Multitek Indonesia: Jurnal Ilmiah
Volume: 20 No. 1, July 2026, Hal. 58 - 70
ISSN: 1907-6223 (print)
ISSN: 2579-3497 (Online)
http://journal.umpo.ac.id/index.php/multitek
doi.org/10.24269/mtkind.v20i1.13819

DETECTION AND CLASSIFICATION OF POST-HARVEST QUALITY USING A HYBRID SSD-EFFICIENTNETV2 MODEL BASED ON TRANSFER LEARNING FOR TOMATO FRUIT

Edga Sukma Pratama[1*], **Danar Putra Pamungkas**[1], **Made Ayu Dusea Widyadara**[1], **Tony Yu**[2]
[1] Faculty of Engineering and Computer Science, Informatics Engineering, Universitas Nusantara PGRI Kediri, Kediri City, East Java.
[2] Electrical and Computer Engineering, Rice University, Houston, Texas, United States
*Corresponding Author's Email : **edgapratama**@gmail.com

Received : 4 May 2026 **Revised : 21 May 2026** **Accepted : 29 May 2026**

Abstract

Manual post-harvest sorting of tomato fruit is prone to subjectivity and inconsistency, necessitating an automated quality assessment approach. Single Shot Detector (SSD) and EfficientNetV2 are both included in the Deep Learning architecture for efficient object detection and classification. This research develops a hybrid model that processes SSD data through a single direct detection, making it lighter than other methods, while EfficientNetV2 serves as the backbone model, capable of producing deep features efficiently. The design of the hybrid SSD-EfficientNetV2 model for the automatic detection and classification of tomato fruit quality (Solanum lycopersicum) into two classes, namely Grade A with fresh and marketable fruit conditions and Grade B with damaged or rotten conditions, is expected to replace the manual sorting process, which is prone to inconsistencies. The data was directly collected from the sales centers and local tomato farms in Nganjuk Regency. The obtained data underwent preprocessing, including resizing, normalization, and augmentation in the form of brightness adjustment, contrast, and hue saturation manipulation. The data is divided into 60% training data, 15% validation data, and 25% testing data. The model was trained for 32 epochs using the AdamW optimizer with a learning rate warm-up and cosine decay scheme. The final evaluation resulted in a classification accuracy of 95.12%, a macro F1 Score of 95.11%, and a Mean Average Precision (mAP) of 85.70% with a precision of Grade A at 94.87% and Grade B at 95.35%. The proposed model offers a reliable contribution as a foundation for an artificial intelligence-based sorting system in the post-harvest tomato industry.

Keywords: Deep Learning, Single Shot Detector, EfficientNetV2, Tomato Classification, Object Detection.

INTRODUCTION

Single Shot Detector (SSD) and EfficientNetV2 are two deep learning architectures that have garnered substantial attention in computer vision and object detection research. SSD excels at completing detection and classification simultaneously within a single forward pass, without requiring a computationally intensive region proposal stage, making it highly efficient for real-time detection applications in the agricultural sector [1]. On the other hand, EfficientNetV2 represents the latest advancement in CNN architecture design, capable of extracting rich feature representations with a significantly more efficient number of parameters[2].

Rapid progress in computer vision and deep learning has opened up wide opportunities to develop

automation solutions that surpass the limitations of manual methods across agricultural sectors. Several previous studies have laid the groundwork for this research, including tomato quality classification based on transfer learning with 88% accuracy [3]. Other studies include the application of YOLOv5 for tomato ripeness classification, which achieved 73% accuracy [4], the implementation of EfficientNetV2L into tomato leaf disease detection with 97,22% performance [5], the application of the Single Shot Detector (SSD) baseline, achieving 87% accuracy and mean average precision (mAP) 80% [6] and *SSD-MobileNetV2* which recorded 94% accuracy in a smart conveyor system [1]. A persistent challenge across most of these studies is the trade-off between inference speed and classification precision when handling subtle visual variations in horticultural products[7].

Advances in computer vision and deep learning have given rise to deep learning-based automation for post-harvest horticultural quality sorting[8]. The quality sorting process, which still relies on human visual assessment, can lead to issues such as subjectivity, inter-rater inconsistency, and low throughput [9][4] and has the potential to cause significant economic losses due to misclassification [10][5]. Tomatoes (Solanum lycopersicum) were selected as the research subject because production volume in Indonesia reached 1.14 million tons in 2023 [11]. This situation positions the standardization of post-harvest tomato quality as an issue that can be addressed through an artificial intelligence-based approach [12].

Existing studies reveal that conventional lighweight backbones commonly pired with SSD, such as MobileNet, have been reported to be less capable of focusing the model attention on the discriminative features of target objects, particulary under complex visual conditions such as varying ripeness levels and surface damage patterns in tomatoes [5]. This limitation directly affects detection reliability in real post-harvest sorting scenarios, where subtle visual differences between quality grades are the primary challenge. The combination of SSD single-pass detection efficiency and EfficientNetV2 parameter-efficient deep feature extraction therefore presents a promising direction to overcome these constrains [13].

This study addreses the identified gap by proposing a hybrid SSD-EfficientNetV2 model, where EfficientNetV2-S replaces convientional lighweight backbones to enchance discriminative feature extraction without sacrificing detection speed. Prior studies have not systematically explored the integration of SSD with EfficientNetV2 specifically for post-harvest tomato quality grading, making this combination a novel contribution to the field. This study aims to comprehensively design, train, and evaluate this hybrid model, optimized for classifying tomato quality into two catagories that is Grade A (fresh and suitable for distribution) and Grade B (damaged or rotten). The scope covers to complete modelling pipeline, from field data aquisition an annotation in Nganjuk Regency, preprocessing stages, and training configuration, to multidimensional evaluation using accuracy, precision, recall, F1-Score, mAP and confusion matrix analysis. By utilizing a dataset collected directly from real agricultural environments, this research aims to deliver a practical and scientifically grounded contribution to the development of automated post-harvest sorting technology.

RESEARCH METHOD

This study applies a Development Research framework that organizes the workflow into four sequential and interrelated stages, including data acquisition, preprocessing, architecture modeling, and performance evaluation.

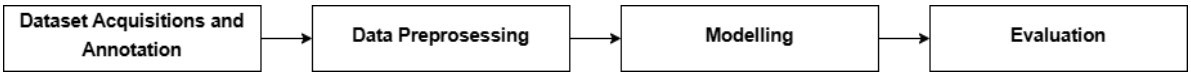

Figure 1. Research workflow

The entire computational process was executed using the Google Colab platform with the PyTorch framework and Python version 3.10, utilizing available GPU acceleration. The stages to be carried out, as shown in Figure 1, include:

2.1 Dataset Acquisition and Annotation

Image data collection was conducted directly at the tomato distribution center located in Gebangkerep Village, Baron district, Nganjuk Regency, East Java. Image capture was performed using two cameras: an iPhone 11 (12 MP sensor resolution, f/1.8 aperture) and an EYD webcam with 1080p resolution. The acquisition yielded 364 images categorized into two main classes. Tomatoes classified as Grade A are characterized by a smooth skin surface with a fresh red or orange color and show no physical damage of any kind, whereas Grade B tomatoes are those that have experienced a decline in quality, such as visible fungal spots or textural changes indicating rot. The annotation process, specifically bounding box annotation, was performed using the labelimg library, with coordinate data saved in XML format following the Pascal VOC standard.

2.2 Data Preprocessing

The pre-processing stage is designed to ensure that the input data has consistent quality and format before being used in model training [14]. This process involves the main operation of resizing the images to 300×300 pixels using bilinear interpolation to meet the input specifications of SSD-EfficientNetV2. Pixel values were subsequently normalized using ImageNet Statistic (mean = [0.485,0.456,0.406]; std = [0.229,0.224,0.225]) to facilitate stable gradient flow and accelerate convergence. The complate dataset of 364 annotated imagaes was partitione into three non overlapping subsets using a stratified split procedure to preserve class distribution across all split 60% for training, 15% for validation, and 25% for testing. A fixed random seed (SEED = 22) was applied troughoutthe splitting process to ensure full reproducibility.

Data augmentation was applied exclusively to the training subset using the Albumentations library, with the following operations is horizontal flipping (probability=0.5), vertical flipping (probability=0.2), random brightness and constrast adjustment (limit = 0.3, probability = 0.5), hue-saturation-value shifts (hue = 15, saturation = 40, value = 20, probaility = 0.5), random gamma correction (range : 80 – 120, probability = 0.3), Gaussian noise injection (variance : 5 – 30, probability = 0.3), Gaussian blur (kernel 3 – 5, probability = 0.2), and shift-scale-rotate transformation (shift = 0.05, scale = 0,1, rotate = 15°, probability = 0.4). Each training image was augmented five times expanding the original 218 training images to 1,090 samples. Validation and test subsets underwent only resizing and normalization with no augmentation applied, to ensure that performance evaluation reflects the model generalization on unmodified real-word data.

2.3 SSD-EfficientNetV2 Model Architecture

The proposed model combines the strengths of EfficientNetV2-S as the backbone and Single Shot Detector (SSD) as the detection head. EfficientNetV2-S was chosen for its efficiency in feature extraction through a combination of Fused-MBConv and standard MBConv blocks [15]. The integration of the Squeeze and Excitation (SE) mechanism and the use of the SiLU activation function

ensure the model can capture complex features with better gradient stability compared to the conventional use of ReLU [16]. On the detection side, this architecture employs a multi-scale approach by extracting features from four different resolutions, ranging from (60×60) to (8×8) pixels. This allows the model to be sensitive to both small and large objects. Each feature level processes localization predictions (bounding boxes) and category classification simultaneously[17]. To support positional accuracy, an anchor generator with seven different scales is implemented, producing 8,732 candidate boxes, thereby accommodating various object shape variations in the input image.

a) SSD Loss Function: MultiBox Loss

SSD training process is governed by the Multibox loss function, which is a weighted combination of two main components this is a localization loss for bounding box regression and a confidence loss for object category classification [18][19]. The equation is formulated as follows [18]:

$$L(x, c, l, g) = \frac{1}{N}\left[L_{conf}(x, c) + \alpha \cdot L_{loc}(x, l, g)\right] \quad (1)$$

Where N is the number of positive anchors (matching the ground truth tomato), (x) is the anchor-to-ground-truth match indicator, (c) is the class confidence prediction, (1) is the bounding box location prediction, (g) is the ground truth bounding box coordinates, and (α) is the balancing factor between the two loss components (default α = 1). The (L_{loc}) component uses Smooth L1 Loss to minimize the difference between the predicted coordinate and the ground truth coordinates of the tomato bounding box, while (L_{conf}) uses Focal Loss to address the imbalance between positive anchors (tomatoes) and negative anchors (background).

b) Intersection over Union (IoU)

Intersection over Union (IoU) is used as the primary criterion in the process of matching anchors with the ground truth tomato bounding box. IoU measures the degree of overlap between the predicted bounding box (A) and the ground truth (B), with the following equation [20]:

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{Area(A \cap B)}{Area(A) + Area(B) - Area(A \cap B)} \quad (2)$$

IoU value ranges from 0 to 1, where 0 means no overlap at all and 1 means the predicted bounding box perfectly matches the ground truth. In this study, an anchor is considered positive (matching the tomato object) if the IoU value is ≥ 0.7 relative to the corresponding ground truth, and is considered negative if IoU < 0.7. This IoU threshold of 0.7 is consistently applied during both training and evaluation using the metric mAP.

2.4 Configuration and Training Strategy

Model training was configured as follows :

1. AdamW optimizer (Adam with weight decay regularization) was applied with a learning rate differentiation between the backbone (lr=0.000005) and the detection head (lr=0.000050). This setting ensures that the pretrained parameters in the backbone do not undergo drastic changes, while providing greater adaptability to the detection head to learn the specific visual patterns of tomatoes.
2. Learning rate scheduling using a combined strategy of linear warmup and cosine decay was chosen to optimize the convergence path. The warmup phase lasts for the first 5 epochs with a gradual increase from zero to the target value. After that, the learning rate is smoothly decreased following a cosine curve until the 32nd epoch.
3. A combination of Smooth L1 Loss for the bounding box regression task and Focal Loss for classification. Focal Loss adaptively increases the weight of hard-to-classify samples (hard

examples) and decreases the weight of easy samples, effectively addressing the imbalance between the number of positive and negative anchors.

4. The hyperparameters used were a batch size of 8 images per iteration and a total duration of 32 epochs. An early stopping mechanism with a patience threshold of 10 epochs based on validation loss was implemented as a safeguard against overfitting.
5. Training was performed on an NVIDIA Tesla T4 GPU via Google Colab using PyTorch version 2.0, with support from the Albumentations library for augmentation, timm for accessing the pretrained EfficientNetV2 model, and torchvision for detection utilities.

2.5 Model Evaluation

Performance evaluation was conducted using a test subset of 82 images separate from the training and validation data. The evaluation metrics applied include classification accuracy, precision, recall, F1-Score (macro and weighted), and the bounding box localization accuracy evaluation using the following formula [21][22]:

1. Accuracy measures the proportion of correct predictions out of the total number of predictions, using the following equation :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

2. Precision measures the proportion of correct positive predictions out of all positive predictions generated by the model:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

3. Recall measures the model's ability to detect all positive samples that actually exist in the data:

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

4. F1-Score is the harmonic mean of precision and recall, providing a balance between the two:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

5. Mean Average Precision (mAP) is the average of the AP values across all object classes, used to evaluate overall object detection performance:

$$mAP = \frac{1}{C}\sum_{c=1}^{C} AP(c) \tag{7}$$

Where TP (True Positive) is the number of correct positive predictions, TN (True Negative) is the number of correct negative predictions, FP (False Positive) is the number of incorrect positive predictions, FN (False Negative) is the number of incorrect negative predictions, C is the number of classes, and AP(c) is the Average Precision for class c.

RESULTS AND DISCUSSION

This section systematically presents and describes the results of the development and implementation of the SSD-EfficientNetV2 architecture in object detection. The presentation covers model performance based on quantitative metrics, visual analysis of prediction results under various image conditions, and an evaluation of the model's ability to accurately distinguish object categories.

3.1 Dataset characteristics and Class Distribution

The analysis of model performance begins with an understanding of the dataset characteristics. A total of 364 images were collected with a balanced distribution between Grade A (163 images) and Grade B (201 images), thereby minimizing the risk of bias without requiring complex class imbalance handling techniques. Controlled lighting conditions during acquisition also ensure that the model can focus on learning discriminative features optimally.

Figure 2. Tomato images characteristics

Visual variations in the dataset include a range of ripeness and texture in Grade A, as seen figure 2 (a) which shows a range of ripeness, and in figure 2 (b) with perfect ripeness, as well as various levels of physical damage and fungal growth in Grade B, as seen in figure 2 (c) where the fruit is damaged by caterpillars and in figure 2 (d) where the tomato is damaged by fungal growth. This class diversity ensures that the model evaluation results are more representative of real-world field conditions compared to the use of synthetic datasets. Nevertheless the relatively small dataset size and single angle image acquisition represent inherent limitations that may affect the model generalization to unseen visual conditions.

3.2 Training and Converegence Analysis

The SSD-EfficientNetV2 model completed the full training process over 32 epochs in approximately 34 minutes using a Tesla T4 GPU. The convergence pattern observed in the training history graph (Figure 3) provides valuable insights into the model's training behavior and stability.

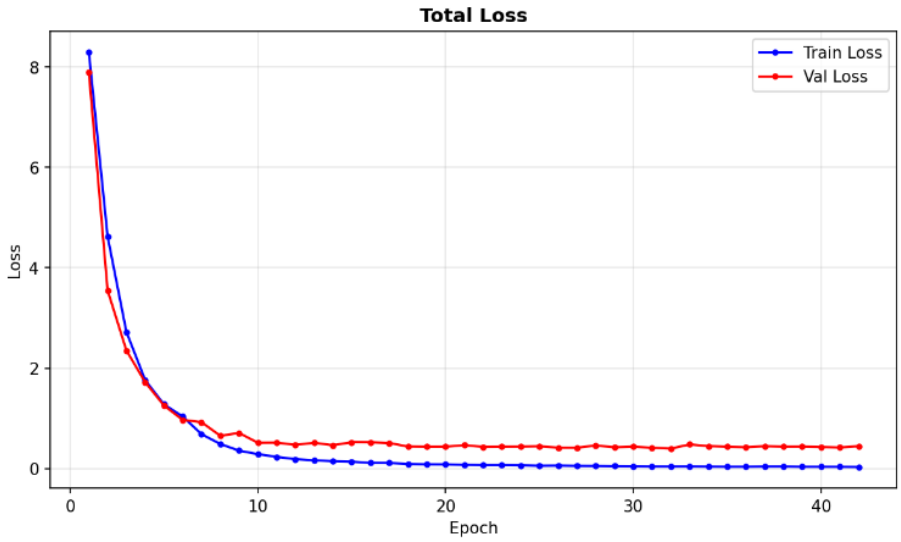

Figure 3. *SSD-EfficientNetV2 Model Training History Total Loss Curve*

As shown in Figure 3 above, during the first 10 epochs, the total loss (combined localization and classification loss) decreased significantly, with the training loss dropping from an initial value of around 2.8 to the 0.6 range. This decrease reflects the effectiveness of the warmup learning rate strategy in guiding the model weights to gradually and efficiently escape from a suboptimal initialization state, without causing gradient vanishing, a common issue in the early stages of deep network training. Model stability becomes evident after the 10th epoch, marked by increasingly smooth and stable changes in loss values. The relationship between the loss metrics on the training data and the validation data serves as the primary parameter for validating the model's reliability. Over the course of 32 epochs, a consistent pattern is observed in both curves, indicating a high level of generalization with no discernible trend differences, suggesting overfitting.

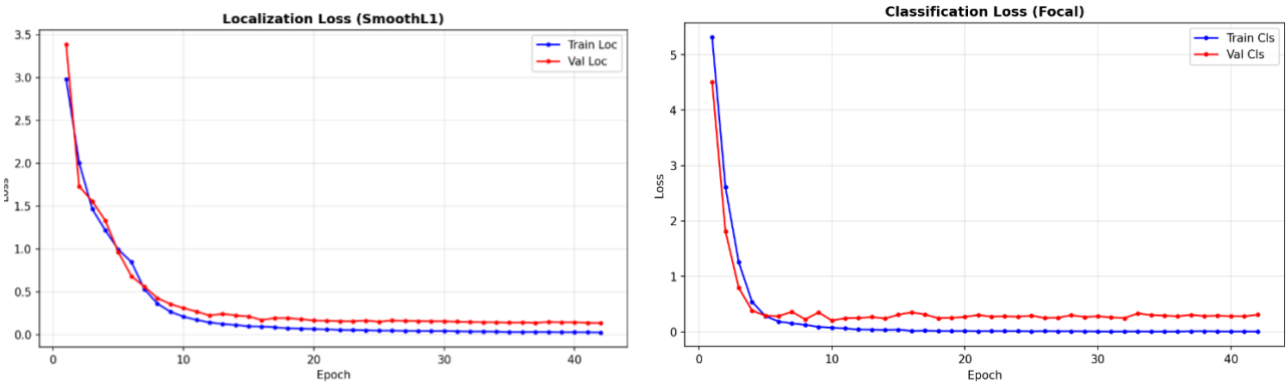

Figure 4. Localization Loss and Classification Loss Curve

The Localization loss (Smooth L1) graph in Figure 4 shows a more stable decline compared to the classification loss. This is understandable given that bounding box coordinates consist of precise numerical values, unlike class probabilities. This is further supported by the fact that the localization loss on the validation data is close to the training loss, indicating the model's ability to accurately place bounding boxes on new data. Initial instability is observed in the classification loss, which uses the focal loss mechanism during the warmup phase. Technically, these fluctuations are triggered by the adaptive nature of the algorithm, which directly adjusts weights based on the confidence level of predictions. This significant variation in loss is a natural response of the model as it seeks an equilibrium point before reaching a stable state capable of consistently accurate predictions.

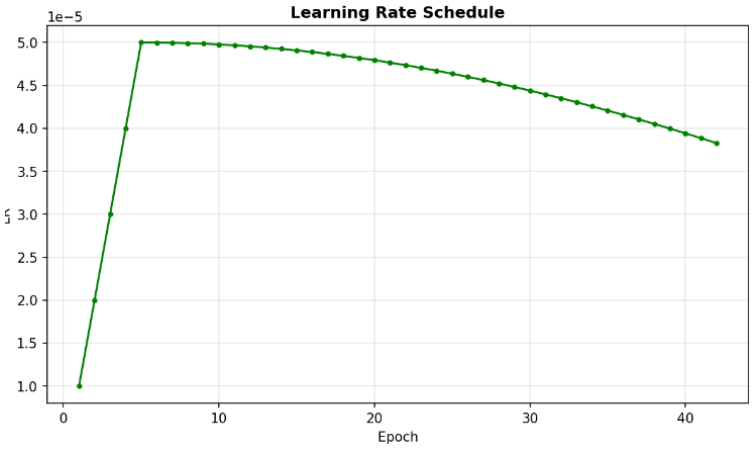

Figure 5. Learning Rate Schedule Curve

In visualization shown in Figure 5 above, the learning rate schedule reflects an optimal configuration that begins with a linear warmup during the first five epochs and continues with a cosine annealing scheme until the end of training. This approach facilitates exploration of the search space, which is useful for avoiding local minima without sacrificing coverage stability. The use of different learning rates, more conservative for the backbone (0,000005) and more progressive for the detection head (0,000050), proved effective in preserving the pretrained ImageNet weights while accelerating model adaptation to local tomato-specific features.

3.3 Detection and Classification Performance Analysis

Evaluation was conducted on 82 test images that were never used during training or validation. Table 1 summarizes the overall evaluation metrics obtained.

Table 1. Summary of Quantitative Evaluation Results for the SSD-EfficientNetV2 Model

Evaluation Metrics	Values
mAP	85,70%
Classification Accuracy	95,12%
F1-Score (Makro)	95,11%
F1-Score (Weighted)	95,12%
Average Precision Grade A	79,79%
Average Precision Grade B	91,60%

Model testing using 82 independent images yielded an mAP of 85.70%. This result indicates that the model effectively integrates detection and classification tasks according to fruit quality classes. At an IoU threshold of 0.7, the predicted bounding boxes were deemed accurate, as they consistently matched the ground truth labels by over 70%. Meanwhile, the classification accuracy reached 95,12%, proving that the visual characteristics distinguishing between classes have been learned very well by the proposed architecture. This reliability is further reinforced by F1-Score values (macro and weighted) that are nearly identical at 95%. This serves as evidence that the model's ability to recognize each class is balanced and not biased toward any single category. An interesting finding was observed in the Average Precision (AP) values, where Grade B (91,60%) was recorded as higher than Grade A (79,79%). This phenomenon is most likely caused by the visual characteristics of Grade B tomatoes (such as rot spots or mold) being more contrasting and easier for the model to recognize. Meanwhile, Grade A tomatoes have a wider range of color and ripeness, creating more complex feature boundaries. This finding suggests that future performance improvements can be achieved by adding specific data variations for the fresh tomato category (Grade A).

Here is a sample calculation based on Equation (2) for a single tomato image sample from the test data to illustrate how the model works in practice. Suppose the model generates a bounding box prediction for a tomato with predicted coordinates (x_1=42, y_1=35, x_2=198, y_2=210) and a ground truth bounding box for Grade A (x_1=50, y_1=40, x_2=205, y_2=218) in pixels on a 300×300 image. The calculation is as follows:

$$wide = min(198, 205) - max(42, 50) = 148\, pixel$$
$$height = min(210, 218) - max(35, 40) = 170\, pixel$$
$$Area(A \cap B) = 148 \times 170 = 25.160\, pixel^2$$

$$Prediction\ area\ =\ 156\ \times\ 175\ =\ 27.300\ pixel^2$$

$$Ground\ truth\ area\ =\ 155\ \times\ 178\ =\ 27.590\ pixel^2$$

$$Area(A\ \cup\ B)\ =\ 27.300\ +\ 27.590\ -\ 25.160\ =\ 29.730\ pixel^2$$

$$IoU\ =\ \frac{25.160}{29.730}\ =\ 0{,}846\ \geq\ 0{,}7\ \rightarrow\ True\ Positive\ (valid\ detection)$$

$$Confidence\ score\ model\ =\ 0{,}84\ (Grade\ A)$$

Based on the above calculations, the model predicts that the tomato is classified as Grade A with a confidence score of 0,84 or 84%.

Here is a sample calculation based on Equation (1) for *MultiBox Loss*, with N = 1 positive anchor and α = 1:

$$L_{loc}\ \approx\ 0{,}042\ (Smooth\ L1)$$

$$L_{conf}\ \approx\ 0{,}073\ (Focal\ Loss)$$

$$L\ =\ \frac{1}{1}\ [0{,}073\ +\ 1\ \times\ 0{,}042]\ =\ 0{,}115$$

Based on the calculation example above, the MultiBox Loss value is 0,115. This low value indicates that the object detection model has adapted very well and has a minimal error rate.

3.4 Confusion Matrix Analysis

The confusion matrix shown in Figure 6 provides a detailed overview of the model classification error patterns at the class level.

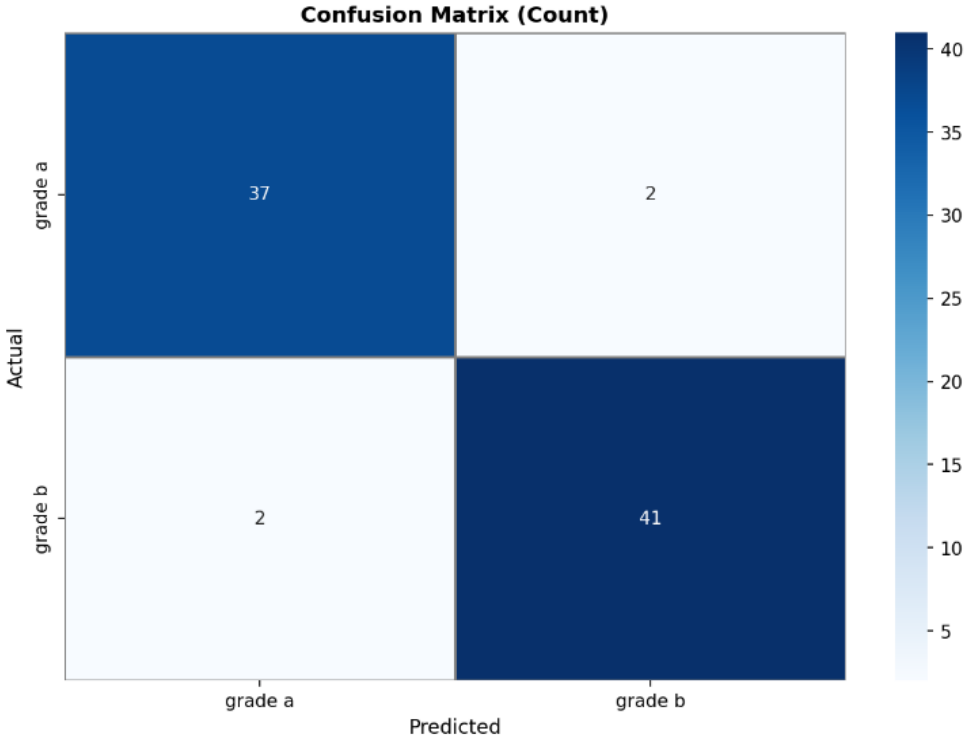

Figure 6. *Confusion Matrix Model SSD-EfficientNetV2*

Based on the test results for 39 Grade A samples, 37 samples (94.87%) were correctly classified, while the remaining 2 samples (5,13%) were incorrectly classified as Grade B. For the Grade B category,

which consists of 43 samples, the model correctly identified 41 samples (95,35%), with only 2 samples (4,65%) incorrectly classified as Grade A.

Analysis of these error patterns provides important operational insights. The two cases of errors in Grade A are attributed to visual ambiguities such as shadow artifacts from uneven lighting or minor surface irregularities that do not reflect actual quality degradation. The two Grade B samples missclassified as a Grade A are suspected to involve damage located outside the camera capture angle, or internal deterioration not yet visibly apparent on the fruit surface consistent with misclassification patterns reported in prior studies involving visual ambiguity[23]. These findings highlight a key dataset limitation is all images were capture from e single angle, which restricts the model ability to detect non-surface or positionally obscured damage. Multi angle image acquisition in future work could meaningfully addres this constraint.

In the post-harvest industry, errors in which damaged tomatoes are classified as fresh produce (Grade A false positive) are considered more risky because they can undermine consumer confidence. However, this error rate of just 4,65% is still considered within the industry's tolerance limits. The low error rates in both classes below 6% also demonstrate that the applied confidence threshold of 0,7 is sufficiently optimal in balancing precision and recall. These results indicate that the proposed system is capable of replacing manual sorting with a level of reliability that meets industrial quality control standarts, making it particulary suited for integration into real-time conveyor based sorting systems in post-harvest.

3.5 Visual Model Prediction Interpretation

The visualization of detection results on test data samples shown in Figure 7 provides an intuitive understanding of the model's working mechanism. Based on visualization results, several key

(a) (b) (c) (d)

Figure 7. Comparasion of Ground Truth vs Model Prediction Results

characteristics of the models' behavior can be identified. As shown in figure 7 (a), this is the ground truth image, followed by the prediction result in figure 7 (b), wich predict that the image is Grade A with a confidence score of 0.99. Similarly, figure 7 (c) shows the ground truth for Grade B, which the system predict as shown in figure 7 (d) with a confidence score of 0.99. The generated bounding boxes demonstrate high precision regarding the shape and position of objects, supported by confidence scores ranging from 0.85 to 0.99, allowing the detection threshold to be tightened without significantly losing valid detections. However, several challenges remain, such as distance between objects, fruits cut off at the edges of the image and lighting interference, which are important considerations for future research on the system.

3.6 Comparison With Similar Studies

To place the contributions of this research in border context, Table 2 presents a systematic comparison with relevant studies that use similar approaches.

Table 2. Comparison of the Proposed Model's Performance with Previous Research

Research	Model/Method	Accuracy	mAP
Aras dkk. (2024) [4]	YOLOv5	73,00%	-
Mputu dkk. (2024) [3]	*Transfer Learning* + ML	88,00%	-
Mustopa dkk. (2024) [5]	*EfficientNetV2L*	97,22%	-
Vitasari dkk. (2025) [1]	*SSD MobileNetV2*	94,00%	-
Adidama dkk. (2025) [6]	*SSD (baseline)*	87,00%	80%
Purposed Model	***SSD-EfficientNetV2***	**95,12%**	**85,70%**

SSD-EfficientNetV2 hybrid model achieved results that were 22,12% better than YOLOv5, as shown in Table 2. This improvement is attributed to the superior backbone of EfficientNetV2, supported by a dataset well-suited to the model. Table 2 shows a 1,12% improvement in the proposed model over the SSD-MobileNetV2 model, proving that the SSD-EfficientNetV2 model can reduce error rate in object classification systems. The proposed EfficientNetV2 variant uses EfficientNetV2-S, which offers efficiency for limited capacity, resulting in a 2,1% diference compared to EfficientNetV2-L as shown in Table 2, due to differences in model capacity and dataset complexity. SSD- EfficientNetV2 model outperforms the SSD baseline model by 8,12%, indicating that integrating SSD with EfficientNetV2 as a backbone can improve model quality, supported by a 5,70% increase in mAP in Table 2, which proves that EfficientNetV2 not only improves true detections but also reduces false detections. Furthermore, the proposed model outperforms the proven effective transfer learning + machine learning method by 7,12%, further reinforcing that the proposed model produces a reliable hybrid method for object detection and classification.

CONCLUSION

The developed Single Shot Detection (SSD)-EfficientNetV2 hybrid model proved capable of distinguishing the quality of post-harvest tomatoes, achieving an accuracy of 95.12% with an mAP value of 85.70% and a macro F1 Score of 95.11%. These results demonstrate that the combination of SSD computational efficiency and the EfficientNetV2 backbone's effective feature extraction capabilities successfully addresses the visual complexity of post-harvest tomatoes. The implementation of this model is expected to improve efficiency and consistency in post-harvest quality management for tomatoes.

REFERENCES

[1] J. Vitasari, R. R. Nugroho, Muhammad Andra Kusuma Ramadhan, Owen Pratama Endramawan, and Mochammad Rifki Ulil Albaab, "Smart Conveyor Real-Time Sort Rotten Tomatoes With Deep Learning Method Integrated IoT Control," *JURNAL ILMIAH RESEARCH AND DEVELOPMENT STUDENT*, vol. 3, no. 1, pp. 242–255, Jan. 2025, doi: 10.59024/jis.v3i1.1135.

[2] M. Tan and Q. V Le, "EfficientNetV2: Smaller Models and Faster Training," *CoRR*, vol. abs/2104.00298, 2021, [Online]. Available: https://arxiv.org/abs/2104.00298

[3] H. S. Mputu, A. Abdel-Mawgood, A. Shimada, and M. S. Sayed, "Tomato Quality Classification Based on Transfer Learning Feature Extraction and Machine Learning Algorithm Classifiers," *IEEE Access*, vol. 12, pp. 8283–8295, 2024, doi: 10.1109/ACCESS.2024.3352745.

[4] S. Aras, P. Tanra, and M. Bazhar, "Deteksi Tingkat Kematangan Buah Tomat Menggunakan YOLOv5," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 2, pp. 623–628, Mar. 2024, doi: 10.57152/malcom.v4i2.1270.

[5] A. Mustopa, U. Khultsum, R. Sabaruddin, R. Ikhsanda, and H. Firmansyah, "Penerapan Model EfficientNetV2L Dalam Mendeteksi Citra Penyakit Daun Tomat untuk Meningkatkan Hasil Panen Petani," *Journal of Information System Research (JOSH)*, vol. 6, no. 1, pp. 100–107, Oct. 2024, doi: 10.47065/josh.v6i1.5886.

[6] M. R. Adidama, R. B. Samudra, W. A. Arrosyid, R. Samsinar, and R. D. Risanty, "Comparison of You Only Look Once (YOLO) and Single Shot Multibox Detector (SSD) Methods for Object Detection Using OpenCV," 2025.

[7] M. Xu *et al.*, "Embracing Limited and Imperfect Data: A Review on Plant Stress Recognition Using Deep Learning," *CoRR*, vol. abs/2305.11533, 2023, doi: 10.48550/ARXIV.2305.11533.

[8] M. Palumbo, M. Cefola, B. Pace, G. Attolico, and G. Colelli, "Computer vision system based on conventional imaging for non-destructively evaluating quality attributes in fresh and packaged fruit and vegetables," Jun. 01, 2023, *Elsevier B.V.* doi: 10.1016/j.postharvbio.2023.112332.

[9] R. Nithya, B. Santhi, R. Manikandan, M. Rahimi, and A. H. Gandomi, "Computer Vision System for Mango Fruit Defect Detection Using Deep Convolutional Neural Network," *Foods*, vol. 11, no. 21, Nov. 2022, doi: 10.3390/foods11213483.

[10] M. Iqbal, D. M. Midyanti, and S. Bahri, "Deteksi Objek Manusia Pada Citra Menggunakan Single Shot Detector (SSD) Berbasis Edge Computing," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 3, pp. 547–556, Jul. 2024, doi: 10.25126/jtiik.938446.

[11] Badan Pusat Statistik Indonesia, "Statistik Indonesia Statistical yearbook of Indonesia," Feb. 2024.

[12] G. Liu, Z. Hou, H. Liu, J. Liu, W. Zhao, and K. Li, "TomatoDet: Anchor-free detector for tomato detection," *Front. Plant Sci.*, vol. Volume 13-2022, 2022, doi: 10.3389/fpls.2022.942875.

[13] M. Tan and Q. V Le, "EfficientNetV2: Smaller Models and Faster Training," 2021. [Online]. Available: https://github.com/google/

[14] R. R. Pratama, "Analisis Model Machine Learning Terhadap Pengenalan Aktifitas Manusia," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 19, no. 2, pp. 302–311, May 2020, doi: 10.30812/matrik.v19i2.688.

[15] X. Ye, T. Ma, and B. Xiao, "Starting from Zero: A No-Pretraining Object Detectors," Jun. 26, 2024. doi: 10.21203/rs.3.rs-4557206/v1.

[16] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3–11, 2018, doi: https://doi.org/10.1016/j.neunet.2017.12.012.

[17] M. Zand, A. Etemad, and M. Greenspan, "Oriented Bounding Boxes for Small and Freely Rotated Objects," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022, doi: 10.1109/tgrs.2021.3076050.

[18] N. Thakur, P. Nagrath, R. Jain, D. S. Saini, N. Sharma, and J. D. Hemanth, "Object Detection in Deep Surveillance," 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:243490911

[19] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," Dec. 2016, doi: 10.1007/978-3-319-46448-0_2.

[20] N. Thakur, P. Nagrath, R. Jain, D. Saini, N. Sharma, and J. Hemanth, "Object Detection in Deep Surveillance," *Res. Sq.*, 2021, doi: 10.21203/rs.3.rs-901583/v1.

[21] J. M. López-Correa, H. Moreno, A. Ribeiro, and D. Andújar, "Intelligent Weed Management Based on Object Detection Neural Networks in Tomato Crops," *Agronomy*, vol. 12, no. 12, 2022, doi: 10.3390/agronomy12122953.

[22] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, p., 2020, doi: 10.1016/j.aci.2018.08.003.

[23] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009, doi: https://doi.org/10.1016/j.ipm.2009.03.002.

[24] D. P. Pamungkas and M. F. Amrulloh, "ANALISIS HASIL KLASIFIKASI PENYAKIT DAUN BAWANG MERAH MENGGUNAKAN CNN ARSITEKTUR EXCEPTION," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 10, no. 1, pp. 359–366, Jan. 2025, doi: 10.29100/jipi.v10i1.5875.