily concentrated in a limited number of underrepresented or visually challenging classes. These findings motivate a deeper examination of class-wise behavior and error patterns in the final test evaluation presented in the subsequent section.

2. Evaluation Metrics on Test Split Set

Evaluation Metrics	Results
Precision (mP)	0.776
Recall (mR)	0.7616
mAP@0.5	0.7947
mAP@0.5:0.95	0.5965
F1-score	0.7687

On the test set, the model achieves a mean precision (mP) of 0.776, mean recall (mR) of 0.762, and an F1-score of 0.769, indicating a well-balanced detection performance. The mAP@0.5 reaches 0.795, demonstrating reliable object detection under moderate localization constraints, while the mAP@0.5:0.95 of 0.597 reflects the increased difficulty of achieving precise localization across stricter IoU thresholds.

The observed gap between mAP@0.5 and mAP@0.5:0.95 is consistent with findings from the exploratory data analysis, particularly the prevalence of

small objects and densely packed food scenes. Overall, these results confirm that the model generalizes well to unseen data and maintains stable detection quality.

3. Evaluate AP50-95 Per Class

Top 10 Best Performance

class_id	class_name	AP50-95
10	class_10	0.806776
19	class_19	0.799181
26	class_26	0.798940
30	class_30	0.792036
20	class_20	0.769546
29	class_29	0.766893
37	class_37	0.757949
17	class_17	0.755597
36	class_36	0.744078
25	class_25	0.732441

Top 10 Worst Performance

class_id	class_name	AP50-95
7	class_7	0.436402
35	class_35	0.423770
1	class_1	0.407964
23	class_23	0.405275
33	class_33	0.371379
5	class_5	0.349307
0	class_0	0.325099
11	class_11	0.317881
8	class_8	0.221598
32	class_32	0.362565

Several classes achieve high average precision, with AP@0.5:0.95 values above 0.78. In particular, class_10, class_19, class_26, and class_30 demonstrate strong and consistent performance, indicating that these categories benefit from clearer visual characteristics and sufficient representation during training. A second group of classes exhibits moderate performance, with AP@0.5:0.95 values ranging between 0.73 and 0.77. While detection quality remains reliable, these classes may experience mild localization challenges or increased intra-class variability. In contrast, a subset of classes shows lower detection performance. Classes such as class_8, class_11, class_0, and class_5 achieve AP@0.5:0.95 values below 0.35, indicating difficulty in both localization and classification. These classes are typically associated with fewer training samples, visually ambiguous appearances, or small object sizes, making them more challenging for the detector.

Additionally, classes with higher representation and clearer visual cues tend to achieve superior performance, while underrepresented or visually similar categories experience reduced average precision. Classes that frequently appear in dense scenes or exhibit small bounding boxes show greater sensitivity to stricter IoU thresholds, contributing to lower AP@0.5:0.95

scores. These findings reinforce the importance of considering dataset imbalance and object characteristics when interpreting detection results.

V. Conclusion & Future Work

1. Conclusion

This study successfully investigated the application of the YOLO11 object detection architecture for the challenging task of multi-class Indonesian food recognition. By meticulously aggregating and validating a dataset spanning 42 distinct food categories, the system achieved a mAP@0.5 of 0.795 on the test set, confirming the viability of deep learning for automated dietary assessment in this complex culinary domain.

The project demonstrated that while the YOLO11 model is robust and stable, achieving balanced performance across most categories, the primary limitations encountered are data-driven. The lower performance observed under stricter mAP@0.5:0.95 thresholds, as well as the majority of qualitative detection errors, are directly attributable to severe class imbalance and the technical difficulty of detecting small and highly overlapping objects inherent in the dataset composition. This affirms that model performance has reached a ceiling imposed by data quality, suggesting the next phase of development must focus on dedicated data enhancement.

2. Future Work

Creation of a Proprietary and High-Quality Dataset. The immediate priority is to transition away from reliance on aggregated public sources to creating a proprietary dataset with standardized imaging, consistent labeling, and balanced class representation.

Expansion of Class Diversity for Comprehensive Coverage. The current 42 classes provide a starting point, but future work should focus on expanding the inventory of Indonesian food classes, as Indonesia possesses a truly massive and diverse cuisine

Integration of Calorie and Nutritional Estimation. The object detection model provides the foundation for an advanced application. The next stage involves integrating the detected bounding box output with a nutritional database to enable real-time calorie and nutrient estimation. This integration would transform the system into a functional application for automated dietary tracking and health management.

VI. References

A. Paper Reference

[1] Wijaya, S. (2019). Indonesian food culture mapping: a starter contribution to promote Indonesian culinary tourism. Journal of Ethnic Foods, 6(1), 9. https://doi.org/10.1186/s42779-019-0009-3

[2]Yu, D., Min, W., Jin, X., Jiang, Q., Jin, Y., & Jiang, S. (2025). Diverse and High-Quality Food Image Generation from Only Food Names. ACM Transactions on Multimedia Computing, Communications and Applications, 21(5), 1-22.

[3]Pranoto, Y. M., Handayani, A. N., Herwanto, H. W., & Kristian, Y. (2025). Optimized image-based grouping of e-commerce products using deep hierarchical clustering. International Journal of Advances in Intelligent Informatics, 11(3).

[4]Subhi, M. A., Ali, S. H., & Mohammed, M. A. (2019). Vision-based approaches for automatic food recognition and dietary assessment: A survey. IEEE Access, 7, 35370-35381.

[5] Ali, M. L., & Zhang, Z. (2024). The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection. Computers, 13(12), 336.

[6] Alhashmi, S. A., & Al-azawi, A. (2025). A Review of the Single-Stage vs. Two-Stage Detectors Algorithm: Comprehensive Insights into Object Detection. International Journal of Environmental Sciences, 11(3s), 775-787.

[7] Vijayakumar, A., & Vairavasundaram, S. (2024). Yolo-based object detection models: A review and its applications. Multimedia Tools and Applications, 83(35), 83535-83574.

[8] Kristia, K., Kovács, S., & Erdey, L. (2024). Generation Z's appetite for traditional food: unveiling the interplay of sustainability values as higher order construct and food influencers in Indonesia. Discover Sustainability, 5(1), 493.

[9] Sonawane, S., & Patil, N. N. (2025). Comparative performance analysis of YOLO object detection algorithms for weed detection in agriculture. Intelligent Decision Technologies, 19(1), 507-519.

[10] Zhang, Y. F., Ren, W., Zhang, Z., Jia, Z., Wang, L., & Tan, T. (2022). Focal and efficient IOU loss for accurate bounding box regression. Neurocomputing, 506, 146-157.

[11] Wang, L., Wang, H., Letchmunan, S., Xiao, R., Ahmed, O. H., & Liu, Z. (2025). A systematic literature review of lightweight YOLO models for object detection. PeerJ Computer Science, 11, e3357.

B. Dataset Reference

Akechie. (2024, December). Food Recognition Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/akechie/food-recognition-7vir7

Apu. (2023, June). Food Classification Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/apu-hme64/food-classification-8mj1l

Bangkit. (2023, December). dataset Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/bangkit-feldj/dataset-9cro2

Bootcamp. (2024, August). indonesian-food Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/bootcamp-o49zr/indonesian-food-uyhxu

Fusion. (2022, June). deteksi makanan indonesia Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/fusion-qvvyj/deteksi-makanan-indonesia

Halo. (2024, April). food recognition Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/halo-jfgrc/food-recognition-6ybvp

Project, M. L. (2025, December). Makanan Indonesia untuk 10 kelas Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/brin-project/makanan-indonesia-untuk-10-kelas-oiybu

Tesis. (2023, June). TRADITIONAL FOOD Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/tesis-3x7b8/traditional-food

Setiawan, A. (2024, August). makananan indonesia Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/asep-setiawan/makananan-indonesia

Maulana, I. (2023, September). Indonesia-Food Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/imam-maulana-b4xet/indonesia-food

Detection, O. (2025, September). Makanan Nutrisi Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/object-detection-vblv5/makanan-nutrisi-1xjal

Utara, U. S. (2023, December). Food Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/universitas-sumatera-utara-h9u4i/food-0yybl

Fona. (2023, December). Food Detection Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/fona/food-detection-yvcmw

ProjectWachi. (2024, December). Boiled-eggs Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/projectwachi/boiled-eggs

Kuo, R. (n.d.). *UEC-Food256 Dataset*. Retrieved from https://www.kaggle.com/datasets/rkuo2000/uecfood256

VII. Appendix

A. Team Contribution Statement

The completion of this project was the result of a collaborative effort with distinct roles defined for each team member. The following outlines the specific contributions of each individual across the project phases:

- Nicholas Victorio : Dataset Collection, Exploratory Data Analysis (EDA) , Data Preprocessing, Model Implementation, Deployment , Report Writing and Proofreading.
- Kenneth Owen : Data Collection, Exploratory Data Analysis (EDA), Data Preprocessing, Final Report Writing, Presentation Slides and Proofreading .
- Nixon Raine Vicsant : Final Report Writing, Presentation Slides, Video Demonstration & Voice Over and Proofreading

B. Github Repository

https://github.com/NicholasVictorio/Object-Detection-for-Indonesia-Food-with-YOLO

C. Screenshoot Demo

Link video demo:
https://drive.google.com/drive/folders/10MgADV_Ur-GIJ3LT88_Sk4zC6RL_78FO?usp=sharing