

## COMPARATIVE ANALYSIS OF YOLO DEEP LEARNING MODEL FOR IMAGE-BASED BEEF FRESHNESS DETECTION

Esi Putri Silmina<sup>1,2\*</sup>; Sunardi<sup>3</sup>; Anton Yudhana<sup>3</sup>

Department of Informatics<sup>1</sup>

Universitas Ahmad Dahlan, Yogyakarta, Indonesia<sup>1</sup>

<https://uad.ac.id><sup>1</sup>

2437083003@webmail.uad.ac.id; sunardi@mti.uad.ac.id

Information Technology Study Program<sup>2</sup>

Universitas 'Aisyiyah Yogyakarta, Yogyakarta, Indonesia<sup>2</sup>

<https://www.unisayogya.ac.id>; esiputrisilmina@unisayogya.ac.id\*

Department of Electrical Engineering<sup>3</sup>

Universitas Ahmad Dahlan, Yogyakarta, Indonesia<sup>3</sup>

<https://uad.ac.id><sup>3</sup>

eyudhana@ee.uad.ac.id

(\*) Corresponding Author

(Responsible for the Quality of Paper Content)



The creation is distributed under the Creative Commons Attribution-Non Commercial 4.0 International License.

**Abstract**—Ensuring beef freshness is essential to protect consumer health and maintain public trust in the food supply chain. However, conventional freshness assessment relies on subjective human sensory judgment and can be inconsistent. This study presents a comparative evaluation of three YOLO models, YOLOv5sM (with targeted augmentations Flip, Rotation, Mosaic), YOLOv8, and YOLOv11 for automated beef freshness detection in digital images. Unlike prior studies focusing on a single YOLO version, this work systematically compares multiple YOLO generations to assess accuracy and computational efficiency. Evaluation metrics included precision, recall, mAP@0.5, mAP@0.5:0.95, and training time. A labeled dataset of 4,000 beef images (fresh and non-fresh) was split into training, validation, and test sets, with augmentation applied only to YOLOv5sM. All three models achieved 100% precision and recall on the test set; however, this likely reflects dataset homogeneity and potential overfitting, limiting interpretation of these results. YOLOv11 achieved the highest localization accuracy ( $mAP@0.5:0.95 = 97.0\%$ ), followed by YOLOv8 (96.9%) and YOLOv5sM (96.2%). YOLOv8 had the shortest training time (54 minutes), whereas YOLOv11 offered the best balance of accuracy, model size (5.4 MB), and computational efficiency. Overall, YOLOv11 emerged as the optimal model, offering superior performance and practical deployment advantages over earlier YOLO versions. As the first systematic comparison of multiple YOLO generations for beef freshness detection, this study provides novel insights into detection accuracy and computational efficiency.

**Keywords:** beef freshness, deep learning, object detection, YOLO

**Intisari**—Menjaga kesegaran daging sapi sangat penting untuk melindungi kesehatan konsumen dan menjaga kepercayaan publik terhadap rantai pasok pangan. Namun, penilaian kesegaran secara konvensional masih bergantung pada penilaian sensorik manusia yang bersifat subjektif dan tidak konsisten. Penelitian ini menyajikan evaluasi komparatif terhadap tiga model YOLO, YOLOv5sM (dengan augmentasi Flip, Rotasi, dan Mosaic), YOLOv8, dan YOLOv11 untuk deteksi otomatis kesegaran daging sapi berbasis citra digital. Berbeda dengan studi sebelumnya yang hanya berfokus pada satu versi YOLO, penelitian ini secara sistematis membandingkan beberapa generasi YOLO dalam hal akurasi dan efisiensi komputasi. Evaluasi



dilakukan menggunakan metrik *precision*, *recall*, *mAP@0.5*, *mAP@0.5:0.95*, dan waktu pelatihan. Dataset berisi 4.000 citra daging sapi (segar dan tidak segar) dibagi menjadi data pelatihan, validasi, dan pengujian, dengan augmentasi hanya diterapkan pada YOLOv5sM. Ketiga model menunjukkan *precision* dan *recall* sebesar 100% pada data uji; namun, hasil ini kemungkinan besar mencerminkan homogenitas dataset dan potensi *overfitting*, sehingga membatasi interpretasi performa sebenarnya. YOLOv11 mencatat akurasi lokalisasi tertinggi (*mAP@0.5:0.95* = 97,0%), diikuti oleh YOLOv8 (96,9%) dan YOLOv5sM (96,2%). YOLOv8 memiliki waktu pelatihan tercepat (54 menit), sementara YOLOv11 menunjukkan keseimbangan terbaik antara akurasi, ukuran model (5,4 MB), dan efisiensi komputasi. Secara keseluruhan, YOLOv11 tampil sebagai model paling optimal dengan keunggulan kinerja dan kesiapan implementasi. Sebagai studi pertama yang membandingkan beberapa generasi YOLO untuk deteksi kesegaran daging sapi, penelitian ini memberikan wawasan baru mengenai keseimbangan antara akurasi deteksi dan efisiensi komputasi.

**Kata Kunci:** deteksi objek, deep learning, kesegaran daging sapi, YOLO.

## INTRODUCTION

Beef is a crucial source of animal protein and essential nutrients for human nutrition [1][2][3], making it one of the most widely consumed foods. However, unscrupulous vendors may mix fresh beef with non-fresh or spoiled meat to increase profits, undermining consumer trust and posing health risks. This adulteration can lead to foodborne illnesses and other health problems [4][5]. Conventional methods for assessing meat freshness rely on human sensory evaluations (visual inspection, odor, and texture), which are inherently subjective and require specialized expertise [6][7][8].

Such traditional techniques are often inconsistent and can yield inaccurate results. These limitations motivate the development of automated, objective detection systems based on digital imaging and artificial intelligence, which can assess meat freshness more reliably and efficiently.

The You Only Look Once (YOLO) family of single-stage object detectors has evolved rapidly since its introduction by Redmon et al. in 2016 [9][10]. Subsequent versions have continuously improved accuracy and efficiency. YOLOv2 [11] introduced enhancements to the original architecture, and YOLOv3 [12][13] further refined these improvements, enhanced the architecture. YOLOv4 [14][13], incorporated advanced augmentation (Mosaic) to boost generalization, and YOLOv5 [15] was the first PyTorch implementation, simplifying deployment and optimization. Later releases achieved state-of-the-art performance: YOLOv6 [16] enhanced speed and practicality, YOLOv7 [17] delivered top real-time detection accuracy, YOLOv8 [11] (Ultralytics) further increased speed and accuracy, and recently YOLOv9 [19], YOLOv10 [20], and YOLOv11 [21] have been proposed for specialized applications.

Multiple studies have benchmarked these YOLO variants across diverse tasks. For example, in smoke/forest-fire detection, YOLOv9, YOLOv10, and YOLOv11 were compared achieving high precision (84.5%) and recall (80.1%) [22]. In weed species detection, comparisons among YOLOv8-v11 and Faster-RCNN found YOLOv11 to be the fastest model while YOLOv9 attained the highest average precision [23]. In detecting parasite eggs, YOLOv5, YOLOv7, YOLOv8, and YOLOv10 were evaluated, with YOLOv7-tiny achieving the highest mAP (98.7%) and YOLOv8-n ensuring the fastest inference [24].

YOLO models have also been applied in automotive and medical imaging. In Advanced DriverAssistance Systems (ADAS), combinations like YOLOv5 or YOLOv8 with tracking achieved real-time performance [25]. For pothole detection, YOLOv9 outperformed YOLOv10 and YOLOv11 [26]. In medical imagery, YOLOv5, YOLOv8, and YOLOv9 were compared on laryngoscopy images, with YOLOv8 obtaining the highest mean average precision [27]. These studies illustrate that different YOLO versions offer varied trade-offs between speed and accuracy depending on the task.

In the domain of protein or food freshness detection, earlier approaches used classical methods. For instance, fish freshness was classified using K-Nearest Neighbors and Naïve Bayes on eye image features, achieving around 97% accuracy [28]. A Mask R-CNN model segmented fish images and achieved 96.5% accuracy [6].

In the case of beef detection, researchers implemented YOLOv5 combined with a ResNet classifier and achieved 99% classification accuracy with an mAP of 89% [29].

Specifically for beef, a YOLOv5 model combined with a ResNet classifier attained 99% classification accuracy [30]. Another study used fluorescence changes (hypoxanthine levels) to detect freshness in beef, chicken, and pork,



achieving 99.6% accuracy [31]. Furthermore, the performance of YOLOv5 was shown to vary with the number of object classes: training on eight classes yielded significantly better precision and recall than training on two or ten classes [32]. These findings highlight the potential of deep learning for meat freshness assessment and the importance of model selection and training strategy. Unlike earlier research that typically evaluated a single YOLO version for meat freshness detection, this study is the first to directly compare three YOLO generations under identical experimental conditions, highlighting differences in accuracy, computational efficiency, and deployment suitability. Recent studies on YOLO in food freshness detection [33][34][35][36] further validate the novelty of this work.

Despite these advances, a systematic comparison of modern YOLO models for beef freshness detection remains unexplored. We explicitly aim to evaluate and compare three YOLO architectures (YOLOv5sM, YOLOv8, and YOLOv11) on a dataset of digital beef images labeled fresh or non-fresh. We hypothesize that the newer models (YOLOv8, YOLOv11) will offer better accuracy and efficiency trade-offs than the older YOLOv5sM. Our objectives are to train each model under consistent conditions and to assess their performance using accuracy metrics (Precision, Recall, mAP@0.5, mAP@0.5:0.95) as well as computational criteria (training time and model size).

This study's novelty lies in its focused comparative analysis of YOLO variants for beef freshness classification, including consideration of training efficiency and model size for deployment in resource-constrained settings. Key contributions include:

1. Implementing and optimizing three YOLO models (YOLOv5sM, YOLOv8, YOLOv11) specifically for automated beef freshness detection.
2. Employing selective data augmentation (horizontal flip, rotation, mosaic) exclusively on the YOLOv5sM training to investigate its impact, while training YOLOv8 and YOLOv11 on original images.
3. Conducting a comprehensive evaluation of model performance, combining detection accuracy (Precision, Recall, mAP metrics) with efficiency metrics (training duration, model file size), to determine the optimal model for practical use in environments with limited computational resources.

## MATERIALS AND METHODS

### Dataset

This study uses a dataset of 4,000 beef images categorized into fresh and not fresh meat. We obtained these images from the public data source Kaggle.

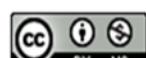
### Preprocessing

Before training, the images undergo a pre-processing stage to ensure quality and reliability. We resize all images to a resolution of 416x416 pixels to meet the standard input of the YOLO Model [37]. The 416x416 pixel resolution was chosen to balance computational efficiency and detection accuracy, allowing the model to effectively process images while maintaining a manageable computational load, ensuring wider accessibility for deployment in environments with different models [38]. We do not apply any additional filtering or enhancement (e.g., no noise reduction or manual color adjustment), so that the models learn from the raw data, reflecting real-world conditions.

### Data Augmentation Technique

Data augmentation is used to overcome the problem of overfitting and is proven to improve model robustness in Deep Learning, by expanding the dataset and exploring the number of features of the training data [39].

We selectively apply data augmentation techniques in the YOLOv5sM model to evaluate its effect in detecting beef freshness, which is an object with high complexity, especially in images with variations in texture, color, and lighting. The YOLOv5sM model, denoting YOLOv5 small augmented with Mosaic, Flip, and Rotation, is a modified variant of the standard YOLOv5s architecture. While Mosaic augmentation was originally introduced in YOLOv4 and supported in the Ultralytics YOLOv5 framework, it is not enabled by default in YOLOv5s. In this study, the 'sM' label explicitly reflects a manual modification by the authors, in which targeted augmentations were intentionally applied during training to improve robustness in visually subtle beef freshness detection tasks. This configuration serves as a controlled benchmark to assess the specific effect of classic augmentation techniques, compared to newer models like YOLOv8 and YOLOv11 that were trained without such augmentations [40]. Augmentation was applied only to YOLOv5sM to test whether traditional augmentation methods (Flip, Rotation, Mosaic) could compensate for limitations of older architectures, while YOLOv8



and YOLOv11 were kept unaugmented to isolate the effects of architecture from augmentation.

This study employs three data augmentation techniques to improve the performance of the YOLOv5sM model in detecting beef freshness from digital images. First, horizontal flipping is applied to the training images to introduce variations in object orientation, allowing the model to better recognize patterns from different viewpoints [41]. Second, random rotation is used to rotate the images at arbitrary angles within a defined range, which enhances the model's robustness against shifts in the camera or object perspective [41]. Finally, the Mosaic technique combines four different images into a single composite image, increasing contextual richness and scene diversity within each training batch, thereby improving the model's generalization capability across complex visual environments [42].

Equation 1 shows the application of augmentation transformations  $\mathcal{A}$  to input  $X$ .  $\mathcal{A}$  to the input data  $X$ .

$$\begin{aligned} X' = \mathcal{A}(X) = \\ \{\text{flip}(x), \text{rotate}(x, \theta), \text{mosaic}(x_1, x_2, \\ x, x_i \in X, \theta \in [-\theta_{max}, +\theta_{max}]\} \quad (1) \end{aligned}$$

### Architecture and Loss Function of YOLO Model

The three YOLO models (YOLOv5sM, YOLOv8, YOLOv11) use the principle of a single-stage CNN-based object detector. We outline their working principles as follows:

#### 1. Output Representation

YOLO is a *single-stage* CNN detector that processes an entire image in one pass. The input image  $x \in \mathbb{R}^{H \times W \times 3}$  is divided into a  $S \times S$  grid, and each grid cell  $(i, j)$  is responsible for detecting objects whose centers fall in that cell [9]. Each cell predicts  $B$  bounding boxes, for each box it outputs five numbers and a class probability vector. Concretely, each bounding-box prediction includes the box center  $(x, y)$  relative to the cell, the box width and height  $(w, h)$  (relative to the image), and an objectness confidence score  $p$  (which reflects both the probability of an object being present and the box IoU with ground truth) [9][43]. In addition, each cell predicts a vector of  $K$  *conditional* class probabilities  $p(c)$  (one value per class) that indicate the class of the object given that *some* object exists in the cell [12]. Altogether, since there are  $B$  boxes with 5 values each, plus  $K$  class scores per cell, the final network output has shape  $S \times S \times (B \times 5 + K)$ .

Formally, for grid cell  $(i, j)$  and box index  $b$ , we can denote the predicted output vector as shown in equation 2 [43].

$$\begin{aligned} \hat{y}_{ijb} \\ = \left( \hat{x}_{ijb}, \hat{y}_{ijb}, \hat{w}_{ijb}, \hat{h}_{ijb}, \hat{p}_{ijb}, \hat{p}_i(1), \dots, \right. \\ \left. \hat{p}_i(K) \right) \quad (2) \end{aligned}$$

Here  $\hat{x}_{ijb}$  and  $\hat{y}_{ijb}$  denote the relative coordinates of the bounding box center within the cell,  $\hat{w}_{ijb}$  and  $\hat{h}_{ijb}$  denote the normalized width and height (scaled to the image), and  $\hat{p}_{ijb}$  is the predicted confidence for box  $b$ . The entries  $\hat{p}_i^{(k)}$  where  $k = 1, \dots, K$  represent the predicted class probabilities for each class  $k$  at grid cell for cell  $(i, j)$ , conditioned on an object being present. During inference the class-specific confidence score for class  $c$  and bounding box  $b$  is taken as  $\hat{p}_{ijb} \times \hat{p}_i^{(c)}$ . This formulation allows YOLO to simultaneously perform object localization and classification in a single forward pass, enabling real-time efficiency with robust performance.

### 2. Loss function on YOLO

The YOLO loss function is composed of three main components that collectively optimize the model for accurate and efficient object detection [12][44][45].

1. Classification loss penalizes discrepancies between the predicted class probabilities and the ground truth, ensuring correct category identification.
2. Localization loss (also known as bounding box regression loss) measures the difference between the predicted bounding box parameters and the ground truth, focusing on spatial precision.
3. Confidence loss evaluates the correctness of the objectness score, distinguishing between object and background regions.

These three components, classification, localization, and confidence, are combined into a single total loss function, which the model optimizes during training. The comprehensive loss function serves as a unified objective that balances detection accuracy with precise localization and reliable confidence estimation, as formally defined in Equation (3) [43] [46].

$$\mathcal{L}_{YOLO} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \quad (3)$$



$$\begin{aligned}
 & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 \right. \\
 & \quad \left. + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (c_i - \hat{c}_i)^2 \\
 & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (c_i - \hat{c}_i)^2 \\
 & + \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2
 \end{aligned}$$

The YOLO total loss function, denoted as  $\mathcal{L}_{YOLO}$  combining multiple components, including coordinate loss, confidence loss, and classification loss. The model assigns a higher weight to coordinate loss using a coefficient  $\lambda_{coord}$  which is typically set to 5, to emphasize the accuracy of bounding box localization. The summation  $\sum_{i=0}^{S^2}$  iterates over all grid cells in the image, where  $S \times S$  represents the total number of grid cells. Each cell predicts  $B$  bounding boxes, and the summation  $\sum_{j=0}^B$  includes all these predictions. The indicator function  $1_{ij}^{obj}$  activates (value 1) if an object is present in cell  $i$  and bounding box  $j$  is responsible for the prediction; otherwise, it is 0. The complementary term  $1_{ij}^{noobj}$  is defined as  $1 - 1_{ij}^{obj}$  serving as a penalty for false positives. The notation with a hat symbol (e.g.,  $\hat{x}$ ) indicates predicted values. Additionally, the loss function uses  $\lambda_{noobj}$  as a penalty coefficient to reduce the influence of predictions made in grid cells without any objects, thereby improving detection precision.

### Key Differences in YOLOv5sM, YOLOv8, and YOLOv11

#### 1. YOLOv5sM with Data Augmentation

YOLOv5sM uses CSPDarknet backbone and PANet neck, and applies data augmentation such as Flip, Rotation, and Mosaic to enrich the variety of training data.

#### 2. YOLOv8 (anchor-free)

YOLOv8 adopts an anchor-free architecture with a decoupled detection head that separates bounding box prediction and classification. The bounding box loss function uses Complete IoU ( $CIoU$ ) loss as shown in equation 4 [47][48].

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \quad (4)$$

This loss function not only evaluates the overlap between predicted and ground truth boxes but also incorporates spatial and aspect ratio constraints to improve convergence. Specifically, the  $CIoU$  loss introduces a penalty based on the Euclidean distance  $\rho$  between the center points of the predicted and ground truth bounding boxes, normalized by the diagonal length  $c$  of the smallest enclosing box. In addition, the term  $\alpha v$  accounts for the difference in aspect ratio between the two boxes, where  $\alpha$  serves as a balancing coefficient. This combination allows YOLOv8 to produce more accurate and stable bounding box predictions across varied object shapes and sizes.

#### 3. YOLOv11

YOLOv11 implements the attention mechanism on the backbone and neck, and the loss function uses Generalized  $IoU$  ( $GIoU$ ) loss as equation 5 [48][49].

$$\begin{aligned}
 \mathcal{L}_{GIoU} = 1 - IoU \\
 + \frac{|\mathcal{C} - (A \cup B)|}{|\mathcal{C}|} \quad (5)
 \end{aligned}$$

The  $GIoU$  loss extends the conventional  $IoU$  by introducing a penalty for non-overlapping bounding boxes, making it more effective in scenarios with poor localization. Specifically, the  $GIoU$  loss calculates the difference between the  $IoU$  score and the normalized area of the smallest enclosing box ( $\mathcal{C}$ ) minus the union of the predicted ( $A$ ) and ground truth ( $B$ ) boxes. This approach allows YOLOv11 to better align predicted bounding boxes with ground truth, especially in complex scenes where objects are partially occluded or closely located.

The differences in loss functions among the models have practical implications for detection performance. YOLOv5sM employs a  $GIoU$ -Based loss, which focuses on bounding box overlap but may be less sensitive to spatial misalignment, particularly for small or irregularly shaped objects. In contrast, YOLOv8 and YOLOv11 adopt  $DIoU$ / $CIoU$  losses that incorporate both overlap and the distance between bounding box centers, as well as aspect ratio consistency. These additional penalty terms enable more accurate localization of small or partially occluded objects, reducing false positives and improving detection precision under challenging visual conditions.

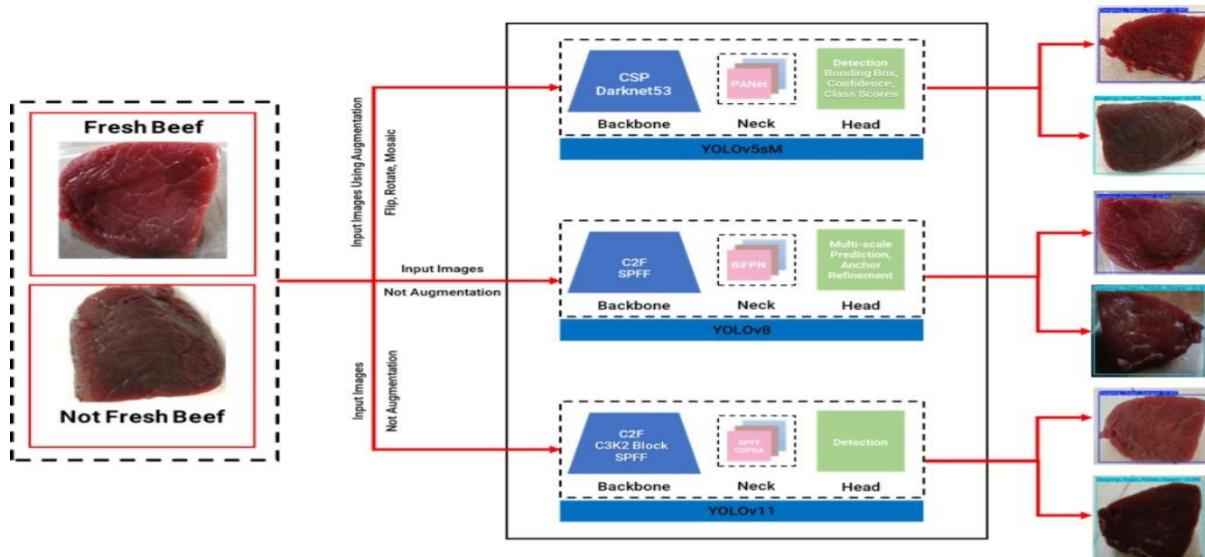
While all three models are based on the YOLO object detection architecture, they were trained under different data preparation schemes. Only YOLOv5sM was deliberately configured with an augmentation pipeline comprising Mosaic, Flip, and



Rotation. This augmentation strategy was explicitly enabled by the authors and is not a default or intrinsic feature of the standard YOLOv5s model. In contrast, YOLOv8 and YOLOv11 were trained using unaugmented original images, allowing this study to isolate and compare the impact of conventional augmentations on performance. This differential setup was intentional to evaluate whether basic augmentation can significantly improve model robustness and generalization in visually subtle tasks such as meat freshness detection.

### Research Design

This study employs a comparative experimental design to measure the performance of three object detection models, namely YOLOv5sM, YOLOv8, and YOLOv11. We apply each model to the same set of fresh and non-fresh beef images to detect the meat's freshness status. We apply each model to the same set of fresh and non-fresh beef images to detect the meat's freshness status. Figure 1 illustrates the overall research flow.



Source: (Research Results, 2025)

Figure 1. Research Flow Diagram

### Model Training

Each model trains on the same dataset using identical parameters and procedures. We split the dataset into 70% training data, 20% validation data, and 10% test data (70-20-10) [50]. This division is via stratified random sampling to maintain balanced representation of fresh and non-fresh categories. We train all models for a fixed 100 epochs and monitor validation performance after each epoch. We deliberately avoid early stopping or validation-based checkpoint selection to ensure consistent comparison across models. However, this choice raises the risk of overtraining, because early stopping is a well-established regularization technique that halts training when the validation loss stops improving, thereby preventing overfitting and often preserving model generalizability [51][52]. To mitigate this risk, we analyze validation learning curves post-training, looking for signs of overfitting, such as rising validation loss or plateauing accuracy in later epochs. We report these trends in the Results section to support our training strategy's transparency and justify not using early stopping.

### Model evaluation

The final stage of the process is model evaluation, which measures each model's performance on unseen test data. After training, we select the model checkpoint with the best validation performance and evaluate it on the test set. This test evaluation yields the final metrics (precision, recall, mAP) for each model, which we use for the comparative analysis.

### Hardware and Computing Environment

We use Google Colab platform equipped with an NVIDIA Tesla T4 GPU. The Tesla T4 GPU has 16 GB of memory and sufficient computational capability for training medium-sized YOLO models. During the experiments, we utilize Python runtimes with Deep Learning libraries (such as PyTorch), as per the implementation needs of each model (Ultralytics YOLOv5/YOLOv8, etc.). All models were trained individually in the same environment to ensure there were no differences in computational conditions. The use of Google Colab allowed for easy standardization of the environment (with



compatible CUDA/cuDNN) and free GPU access, albeit limited to specific time sessions. This hardware configuration was noted as it affected the observed training time and memory usage, which was later reported as part of the experimental results.

## RESULTS AND DISCUSSION

### Detection Result

Table 1 shows the training results for 100 epochs, the three YOLO models show a consistent trend of decreasing loss and increasing accuracy. YOLOv11 recorded the best performance with the lowest train loss value of 0.045 at the 100th epoch, followed by YOLOv8 at 0.072 and YOLOv5sM at 0.154. This is also reflected in the mAP@0.5 value during training, where YOLOv11 achieved a score of 99.9%, higher than YOLOv8 at 99.7% and YOLOv5sM at 99.2%. In addition, YOLOv11 also shows better training time efficiency, requiring only 22 seconds per epoch on average, compared to 28 seconds for YOLOv8 and 36 seconds for YOLOv5sM.

Table 2 shows the validation results reinforcing the superiority of YOLOv11, achieving the lowest val loss (0.058) and the highest mAP@0.5:0.95 of 99.5 at the 100th epoch. YOLOv8 came in second with a val loss of 0.092 and mAP@0.5:0.95% of 98.2%, while YOLOv5sM recorded a val loss of 0.202 and mAP@0.5:0.95 of 96.2%. This comparison shows that the architectural improvements in YOLOv11 contribute significantly to improving the accuracy and efficiency of the model in detecting objects with precision. Thus, YOLOv11 can be recommended as the best model in the digital image-based meat freshness detection task in this scenario.

The superiority of YOLOv11 can be understood in light of its architectural and loss-function innovations, which have been shown to improve detection performance in other domains. In particular, YOLOv11 replaces the YOLOv8 backbone's CSP C2f blocks with a novel C3k2 module (two smaller convolutions) and adds a Cross-Stage Partial block with Spatial Attention (C2PSA) after the SPPF stage [53]. These changes enhance feature extraction and force the network to focus on the most relevant image regions. In parallel, YOLOv11 uses generalized IoU (GIoU)-based bounding-box loss (and related novel loss terms) to improve localization precision [54]. Together, these technical enhancements explain why YOLOv11 attains higher precision/recall and mAP: sharper attention and more precise box regression reduce missed detections and false positives. Indeed, recent studies report results

consistent with our findings. Cerqueira et al. found that YOLOv11 achieved the highest precision and recall on a food-plate waste dataset – significantly outperforming YOLOv8 and nearly matching YOLOv5's precision [55]. Likewise, Sapkota et al. showed YOLOv11-based models achieving superior mask and box mAP in fruit-segmentation tasks compared to YOLOv8 [56]. Then comparing YOLOv5, YOLOv8, and YOLOv11 on solar panel defect detection, found that YOLOv11 achieved the highest mAP@0.5 (93.4%) while maintaining efficient inference speed (~7.7 ms), thus outperforming YOLOv8 and YOLOv5 in both accuracy and processing time [57]. In summary, these reports attribute YOLOv11's gains to its improved CSP blocks, spatial-attention modules, and optimized loss functions – exactly the factors that, in our experiments, gave YOLOv11 an edge in detecting subtle freshness cues in beef. Thus, the literature supports that YOLOv11's architectural refinements directly translate into the accuracy advantage observed in this study.

Table 1. Evaluation Results During Training

Epoch	Model	Train Loss	Train mAP@0.5 (%)	Time/Epoch (s)
10	YOLOv5sM	2.876	62.4	36
	YOLOv8	2.215	71.5	28
	YOLOv11	1.876	80.2	22
20	YOLOv5sM	1.954	78.2	36
	YOLOv8	1.423	84.2	28
	YOLOv11	1.035	89.2	22
30	YOLOv5sM	1.254	86.2	36
	YOLOv8	0.876	91.2	28
	YOLOv11	0.652	93.8	22
40	YOLOv5sM	0.872	91.2	36
	YOLOv8	0.532	95.2	28
	YOLOv11	0.385	97.2	22
50	YOLOv5sM	0.624	94.2	36
	YOLOv8	0.352	97.2	28
	YOLOv11	0.245	98.5	22
60	YOLOv5sM	0.452	96.2	36
	YOLOv8	0.248	98.2	28
	YOLOv11	0.168	99.2	22
70	YOLOv5sM	0.328	97.5	36
	YOLOv8	0.182	98.8	28
	YOLOv11	0.118	99.5	22
80	YOLOv5sM	0.245	98.2	36
	YOLOv8	0.132	99.2	28
	YOLOv11	0.085	99.8	22
90	YOLOv5sM	0.188	98.8	36
	YOLOv8	0.095	99.5	28
	YOLOv11	0.062	99.9	22
100	YOLOv5sM	0.154	99.2	36
	YOLOv8	0.072	99.7	28
	YOLOv11	<b>0.045</b>	<b>99.9</b>	<b>22</b>

Source: (Research Results, 2025)



Table 2. Evaluation Results During Validation

Epoch	Model	Val Loss	Val mAP@0.5 (%)	Val mAP@0.5: 0.95 (%)
10	YOLOv5sM	3.012	59.8	41.2
	YOLOv8	2.324	69.2	52.3
	YOLOv11	1.952	78.5	63.5
20	YOLOv5sM	2.105	76.5	61.2
	YOLOv8	1.532	83.1	71.2
	YOLOv11	1.124	88.5	79.8
30	YOLOv5sM	1.372	84.5	73.2
	YOLOv8	0.942	90.2	82.3
	YOLOv11	0.712	93.2	87.5
40	YOLOv5sM	0.953	90.2	81.5
	YOLOv8	0.598	94.8	89.2
	YOLOv11	0.432	96.8	92.5
50	YOLOv5sM	0.712	93.5	86.5
	YOLOv8	0.412	96.8	92.8
	YOLOv11	0.285	98.2	95.2
60	YOLOv5sM	0.532	95.5	89.5
	YOLOv8	0.302	97.8	94.2
	YOLOv11	0.198	98.8	96.8
70	YOLOv5sM	0.412	96.8	92.2
	YOLOv8	0.228	98.5	95.8
	YOLOv11	0.142	99.2	97.8
80	YOLOv5sM	0.312	97.8	94.2
	YOLOv8	0.168	98.8	96.8
	YOLOv11	0.105	99.5	98.5
90	YOLOv5sM	0.245	98.5	95.2
	YOLOv8	0.122	99.2	97.5
	YOLOv11	0.078	99.7	99.2
100	YOLOv5sM	0.202	98.8	96.2
	YOLOv8	0.092	99.5	98.2
	YOLOv11	<b>0.058</b>	<b>99.8</b>	<b>99.5</b>

Source: (Research Results, 2025)

### Evaluation Metrics

Figures 2, 3, and 4 show the confusion matrix analysis, which confirms that all three YOLO models, YOLOv5sM, YOLOv8, and YOLOv11, accurately classify “Daging\_Sapi\_Segar” and “Daging\_Sapi\_Tidak\_Segar” images with consistent performance. Each model achieves identical True Positive (TP) values, with 380 for fresh beef and 423 for non-fresh beef, without producing any False Positives (FP) or False Negatives (FN). This result indicates that the models detect the target classes with high precision and supports earlier quantitative findings showing mAP values of up to 99.9%. However, this seemingly perfect result deserves further clarification. While the confusion matrices show 100% True Positives with no False Positives or False Negatives, the mAP@0.5 remains slightly below 100% (at 99.5%). This is not a contradiction, but rather a reflection of how the metrics differ. The confusion matrix reflects classification accuracy for successfully detected objects, while the mAP metric considers both classification and localization quality across varying confidence and IoU thresholds. Thus, minor

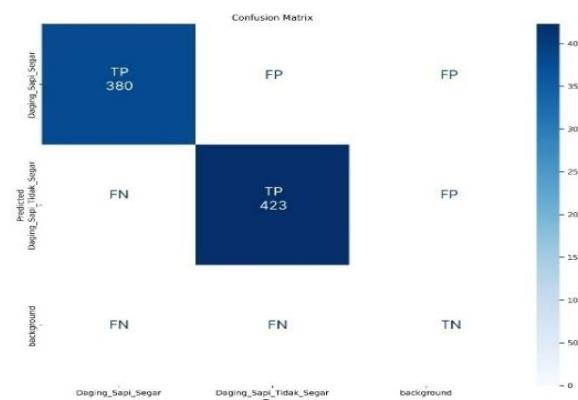
localization errors or confidence threshold effects can reduce mAP without appearing as errors in the confusion matrix.

This nearly perfect detection performance suggests that the models not only provide high classification accuracy but also maintain stable feature recognition across the test data. However, since the test dataset exhibits relatively uniform and distinguishable visual patterns, further evaluation on more complex and varied datasets is necessary. This step aims to assess the generalization ability of each YOLO model in real-world scenarios where object characteristics may vary significantly. This reinforces a critical limitation of our current study: the models were evaluated on a relatively uniform dataset with minimal variation in lighting, texture, and background. Such homogeneity likely made the classification task easier than it would be in practical applications. As a result, the high performance seen in both the confusion matrices and mAP scores must be interpreted with caution. These findings demonstrate what is achievable under controlled conditions, not necessarily what would be expected in real-world deployments. Future studies should incorporate more complex and diverse datasets to assess the true robustness of the models.



Source: (Research Results, 2025)

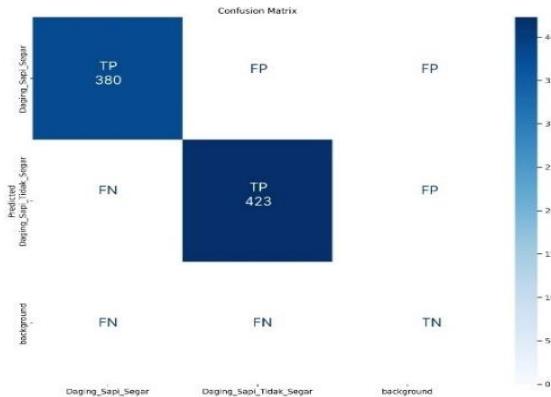
Figure 2. Confusion Matrix of Model YOLOv5sM



Source: (Research Results, 2025)

Figure 3. Confusion Matrix of Model YOLOv8





Source: (Research Results, 2025)

Figure 4. Confusion Matrix of Model YOLOv11

### Classification Report

Table 3 presents the key evaluation metrics for each model on the test data. In general, all three models were able to achieve very high detection performance. It can be seen that the precision and Recall of the three models both reach almost 100%, which means that the models successfully detect all beef objects with very few (false positive and false negative are almost zero). The mAP50 metric (Mean Average Precision at IoU 0.5) is also very high for all models (99.5%) indicating that at a loose IoU threshold, all three models can detect and localize objects accurately. New performance differences are seen in the mAP@0.5:0.95 metric (average mAP at IoU 0.5 to 0.95), where YOLOv5SM is around 96.2%, while YOLOv8 is slightly higher at 96.9%, and YOLOv11 is highest with 97.0%. This indicates that YOLOv11 has a slightly better ability to localize objects with high precision (tighter IoU) than the other two models, although the difference is slight.

Table 3. Classification Report

Model	Evaluation Metrics (%)				Time (Second)
	Precess ion	Rec all	mAP@ 0.5	mAP@0.5: 0.95	
YOLOv 5sM	100	100	99.5	96.2	1358
YOLOv 8	100	100	99.5	96.9	3272
YOLOv 11	100	100	99.5	97.0	3841

Source: (Research Results, 2025)

Table 3 shows that YOLOv5SM is able to achieve excellent detection performance (100% Precision and Recall). While the models achieved perfect precision and recall on the test set, this may reflect overfitting due to the dataset's homogeneity and limited variation in meat parts. Because the training and testing data were collected under controlled conditions rather than diverse real-

world environments such as meat markets or slaughterhouses, there is a risk that performance may degrade when faced with more complex visual scenarios. For example, models trained on a single homogeneous dataset often demonstrate overfitting and poor generalization when applied to new conditions [58]. Similarly, models developed solely on "clean" or limited-variation data have been observed to suffer performance drops in unconstrained, real-world environments [59]. This suggests that our near-perfect results on the test set might not fully translate to other contexts beyond this controlled dataset. Indeed, recent studies emphasize that a small, uniform dataset cannot capture the breadth of real-world variability, potentially introducing hidden biases and recommend using larger, more diverse data to properly assess and improve model generalizability [60]. The use of data augmentation techniques (Flip, Rotation, Mosaic) during training helps YOLOv5SM to recognize objects under various conditions, as evidenced by the high recall (the model does not miss objects) and precision (almost no detection errors). YOLOv5SM's mAP0.5:0.95 value of 96.2% shows that there is a slight drop in performance at stricter IoU thresholds, but the overall accuracy remains high. This performance indicates that although YOLOv5SM is an older architecture, with proper augmentation, it is still very competent for this two-class object detection task.

The YOLOv8 model also showed superior evaluation results. The precision and recall of YOLOv8 reached 100%, which means that every piece of meat in the test data was detected correctly without error. Even without data augmentation, YOLOv8 achieved an mAP@0.5 of 99.5%, equivalent to YOLOv5SM, and a slightly higher mAP@0.5:0.95 (96.9%). This reflects the advantages of YOLOv8's newer architecture: the anchor-free detection head and enhanced backbone model enable it to better capture object features so that performance remains high even when training data is not artificially augmented. Compared to YOLOv5SM, YOLOv8 has a slight edge in terms of high-precision localization capabilities. In other words, YOLOv8 is able to provide a slightly more precise bounding box at high IoU levels, although the difference is marginal.

As for YOLOv11, this model came out as the best in terms of accuracy among the three. The evaluation results of YOLOv11 show 100% Precision and Recall (same as the other two models for overall detection), mAP@0.5 of 99.5%, and the highest mAP@0.5:0.95 of 97.0%. The highest mAP50-95 metric indicates that YOLOv11 is best at detecting objects with precise localization at



various IoU threshold levels. This achievement was obtained without data augmentation, signifying the design advantage of YOLOv11. As the latest generation model, YOLOv11 has likely integrated various updates (e.g. improvements to the backbone component, neck, or training algorithms such as NMS-free or certain attention mechanisms) that provide a slight improvement in accuracy over its predecessors. Although the accuracy difference with YOLOv8 is very small, YOLOv11's consistency in maintaining high performance across various metrics confirms its reliability.

It is also important to clarify the slight discrepancy between the mAP@0.5 values reported for training and validation (99.9% and 99.8%, respectively) and the 99.5% values shown in Table 3 for the test set. This difference is expected and results from evaluating on different data partitions. While training and validation sets are used during model development and tuning, the test set represents previously unseen data. The minor drop of approximately 0.3–0.4% in test performance reflects a small and normal generalization gap, indicating that the models maintain robust predictive ability even on unseen samples. Therefore, this difference is not inconsistent but demonstrates that the models generalize well beyond their training conditions.



Source: (Research Results, 2025)

Figure 5. Detection results of fresh beef by the YOLOv5sM model

Figure 5 provides a visual example of YOLOv5sM detecting a fresh beef sample, supporting the quantitative evaluation with 100% precision and recall. The model accurately identifies a fresh beef cut, as shown by the highlighted bounding box labeled 'fresh beef'. This qualitative evidence confirms the quantitative results 100% precision and recall on the test set, with an mAP@0.5:0.95 of 96.2%. Even with its simpler architecture and targeted data augmentation, YOLOv5sM provides tight bounding boxes

consistent with its high evaluation scores, reinforcing the confusion matrix findings.



Source: (Research Results, 2025)

Figure 6. Detection results of fresh beef by the YOLOv8 model

As shown in Figure 6, YOLOv8 also successfully detects fresh beef with slightly improved localization accuracy compared to YOLOv5sM. The model correctly identifies the fresh beef sample with no false negatives, in line with its perfect recall. YOLOv8 achieved 100% precision and recall, with a slightly higher mAP@0.5:0.95 (96.9%) than YOLOv5sM, indicating marginally improved localization accuracy. The bounding box closely encloses the meat, supporting the precision/recall metrics and validating the confusion matrix results.



Source: (Research Results, 2025)

Figure 7. Detection results of fresh beef by the YOLOv11 model

Figure 7 highlights YOLOv11's ability to detect fresh beef with the highest localization accuracy among the evaluated models. The model successfully detects the beef with perfect classification, achieving the highest mAP@0.5:0.95 (97.0%) among the models. The bounding box tightly matches the beef's contours, visually confirming YOLOv11's superior localization performance. Together with Figures 5 and 6, this reinforces that all models reliably detect fresh beef with no missed detections or false positives.





Source: (Research Results, 2025)

Figure 8. Detection results of non-fresh beef by the YOLOv5sM model

Figure 8 demonstrates YOLOv5sM's detection capability for non-fresh beef, consistent with its perfect confusion matrix performance. The model correctly identifies the non-fresh meat sample, consistent with its perfect confusion matrix results for both classes. Despite being trained on augmented data, YOLOv5sM maintains 100% precision and recall for non-fresh beef, producing tight bounding boxes that match its reported mAP@0.5:0.95 (96.2%). This shows its robustness across both fresh and non-fresh categories.



Source: (Research Results, 2025)

Figure 9 Detection results of non-fresh beef by the YOLOv8 model

As depicted in Figure 9, YOLOv8 effectively detects non-fresh beef, confirming its robustness across both classes. The detection overlay confirms accurate identification of all non-fresh meat portions with no false negatives. This supports the model's perfect precision and recall and its high mAP@0.5:0.95 (96.9%), demonstrating precise localization. The bounding boxes further validate YOLOv8's ability to distinguish non-fresh beef reliably across all test samples.



Source: (Research Results, 2025)

Figure 10. Detection results of non-fresh beef by the YOLOv11 model

Finally, Figure 10 illustrates YOLOv11 detecting non-fresh beef with the highest overall localization precision, further validating the quantitative results. The model detects the large non-fresh beef cut with perfect accuracy, achieving the highest mAP@0.5:0.95 (97.0%) among all models. The bounding box closely matches the meat's outline, confirming YOLOv11's exceptional localization ability. Figures 8–10 collectively show that all three YOLO models consistently detect both fresh and non-fresh beef with zero detection errors, supporting the quantitative results.

#### Comparison of Resource Training

Additionally, the "Time (Second)" column in Table 3 refers to the total training time required for each model to complete 100 epochs. This is not the inference time per image but the full cumulative training duration. Table 4 presents the same data in a more reader-friendly format (hours:minutes), illustrating, for example, that 13,585 seconds for YOLOv5sM is approximately equivalent to 3 hours and 46 minutes. By providing both formats, we aim to maintain clarity and consistency across the manuscript, ensuring that readers understand these metrics represent full model training time under consistent conditions.

The evaluation of model performance is not only based on accuracy, but also on the efficiency of each model. Table 4 summarizes the comparison of training time, iteration speed, GPU memory usage, and model file size for YOLOv5SM, YOLOv8, and YOLOv11. This comparison is important to assess the feasibility of the models in real applications that may have limited computing resources.

Table 4. Comparison of Training Performance and Model Complexity

Model	Training Time	Speed (it/s)	GPU Memory (GB)	Model Size (MB)
YOLOv5sM	3 hours 46 minutes	2.59	1.98	14.8
YOLOv8	54 minutes 32 seconds	3.99	1.00	6.2
YOLOv11	1 hour 4 minutes	4.64	1.05	5.4

Source: (Research Results, 2025)

Table 4 shows that YOLOv5SM has the longest training time, which is about 3 hours and 46 minutes for 100 epochs. This long time is due to two factors, namely the YOLOv5SM architecture, which (relative to the YOLOv8/YOLOv11 variants in this experiment) is more complex or larger, and the use



of augmentation (especially mosaic), which adds computational overhead per iteration. The average iteration speed of YOLOv5SM is only 2.59 it/s, the slowest among the three, with the highest GPU memory usage of 1.98 GB. The size of the YOLOv5SM model after training was 14.8 MB, indicating that it has a larger number of parameters and thus takes up the most space. Consequently, in terms of efficiency, YOLOv5SM is less favorable for deployment on systems with memory constraints or real-time inference needs, despite its high accuracy.

The YOLOv8 model stands out in terms of training efficiency. The training time of YOLOv8 is only about 54 minutes for 100 epochs, which is the fastest among the three models. Its iteration speed is 3.99 it/s, higher than YOLOv5SM, indicating that each batch can be processed faster. The GPU memory usage of YOLOv8 is also very low, only 1.0 GB, indicating its lightweight architecture. This is in line with YOLOv8's model size of only 6.2 MB, much smaller than YOLOv5SM, with a more compact model but equivalent accuracy. YOLOv8 offers an optimal trade-off between accuracy and efficiency. It is suitable for scenarios where training time and resources are limited, without having to sacrifice detection accuracy.

YOLOv11 shows an interesting performance as it successfully combines high accuracy with good efficiency. The training time of YOLOv11 is about 1 hour and 4 minutes, slightly longer than YOLOv8, but still much faster than YOLOv5SM. The iteration speed is the highest, reaching 4.64 it/s, which means YOLOv11 processes the fastest *batches* per second. The GPU memory usage is 1.05 GB, only slightly above YOLOv8, and its model size of 5.4 MB is actually the smallest among the three models. This result is significant because while YOLOv11 provides a slight improvement in accuracy (highest mAP@0.5:0.95), the model complexity and resource requirements are the lowest. This small model size is advantageous for deployment in resource-limited environments (e.g., edge or mobile devices) where memory and storage are constrained. Possibly, YOLOv11 is designed with a more efficient architecture (e.g. layer optimization or better label assignment algorithm) so that it can marginally outperform YOLOv8 in accuracy, while still maintaining a small model size and high speed.

### Comparative Analysis

Based on the results of Table 1, Table 2, Table 3, and Table 4, the three tested YOLO models all performed very well for the two-class fresh vs. non-fresh beef detection task. Precision and *recall* reached 100% for all of them, which means that in this dataset, all three models were able to detect

each beef object correctly without error. This suggests that either the test cases are relatively simple or the models are very capable of handling the task. However, this "perfect classification" result should be interpreted with caution. The test dataset is relatively homogeneous, which likely makes the task easier and can lead to overestimation of model performance. In practice, such flawless accuracy may not generalize to more complex or varied data, highlighting a potential gap in generalization. Additionally, no early stopping was used during training; although we evaluated the best validation checkpoint, training for a fixed number of epochs without an early-stop mechanism can risk overtraining beyond the optimal point. The difference in performance is more pronounced in the localization metrics (mAP@50-95) and computational efficiency aspects.

In terms of detection accuracy, YOLOv11 was slightly ahead with the highest mAP@0.5:0.95 (97.0%), followed very closely by YOLOv8 (96.9%), and then YOLOv5SM (96.2%). The difference is less than 1% absolute, which may not be practically significant for the detection output (all of these differences are difficult to see with the naked eye in the detection results). However, this trend consistently shows that newer model generations provide accuracy improvements, albeit small ones. One important finding is that YOLOv5SM with intensive augmentation still slightly lags behind YOLOv8 and YOLOv11 without augmentation. This indicates that the model architecture plays more of a role in improving accuracy than just data augmentation, at least in this scenario and dataset. Augmentation in YOLOv5SM helped the model achieve very high performance, but not enough to surpass the built-in capabilities of the more modern YOLOv8/YOLOv11.

This study compares the computational efficiency of three YOLO models and identifies notable differences. YOLOv5sM achieves high accuracy but requires nearly 4 hours for training, which is significantly longer than YOLOv8 and YOLOv11, which complete training in less than 1.5 hours. The YOLOv5sM model also consumes more memory, as its file size is 2–3 times larger than the others. In contrast, YOLOv11 demonstrates the best performance by combining high detection accuracy, especially with the highest mAP@0.5:0.95, and efficient resource usage. YOLOv8 delivers the fastest training time and offers accuracy that closely matches YOLOv11, making it a strong alternative when time is limited. Overall, this study recommends YOLOv11 for real-world scenarios that require accurate and efficient object detection, especially in resource-constrained environments.



## CONCLUSION

This research has compared the performance of three YOLO models, YOLOv5sM, YOLOv8, and YOLOv11, in detecting meat freshness from digital images. Based on the test results and analysis, all three models achieved very high accuracy in detecting beef freshness, with precision and recall values reaching 100% on the test set. However, it is essential to emphasize that these results were obtained on a relatively uniform test dataset characterized by consistent visual patterns, which may have influenced the performance metrics. As such, further validation on more complex and heterogeneous datasets is necessary to assess the generalizability and robustness of the models in real-world applications. Among the models evaluated, YOLOv11 exhibited the best overall performance, achieving the highest mAP@0.5:0.95 (97%) and superior efficiency compared to the others. However, the performance gain of YOLOv11 over YOLOv8 was marginal (less than 1% mAP difference), indicating that the two models performed almost equally well on this task. This slight advantage may not hold on more complex datasets where more variability could diminish YOLOv11's edge. YOLOv8 demonstrated advantages in training speed and memory usage, whereas YOLOv5sM, although enhanced with data augmentation, did not surpass the performance of the newer models. These comparative experiments consistently indicate that newer YOLO generations (YOLOv8, YOLOv11) provide notable improvements in both detection accuracy and computational efficiency, largely due to advancements in model architecture and learning algorithms. For similar image-based object detection tasks, the adoption of the latest YOLO versions, particularly YOLOv11, is recommended to achieve optimal results. It is important to note that the dataset used in this study was collected under controlled and visually uniform conditions, which may limit the generalisability of these results. Further validation on more heterogeneous and complex datasets is essential before large-scale deployment. Future research should focus on enhancing the YOLOv11 architecture with advanced modules, such as lightweight attention mechanisms, multi-scale feature fusion, or self-supervised learning strategies to further improve training efficiency and generalization capability across diverse environments.

## REFERENCE

- [1] N. W. Smith, Andrew J. Fletcher, Jeremy P. Hill, and Warren C. McNab, "The Role of Meat in the Human Diet: Evolutionary Aspects and Nutritional Value," *Anim. Front.*, vol. 13, no. 2, pp. 11–18, 2023, doi: 10.1093/af/vfac093.
- [2] N. W. S. A. J. F. J. P. H. W. C. McNabb, "Modeling the Contribution of Meat to Global Nutrient Availability," *Front. Nutr.*, vol. 9, 2022, doi: 10.3389/fnut.2022.766796.
- [3] Melissa Kavanaugh, Diana Rodgers, Melissa Kavanaugh, and F. Leroy, "Considering the Nutritional benefits and Health Implications of Red Meat in the Era of Meatless Initiatives," *Front. Nutr.*, vol. 12, 2025, doi: <https://doi.org/10.3389/fnut.2025.152501> 1.
- [4] A. N. Mafe *et al.*, "A Review on Food Spoilage Mechanisms, Food Borne Diseases and Commercial Aspects of Food Preservation and Processing," *Food Chem. Adv.*, vol. 5, 2024, doi: <https://doi.org/10.1016/j.focha.2024.100852>.
- [5] N. J. Killia, O. J. Nyarongi, and G. P. Baaro, "Assessment of Meat Preservation Methods Used by Retailers and the Estimation of Direct Economic Losses Associated with Meat Spoilage in Kenya," *J. Food Saf. Hyg.*, 2024, doi: 10.18502/jfsh.v9i4.14999.
- [6] M. Hasan *et al.*, "Framework for Fish Freshness Detection and Rotten Fish Removal in Bangladesh Using Mask R-CNN Method with Robotic Arm and Fisheye Analysis," *J. Agric. Food Res.*, vol. 16, no. April, p. 101139, 2024, doi: 10.1016/j.jafr.2024.101139.
- [7] B. A. Altmann *et al.*, "Human Perception of Color Differences Using Computer Vision System Measurements of Raw Pork Loin," *Meat Sci.*, vol. 188, 2022, doi: <https://doi.org/10.1016/j.meatsci.2022.108766>.
- [8] L. Fan *et al.*, "Application of Visual Intelligent Labels in the Assessment of Meat Freshness," *Food Chem.*, vol. 460, 2024, doi: <https://doi.org/10.1016/j.foodchem.2024.140562>.
- [9] M. Adil Raja, R. Loughran, and F. Mc Caffery, "A Review of Performance of Recent YOLO Models on Cholecystectomy Tool Detection," *Meas. Digit.*, vol. 2–3, p. 100007, 2025, doi: <https://doi.org/10.1016/j.meadig.2025.10>



- 0007.
- [10] R. Sapkota *et al.*, "YOLO Advances to its Genesis: a Decadal and Comprehensive Review of the You Only Look Once (YOLO) Series," *Artif. Intell. Rev.*, vol. 58, no. 9, p. 274, 2025, doi: 10.1007/s10462-025-11253-3.
- [11] P. Vilcapoma *et al.*, "Comparison of Faster R-CNN, YOLO, and SSD for Third Molar Angle Detection in Dental Panoramic X-rays," *Sensors*, vol. 24, no. 18, 2024, doi: 10.3390/s24186053.
- [12] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.
- [13] A. S. Geetha, "YOLOv4: A Breakthrough in Real-Time Object Detection." 2025. [Online]. Available: <https://arxiv.org/abs/2502.04161>
- [14] D. Sukumarran *et al.*, "An optimised YOLOv4 Deep Learning Model for Efficient Malarial Cell Detection in Thin Blood Smear Images," *Parasit. Vectors*, vol. 17, no. 1, p. 188, 2024, doi: 10.1186/s13071-024-06215-7.
- [15] I. Ahmad *et al.*, "Deep Learning Based Detector YOLOv5 for Identifying Insect Pests," *Appl. Sci.*, vol. 12, no. 19, 2022, doi: 10.3390/app121910167.
- [16] C. Li *et al.*, "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," Sep. 2022, doi: <https://doi.org/10.48550/arXiv.2209.02976>.
- [17] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," *arXiv*, pp. 1–15, Jul. 2022, doi: <https://doi.org/10.48550/arXiv.2207.02696>.
- [18] M. Yaseen, "What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector," *arXiv*, vol. 8, pp. 1–10, Aug. 2024, doi: <https://doi.org/10.48550/arXiv.2408.15857>.
- [19] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," *arXiv*, 2024, doi: 10.1007/978-3-031-72751-1\_1.
- [20] A. Wang *et al.*, "YOLOv10: Real-Time End-to-End Object Detection," *arXiv*, pp. 1–21, May 2024,
- [21] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," *arXiv*, Oct. 2024, doi: <https://doi.org/10.48550/arXiv.2410.17725>.
- [22] E. H. Alkhammash, "A Comparative Analysis of YOLOv9, YOLOv10, YOLOv11 for Smoke and Fire Detection," *Fire*, vol. 8, no. 1, 2025, doi: 10.3390/fire8010026.
- [23] A. Sharma, V. Kumar, and L. Longchamps, "Comparative Performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN Models for Detection of Multiple Weed Species," *Smart Agric. Technol.*, vol. 9, 2024, doi: <https://doi.org/10.1016/j.atech.2024.100648>.
- [24] K. Venkatesan, M. Muthulakshmi, B. Prasanalakshmi, E. Karthickeien, H. Pabbisetty, and R. Syarifah Bahiyah, "Comparative Analysis of Resource-Efficient YOLO Models for Rapid and Accurate Recognition of Intestinal Parasitic Eggs in Stool Microscopy," *Intell. Based Med.*, vol. 11, p. 100212, 2025, doi: 10.1016/j.ibmed.2025.100212.
- [25] P. Azevedo and V. Santos, "Comparative Analysis of Multiple YOLO-Based Target Detectors and Trackers for ADAS in Edge Devices," *Rob. Auton. Syst.*, vol. 171, p. 104558, 2024, doi: 10.1016/j.robot.2023.104558.
- [26] L. V Fortin and O. E. Llantos, "Performance Analysis of YOLO Versions for Real-Time Pothole Detection," *Procedia Comput. Sci.*, vol. 257, pp. 77–84, 2025, doi: 10.1016/j.procs.2025.03.013.
- [27] J. Steinbach, J. Vrba, T. Jirsa, M. Cejnek, and Z. Urbániová, "Comparative Analysis of YOLO-based Models for Vocal Cord Segmentation in Laryngoscopic Images," *Procedia Comput. Sci.*, vol. 246, pp. 4998–5006, 2024, doi: 10.1016/j.procs.2024.09.457.
- [28] A. Yudhana, R. Umar, and S. Saputra, "Fish Freshness Identification Using Machine Learning: Performance Comparison of k-NN and Naïve Bayes Classifier," *Journal of Computing Science and Engineering*, vol. 16, no. 3, pp. 153–164, 2022. doi: 10.5626/JCSE.2022.16.3.153.
- [29] X. Wu *et al.*, "Quality Non-Destructive Sorting of Large Yellow Croaker Based on Image Recognition," *J. Food Eng.*, vol. 383, no.



- July, p. 112227, 2024, doi: 10.1016/j.jfoodeng.2024.112227.
- [30] O. Jarkas *et al.*, "ResNet and Yolov5-Enabled Non-Invasive Meat Identification for High-Accuracy Box Label Verification," *Eng. Appl. Artif. Intell.*, vol. 125, no. March, p. 106679, 2023, doi: 10.1016/j.engappai.2023.106679.
- [31] X. Liu *et al.*, "BCNO Quantum Dots-Based Ratiometric Fluorescence Platform Integrated with Portable Device: Hypoxanthine Sensing for On-Site Assessment of Meat Freshness with Deep Learning," *Chem. Eng. J.*, vol. 480, no. August 2023, 2024, doi: 10.1016/j.cej.2023.147917.
- [32] E. P. Silmina, T. Hardiani, and S. L. Mahfida, "The Effect of the Number of Classes on the Values Resulting from Evaluation Metrics in the YOLOv5 Model," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 15, no. 2, pp. 419–425, 2025, doi: 10.18517/ijaseit.15.2.20495.
- [33] R. Jian, G. Li, X. Jun, and G. Shi, "Nondestructive Freshness Recognition of Chicken Breast Meat Based on Deep Learning," *Sci. Rep.*, vol. 15, no. 1, p. 27538, 2025, doi: 10.1038/s41598-025-13576-1.
- [34] M. Hou, X. Zhong, O. Zheng, Q. Sun, S. Liu, and M. Liu, "Innovation in Seafood Freshness Quality: Non-Destructive Detection of Freshness in Litopenaeus Vannamei Using the YOLO-shrimp model," *Food Chem.*, vol. 463, no. P1, p. 141192, Jan. 2025, doi: 10.1016/j.foodchem.2024.141192.
- [35] W. X. Sheng Gao, Wei Wang, Yuanmeng Lv, Chenghua Chen, "Intelligent Classification and Identification Method for Conger Myriaster Freshness Based on DWG-YOLOv8 Network Model," *Food Bioeng.*, vol. 3, no. 3, p. 269–279, 2024, doi: <https://doi.org/10.1002/fbe2.12097>.
- [36] R. Ye, G. Shao, Q. Gao, H. Zhang, and T. Li, "CR-YOLOv9: Improved YOLOv9 Multi-Stage Strawberry Fruit Maturity Detection Application Integrated with CRNET," *Foods*, vol. 13, no. 16, 2024, doi: 10.3390/foods13162571.
- [37] D. Wan, R. Lu, T. Xu, S. Shen, X. Lang, and Z. Ren, "Random Interpolation Resize: A Free Image Data Augmentation Method for Object Detection in Industry," *Expert Syst. Appl.*, vol. 228, p. 120355, Oct. 2023, doi: 10.1016/j.eswa.2023.120355.
- [38] Z. He, K. Wang, T. Fang, L. Su, R. Chen, and X. Fei, "Comprehensive Performance Evaluation of YOLOv11, YOLOv10, YOLOv9, YOLOv8 and YOLOv5 on Object Detection of Power Equipment," *arXiv*, 2024, doi: <https://doi.org/10.48550/arXiv.2411.18871>.
- [39] T. Islam, M. S. Hafiz, J. R. Jim, M. M. Kabir, and M. F. Mridha, "A Systematic Review of Deep Learning Data Augmentation in Medical Imaging: Recent Advances and Future Research Directions," *Healthc. Anal.*, vol. 5, p. 100340, 2024, doi: <https://doi.org/10.1016/j.health.2024.100340>.
- [40] A. Yudhana, E. P. Silmina, and Sunardi, "Deteksi Kesegaran Daging Sapi Menggunakan Augmentasi Data Mosaic pada Model YOLOv5sM Beef Freshness Detection by Using Mosaic Data Augmentation on YOLOv5sM," *J. Ris. Sains dan Teknol.*, vol. 9, no. 1, pp. 63–71, 2025, doi: <https://doi.org/10.30595/jrst.v9i1.24990>.
- [41] J. Su, X. Yu, X. Wang, Z. Wang, and G. Chao, "Enhanced Transfer Learning with Data Augmentation," *Eng. Appl. Artif. Intell.*, vol. 129, no. August 2023, 2024.
- [42] T. Bahadur, A. Neupane, R. Koech, and K. Walsh, "Detection and Counting of Root-Knot Nematodes Using YOLO Models with Mosaic Augmentation," *Biosens. Bioelectron. X*, vol. 15, p. 100407, 2023, doi: 10.1016/j.biosx.2023.100407.
- [43] A. A. Murat and M. S. Kiran, "A Comprehensive Review on YOLO Versions for Object Detection," *Eng. Sci. Technol. an Int. J.*, vol. 70, p. 102161, 2025, doi: <https://doi.org/10.1016/j.jestch.2025.102161>.
- [44] J. Terven, D.-M. Cordova-Esparza, J.-A. Romero-González, A. Ramírez-Pedraza, and E. A. Chávez-Urbiola, "A Comprehensive Survey of Loss Functions and Metrics in Deep Learning," *Artif. Intell. Rev.*, vol. 58, no. 7, p. 195, 2025, doi: 10.1007/s10462-025-11198-7.
- [45] U. Sirisha, S. P. Praveen, P. N. Srinivasu, P. Barsocchi, and A. K. Bhoi, "Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, pp. 1–29, 2023, doi: 10.1007/s44196-023-00302-w.
- [46] M. L. Ali and Z. Zhang, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection," *Computers*, vol. 13, no. 12, 2024, doi: 10.3390/computers13120336.



- [47] K. V Houde, P. M. Kamble, and R. S. Hegadi, "A Comparative Analysis of IoU-Based Losses with YOLOv8: Fruit Tree Detection from UAV Images," *Procedia Comput. Sci.*, vol. 260, pp. 898–905, 2025, doi: <https://doi.org/10.1016/j.procs.2025.03.272>.
- [48] K. Su, L. Cao, B. Zhao, N. Li, D. Wu, and X. Han, "N-IoU: better IoU-Based Bounding Box Regression Loss for Object Detection," *Neural Comput. Appl.*, vol. 36, no. 6, pp. 3049–3063, 2024, doi: 10.1007/s00521-023-09133-4.
- [49] X. Qian, N. Zhang, and W. Wang, "Smooth GIoU Loss for Oriented Object Detection in Remote Sensing Images," *Remote Sens.*, vol. 15, no. 5, 2023, doi: 10.3390/rs15051259.
- [50] J. Sánchez-Montejo *et al.*, "AxiWorm: a New Tool Using YOLOv5 to Test Antiparasitic Drugs Against *Trichinella spiralis*," *Parasit. Vectors*, vol. 18, no. 1, p. 36, Feb. 2025, doi: 10.1186/s13071-025-06664-8.
- [51] B. Hussein and S. Shareef, "An Empirical Study on the Correlation between Early Stopping Patience and Epochs in Deep Learning," *ITM Web Conf.*, vol. 64, Jul. 2024, doi: 10.1051/itmconf/20246401003.
- [52] B. A. S. Al-rimy, F. Saeed, M. Al-Sarem, A. M. Albarak, and S. N. Qasem, "An Adaptive Early Stopping Technique for DenseNet169-Based Knee Osteoarthritis Detection Model," *Diagnostics*, vol. 13, no. 11, 2023, doi: 10.3390/diagnostics13111903.
- [53] J. Bento, T. Paixão, and A. B. Alvarez, "Performance Evaluation of YOLOv8 , YOLOv9 , YOLOv10 , and YOLOv11 for Stamp Detection in Scanned Documents," *Appl. Sci.*, vol. 15, no. 6, p. 3154, 2025, doi: <https://doi.org/10.3390/app15063154>.
- [54] F. Wang, "Improving YOLOv11 for Marine Water Quality Monitoring and Pollution Source Identification," *Sci. Rep.*, vol. 15, no. 1, p. 21367, 2025, doi: <https://doi.org/10.1038/s41598-025-04842-3>.
- [55] J. Ferreira and P. Cerqueira, "Food Waste Detection in Canteen Plates Using YOLOv11," *Appl. Sci.*, vol. 15, no. 13, 2025, doi: <https://doi.org/10.3390/app15137137>.
- [56] R. Sapkota and M. Karkee, "Comparing YOLOv11 and YOLOv8 for Instance Segmentation of Occluded and Non-Occluded Immature Green Fruits in Complex Orchard Environment," *arXiv*, pp. 1–17, 2025, doi: <https://doi.org/10.48550/arXiv.2410.19869>.
- [57] R. Khanam, T. Asghar, and M. Hussain, "Comparative Performance Evaluation of YOLOv5 , YOLOv8 , and YOLOv11 for Solar Panel Defect Detection," *Solar*, vol. 5, no. 6, pp. 1–25, 2025, doi: <https://doi.org/10.3390/solar5010006>.
- [58] F. Ma *et al.*, "Every Dataset Counts: Scaling up Monocular 3D Object Detection with Joint Datasets Training." 2024. [Online]. Available: <https://arxiv.org/abs/2310.00920>
- [59] D. Wen *et al.*, "RoHOI: Robustness Benchmark for Human-Object Interaction Detection." 2025. [Online]. Available: <https://arxiv.org/abs/2507.09111>
- [60] A. Noor, H. Almukhalfi, A. Souza, and T. H. Noor, "Harnessing YOLOv11 for Enhanced Detection of Typical Autism Spectrum Disorder Behaviors Through Body Movements," *Diagnostics*, vol. 15, no. 14, 2025, doi: 10.3390/diagnostics15141786.

