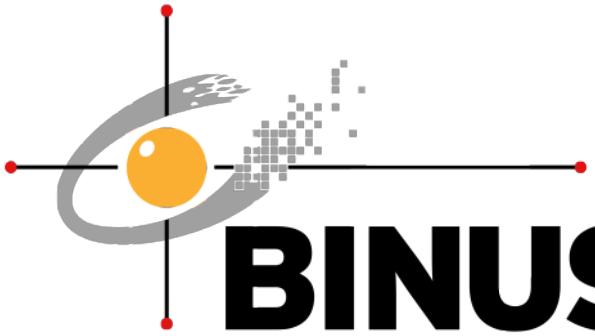


BINUS UNIVERSITY

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 research explores the use of the YOLO11 object detection model to automatically recognize Indonesian food, a task made challenging by the high visual similarity between different dishes and the large variation within the same food categories. A multi-source dataset containing 6,590 images from 40 Indonesian food classes was used to train the model, with particular attention to challenges such as crowded food presentations and imbalanced class distributions. The trained model performed well on previously unseen data, achieving a mAP@0.5 of 0.800, precision of 0.785, and recall of 0.766. Although the model demonstrated stable training behavior and strong generalization, the lower mAP@0.5:0.95 score of 0.591 indicates ongoing difficulty in accurately localizing objects within complex dining environments. Overall, the results show that YOLO11 is a promising approach for Indonesian food detection and provide a solid baseline for future work, including broader food category coverage and the incorporation of real-time calorie and nutritional analysis for dietary monitoring.

I. Introduction

Indonesia has 17,000 islands and hundreds of ethnic groups, resulting in thousands of unique traditional dishes [1]. This diversity presents both challenges and opportunities for automated food recognition systems. On the one hand, there is visual variation within the same dish due to differences in ingredients, cooking styles, and presentation. On the other hand, many different dishes share similar visual characteristics but differ in their presentation. These factors make Indonesian food recognition a challenging problem that requires advanced visual understanding.

Conventional image recognition methods often struggle to address high intra-class variation and inter-class similarity. Challenges such as multiple dishes being served simultaneously limit accurate classification. Challenges such as varying lighting conditions, inconsistent presentation styles, and the complex textures of prepared foods further increase the difficulty of this task.

To address these challenges, this study attempted a detection approach using YOLO11 (You Only Look Once version 11). YOLO11 is a modern single-stage Convolutional Neural Network (CNN) designed to maintain real-time performance while also achieving high accuracy [2]. By training the model on an appropriate Indonesian cuisine dataset, the proposed system is able to learn discriminatory visual features directly from the raw pixel data. This approach enables efficient and reliable detection of Indonesian food.

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 [3]. At the same time, many Indonesian dishes share highly similar visual characteristics, resulting in high inter-class similarity [4]. 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 [5].

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 [2]. 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

Indonesian food detection requires a diverse and well-annotated dataset capable of capturing the visual complexity of Indonesian cuisine. As no single comprehensive public dataset exists for this task, a multi-source aggregation strategy was adopted. The dataset was constructed by integrating multiple public and community-contributed sources obtained from Roboflow Universe, along with the UEC Food-256 dataset hosted on Kaggle. While several Roboflow datasets were already provided in YOLO format, variations in class definitions and label indexing across sources required systematic class harmonization, class ID remapping, and label consolidation.

The UEC Food-256 dataset, which was originally annotated in Pascal VOC format, was converted into the YOLO annotation format. Then, an extensive data cleaning was performed, including the removal of images without corresponding labels, the elimination of duplicate images across classes, the correction of class id mismatches, and the filtering of invalid or empty bounding box annotations. The finalized dataset consists of 6590 images with 11110 bounding box annotations for 40 Indonesian food classes.

B. Data Preprocessing

A structured preprocessing workflow was applied to ensure dataset consistency and compatibility with the YOLO11 framework. Duplicate and near-duplicate images were identified and removed, and samples with incorrect or missing annotations were excluded to maintain label integrity.

No manual image resizing was performed before training. Instead, input images were automatically resized to 640 × 640 pixels during training, following the default input configuration of YOLO11. Pixel value normalization and aspect ratio preservation were handled internally by the YOLO11 input pipeline to ensure consistent input scaling.

Data augmentation was applied during training to improve model generalization while preserving the visual realism of food objects. Augmentation techniques included horizontal flipping, mild scaling (±10%), translation (±10%), and slight rotation (±3°) to simulate common variations in food photography. Color augmentations were limited to brightness and saturation adjustments to account for diverse lighting conditions. Vertical flipping, large rotations, and perspective distortions were deliberately avoided, as such transformations are unrealistic for food presentation scenarios. These augmentations were applied implicitly through YOLO11's training configuration.

Finally, the dataset was divided into training, validation, and test sets using a stratified splitting strategy to preserve class distribution across all 40 Indonesian food categories. The dataset was split into 70% training, 15% validation, and 15% testing subsets to support model training and performance evaluation.

C. Model Architecture

This study adopts the YOLO11 object detection architecture, a single stage deep learning model designed for a real-time object detection with high accuracy and computational efficiency [12]. YOLO11 integrates feature extraction, multi-scale feature aggregation, and bounding box prediction into one end-to-end framework, making it suitable for this task. As illustrated in Figure 1, the YOLO11 architecture consists of three main components: the backbone, the neck, and the head.

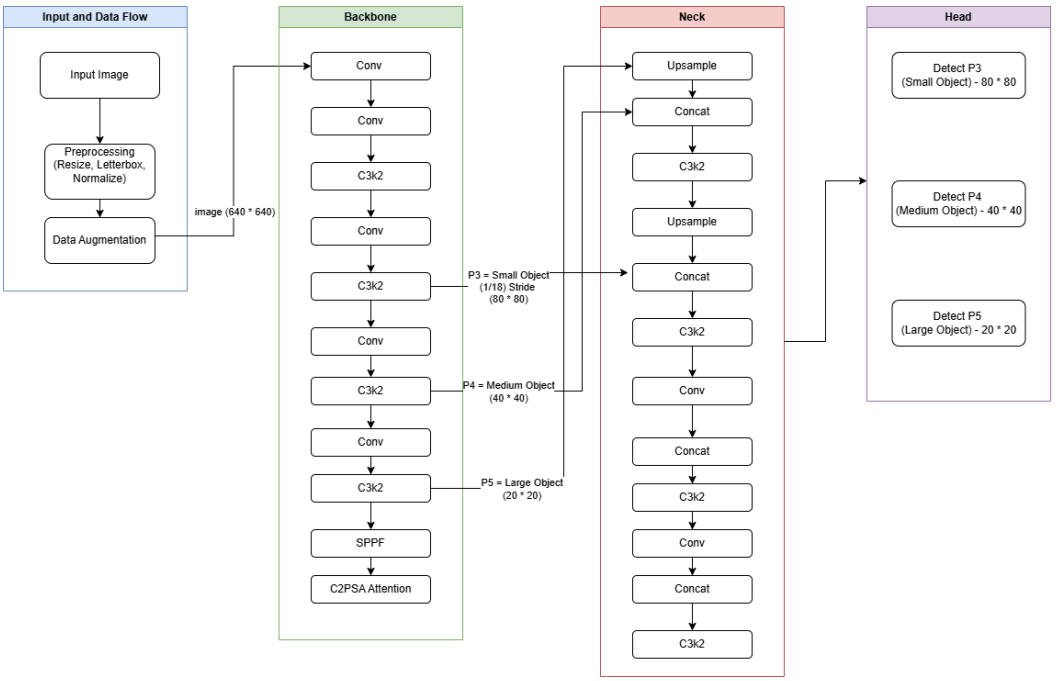


Figure 1. Model Architecture

The backbone is responsible for hierarchical feature extraction from the input images resized to 640 × 640 pixels during training. It is composed of a sequence of convolutional layers and C3K2 modules that progressively reduce spatial resolution while increasing feature depth. These layers enable the model to learn both low-level visual features such as edges and textures, as well as high-level semantic representations related to food objects. At the final stage of the backbone, the Spatial Pyramid Pooling Fast (SPPF) module aggregates contextual information across multiple receptive fields, followed by the C2PSA (Cross-Stage Partial with Spatial Attention) module. The C2PSA module applies spatial and channel attention to refine the most abstract features, ensuring the model focuses on the most discriminative parts of the food items before the data enters the neck [12].

The neck component uses a feature pyramid to combine information from multiple backbone stages. This setup helps the model capture details at different scales. By applying upsampling and concatenation, fine-grained spatial details from earlier layers mix with richer semantic information from deeper layers. Within the neck, additional C3K2 blocks are utilized to process these fused features, which helps in maintaining feature consistency and enhancing the representation of food objects across various scales [12].

The detection head locates and classifies objects at different scales by using separate detection layers for feature maps of various resolutions. This multi-scale approach helps the model reliably detect food items of different sizes. Each detection layer predicts bounding box locations, confidence scores, and class probabilities all at once, allowing YOLO11 to balance accuracy and real-time performance [12].

This study adopts the YOLO11s (small variant) object detection architecture, a single-stage deep learning model designed for real-time object detection with high accuracy and computational efficiency [12]. The YOLO11s variant was selected to balance detection performance and computational cost, making it suitable for training on limited hardware resources while maintaining robust accuracy.

D. Training Setup

All experiments were conducted using the YOLO11s model variant to accommodate GPU memory constraints while maintaining real-time inference capability on consumer-grade hardware. The object detection model was trained using the YOLO11 framework with pretrained weights to accelerate convergence and improve generalization.

All input images were resized to a fixed spatial resolution of 640×640 pixels to ensure consistent feature extraction and efficient GPU utilization. Model training was performed for a maximum of 150 epochs with early stopping enabled using a patience threshold of 15 epochs, reducing the risk of overfitting and unnecessary computation.

Model optimization followed the default YOLO11's training configuration. The AdamW optimizer was employed due to its adaptive learning rate behavior and decoupled weight decay, which improves convergence stability in deep convolutional networks. A cosine learning rate scheduling strategy was applied to gradually reduce the learning rate over the training process, enabling stable convergence and improved final performance. A batch size of 16 was selected to balance gradient stability with GPU memory constraints.

To improve training efficiency, data loading was parallelized using multiple worker threads, and disk-based image caching was enabled to reduce

input/output latency during training. Data augmentation operations, including geometric and photometric transformations, were applied implicitly through the YOLO framework during training to enhance robustness against variations in scale, illumination, and object appearance.

Furthermore, to ensure experimental reproducibility, deterministic training behavior and a fixed random seed were applied across all training runs. Model checkpoints were saved automatically during training. The best-performing validation checkpoints are selected based on the highest validation mAP@50–95 as defined by the YOLO11 fitness function, which will be used for final evaluation.

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 [9]. 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 proposed object detection system was implemented in Python using the Ultralytics YOLO framework built on the PyTorch deep learning library. The system architecture follows a single-stage detection pipeline, where input images are processed through a convolutional backbone, feature aggregation layers, and a detection head in a single forward pass.

All experiments were conducted on a local workstation equipped with hardware acceleration to support efficient training and inference. The system was configured with an NVIDIA RTX 3070 Ti Laptop GPU, 32 GB of system memory, and an AMD Ryzen 9 6900HX processor, running on the Windows 11 operating system. GPU acceleration was enabled using PyTorch version 2.4.1 with CUDA support (cu121), along with CUDA 13.0 drivers and cuDNN to optimize low-level tensor operations.

During both training and inference, input images were processed in RGB format and resized to match the model's expected input resolution. Post-processing operations, including confidence thresholding and non-maximum suppression, were handled internally by the YOLO framework to remove redundant detections and refine prediction outputs. Detection results were generated in the form of bounding boxes with associated class labels and confidence scores, which were used for both quantitative evaluation and qualitative visualization.

The system supports offline batch inference on static images, enabling large-scale evaluation across validation and test datasets. Detection outputs were stored and visualized using bounding box overlays to facilitate result interpretation and error analysis. This implementation design allows for efficient experimentation, reproducibility, and straightforward extension to real-time or deployment-oriented scenarios.

B. Experimental Setup

Experiments were conducted using a fixed train validation test split, where the test set was held out and not used during model training or hyperparameter tuning. All models were trained under identical training configurations to ensure a fair comparison, with only the model scale varied across experiments.

Model performance was evaluated using standard object detection metrics, including precision, recall, mean Average Precision at an Intersection over Union (IoU) threshold of 0.5 (mAP@0.5), and mAP averaged across IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95). These metrics were selected to jointly assess localization accuracy and classification performance.

During evaluation, inference was performed at a fixed input resolution of 640×640 pixels using a consistent confidence threshold and non-maximum suppression settings across all models. All evaluations were performed on the test set using the best-performing checkpoint selected based on validation performance. To ensure reproducibility and fair comparison, identical random seeds and evaluation conditions were maintained across all experiments.

C. Visualizations

A. Training/Val Loss Curves

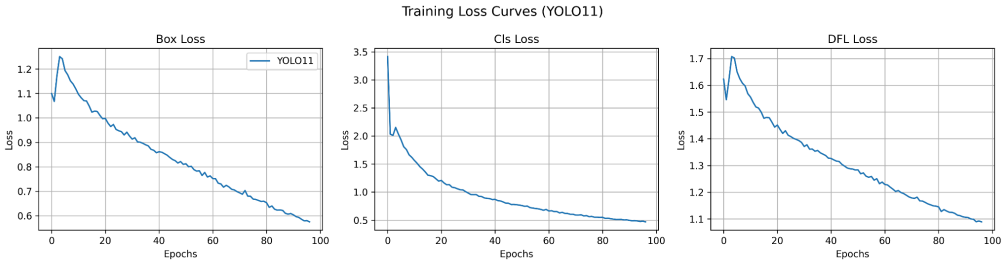


Figure 2. Training Loss Curves

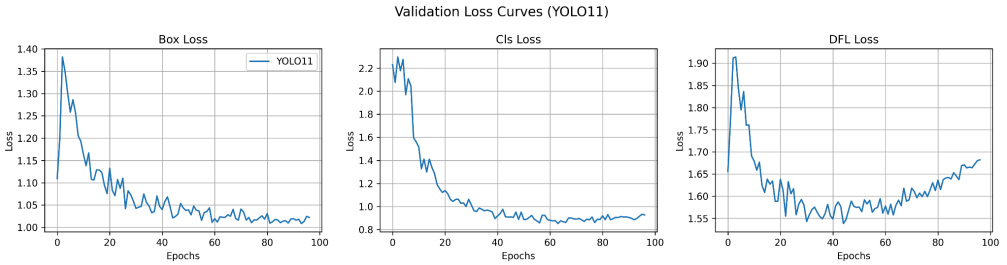


Figure 3. Validation Loss Curves

During training, the YOLO11 model exhibits stable and consistent convergence across all three loss components, such as bounding box regression loss, classification loss, and distribution focal loss, indicating effective optimization throughout the training process. The box loss demonstrates a steady downward trend after the initial epochs, gradually converging to a low and stable value. This behavior suggests that the model increasingly refines its bounding box predictions and achieves improved localization accuracy as training progresses.

Similarly, the classification loss decreases sharply during the early training phase and continues to decline more gradually in later epochs. This pattern indicates that the model quickly learns discriminative class features and steadily

improves its classification performance over time. The distribution focal loss follows a comparable trajectory, showing a consistent reduction across epochs, which reflects progressive improvements in fine-grained localization and bounding box refinement.

The validation loss curves largely mirror the trends observed during training, providing insight into the model's generalization performance. Validation box and classification losses decrease rapidly in the early epochs before stabilizing, indicating that the learned representations transfer well to unseen data. Although the validation distribution focal loss shows minor fluctuations and a slight increase toward the later epochs, the overall trend remains stable, suggesting no severe degradation in localization performance.

Importantly, no significant divergence between training and validation losses was present across any loss component. This indicates that the model does not suffer from pronounced overfitting and maintains balanced learning between the training and validation sets. Overall, the combined analysis of training and validation loss curves confirms that YOLO11 converges reliably, learns robust feature representations, and demonstrates satisfactory generalization performance on the validation data.

B. Confusion Matrix

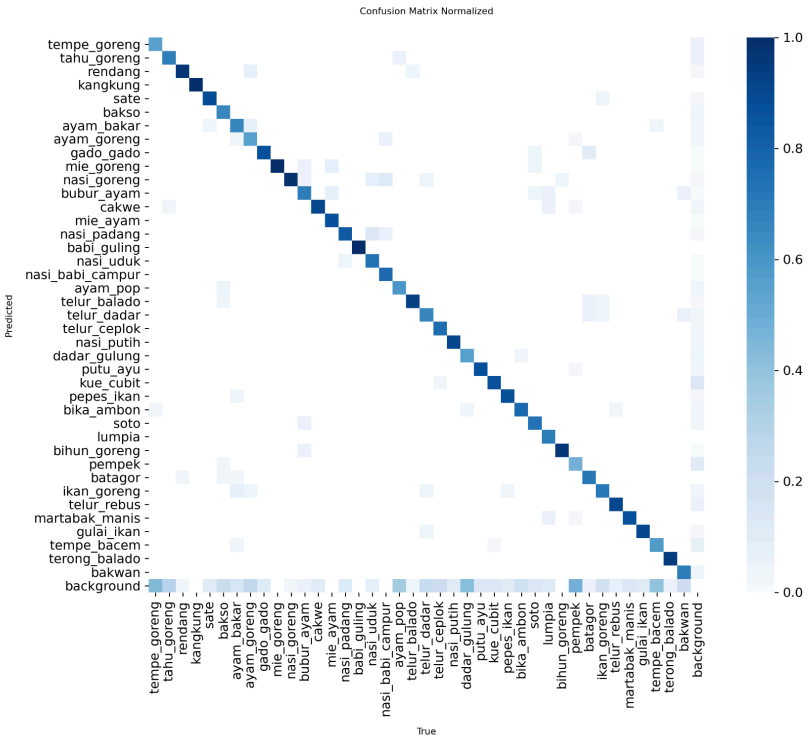


Figure 4. Confusion Matrix

The normalized confusion matrix indicates strong overall classification performance, with most classes exhibiting high values along the main diagonal, reflecting accurate predictions across the majority of Indonesian food categories. This suggests that the model effectively learns discriminative visual features for visually distinct dishes.

Misclassifications are relatively sparse and primarily occur between visually similar food classes, indicating residual inter-class similarity rather than systematic errors. The background class shows minor confusion with certain food categories, suggesting that some food instances are occasionally missed or partially detected, particularly in complex scenes.

Overall, the confusion matrix demonstrates that YOLO11 achieves reliable class-level discrimination, while remaining challenges are mainly attributable to fine-grained visual similarities among certain dishes.

D. Result

a. Evaluation Metrics on Val Split Set

Evaluation Metrics	Results
Precision (mP)	0.7980
Recall (mR)	0.7840
mAP@0.5	0.8319
mAP@0.5:0.95	0.6192
F1-score	0.7909

The validation results indicate balanced detection performance, with a precision of 0.7980 and a recall of 0.7840, suggesting that the model maintains a good trade-off between false positives and false negatives. The achieved mAP@0.5 of 0.8319 reflects strong detection accuracy under moderate localization constraints, while the mAP@0.5:0.95 of 0.6192 highlights the increased difficulty of precise localization at higher IoU thresholds. The

F1-score of 0.7909 further confirms stable and consistent performance across classes on the validation set.

b. Evaluation Metrics on Test Split Set

Evaluation Metrics	Results
Precision (mP)	0.7851
Recall (mR)	0.7661
mAP@0.5	0.8000
mAP@0.5:0.95	0.5910
F1-score	0.7755

On the test set, the model demonstrates comparable performance, with a precision of 0.7851 and recall of 0.7661, indicating effective generalization to unseen data. The mAP@0.5 score of 0.8000 shows a slight decrease compared to validation results, which is expected due to increased data variability. Similarly, the mAP@0.5:0.95 value of 0.5910 indicates that fine-grained localization remains more challenging, while the F1-score of 0.7755 confirms that overall detection performance remains robust without significant degradation.

V. Conclusion & Future Work

A. Conclusion

This study explored the use of the YOLO11 object detection architecture for multi-class Indonesian food recognition. A dataset consisting of 40 Indonesian food categories was constructed and validated. Using this dataset, the proposed system achieved a test-set mAP@0.5 of 0.8000. These results demonstrate that deep learning–based object detection is effective for recognizing Indonesian food in visually complex environments.

The YOLO11s model showed stable learning behavior and consistent performance across most food categories. Performance decreased under stricter localization requirements, as indicated by the lower mAP@0.5:0.95 score. This

limitation is mainly caused by dataset-related factors rather than model capacity. Class imbalance, visual similarity between dishes, and the presence of small or overlapping objects reduce localization accuracy and increase detection difficulty. Overall, the findings indicate that further performance improvements are more likely to be achieved through data-focused enhancements rather than changes to the model architecture.

B. Future Work

A. 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.

B. Expansion of Class Diversity for Comprehensive Coverage
The current 40 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

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

[3] 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.

[4] 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).

[5] 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.

[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.

[12] Khanam, R., & Hussain, M. (2024). YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv Preprint arXiv:2410. 17725*. Retrieved from http://arxiv.org/abs/2410.17725

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

Ta. (2022, December). 7. Pempek Dataset. *Roboflow Universe*. Retrieved from https://universe.roboflow.com/ta-4ec3w/7.-pempek

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:

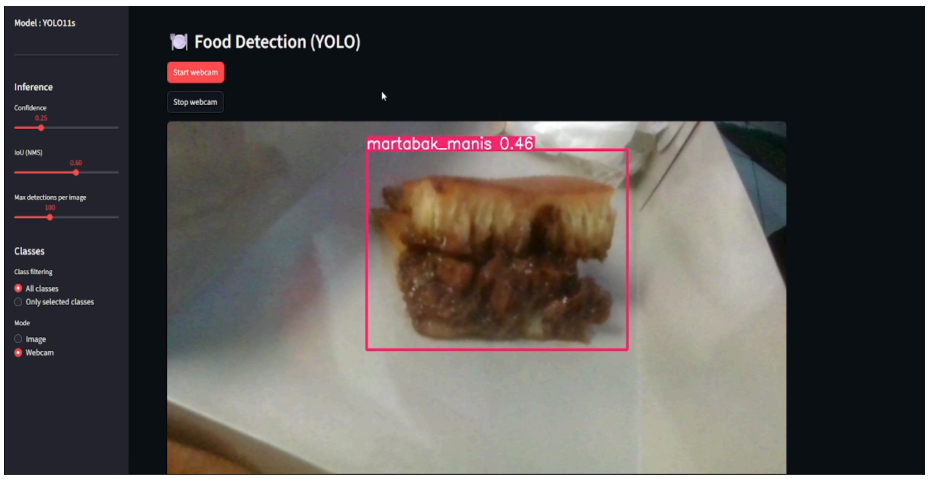
- Nicholas Victorio: Dataset Collection, Exploratory Data Analysis, Preprocessing, Model Implementation, App Deployment, Report Writing, and Proofreading.
- Kenneth Owen: Data Collection, Exploratory Data Analysis (EDA), Data Preprocessing, Final Report Writing, Presentation Slides, and Proofreading.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

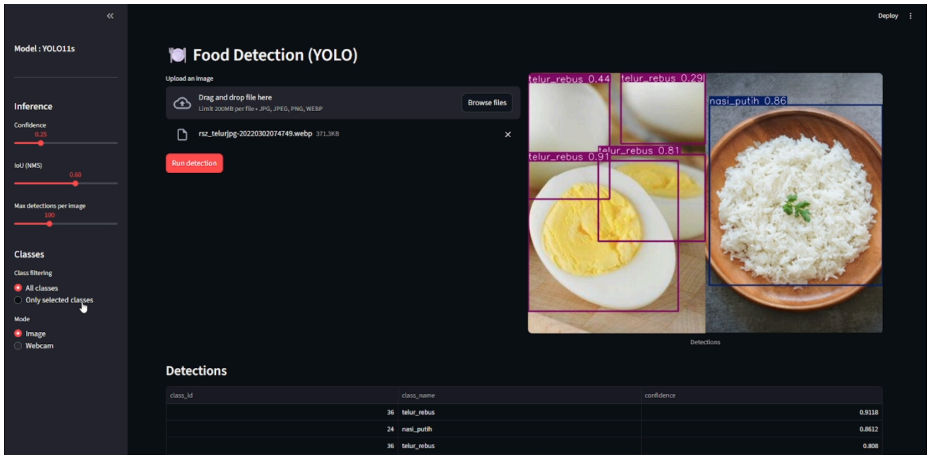
B. Github Repository

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

C. Screenshoot Demo

Link video demo:
https://drive.google.com/drive/folders/1lYvM1GhUd9jEz45E9dfP7diXF2Qf07Qw?usp=sharing


Model : YOLO11s
Food Detection (YOLO)
Start webcam
Stop webcam
martabak_manis 0.46
Inference
Confidence
0.25
IoU (NMS)
0.60
Max detections per image
100
Classes
Class filtering
All classes
Only selected classes
Mode
Image
Webcam


Deploy
Model : YOLO11s
Food Detection (YOLO)
Upload an image
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG, WEBP
Browse files
rsz_telurjpg-20220302074749.webp 371.3KB
Run detection
telur_rebus 0.44
telur_rebus 0.29
nasi_putih 0.86
telur_rebus 0.81
telur_rebus 0.91
Detections
Detections
class_id
class_name
confidence
36 telur_rebus 0.9118
24 nasi_putih 0.8612
36 telur_rebus 0.808
Inference
Classes
Mode
Image
Webcam


Model : YOLO11s
Food Detection (YOLO)
Start webcam
Stop webcam
ayam_goreng 0.80
Inference
Classes
Mode
Image
Webcam