



Department of Digital Business

Journal of Artificial Intelligence and Digital Business (RIGGS)

Homepage: <https://journal.ilmudata.co.id/index.php/RIGGS>

Vol. 5 No. 2 (2026) pp: 17226-17234

P-ISSN: 2963-9298, e-ISSN: 2963-914X

MobileNetV2 Hyperparameter Optimization for Freshness Classification of Tuna Fish Based on Fish Eye Imagery Using Optuna Tree-structured Parzen Estimator (TPE)

Wa Ode Asriyani, Muhammad Ihsan Sarita, Sutardi

^{1,2,3}Department of Informatics, Faculty of Engineering, Halu Oleo University

^{1*}waodeasriyani.e1e122034@gmail.com

Abstract

*Manually assessing the freshness of tuna (*Euthynnus affinis*) within traditional auction markets is frequently subjective, inconsistent, and prone to human error. To address these limitations, this study implemented a robust Deep Learning model utilizing the MobileNetV2 architecture to classify fish freshness levels based on ocular imagery into three distinct categories: Fresh, Less Fresh, and Not Fresh. Although the MobileNetV2 architecture is inherently reliable for image classification, conventional manual hyperparameter tuning often leads to suboptimal model configurations and severe overfitting challenges. As a strategic solution, this study implemented the advanced Tree-structured Parzen Estimator (TPE) algorithm through the Optuna framework to automatically optimize critical hyperparameters, including learning rate, dropout rate, optimizer type, and the fine-tune depth of the convolutional layers. Empirical testing results rigorously proved that the strategic integration of Optuna TPE with the MobileNetV2 model successfully provided a significant performance boost. The optimized model achieved an accuracy of 93.51%, representing a substantial increase of 32.04% compared to the 61.47% accuracy observed in the manual baseline model. Correspondingly impressive improvements were recorded in key performance metrics, specifically achieving a Precision of 93.64%, a Recall of 93.37%, and an F1-Score of 93.49%. In conclusion, this research demonstrates that automated hyperparameter optimization using the Optuna TPE framework significantly enhances the reliability, efficiency, and generalization capability of Deep Learning models, establishing a highly accurate and practical solution for the automated detection of tuna freshness in real-world fishery environments.*

Keywords: Tuna, Fish Freshness, MobileNetV2, Optuna, Transfer Learning, Tree-structured Parzen Estimator (TPE)

1. Background Behind

Mackerel tuna (*Euthynnus affinis*) is one of the most sought-after fishery commodities in Indonesia. Its protein content is almost equivalent to that of tuna, but its price is more affordable, making it a popular choice [1]. However, fish are susceptible to spoilage due to internal factors (enzymes, bacteria, fat oxidation) and external factors (post-catch handling). The freshness of fish significantly affects its quality and economic value [2]. According to the Indonesian National Standard (SNI 01-27291-2006), fish freshness can be identified by the appearance of its eyes. Fish that are not fresh are characterized by sunken eyes, milky-white pupils, and cloudy to yellowish corneas [3]. However, for differentiate freshness itself according to eye If at a glance so will difficult For know it . Then from that it is needed technology that can identify fish freshness based on image eye [4].

Convolutional Neural Networks (CNN) have proven to be very powerful in image classification tasks due to their ability to extract features hierarchically and automatically [5]. MobileNetV2 is a lightweight and efficient CNN architecture designed for resource-constrained devices, but its performance is highly dependent on hyperparameter configuration [6]. Manual hyperparameter tuning often results in suboptimal models, leading to overfitting and convergence stagnation. Hyperparameter optimization can be performed using various methods. Grid search and random search are inefficient for large parameter spaces. Genetic algorithms tend to converge to local optima, while Bayesian optimization with Gaussian processes is computationally intensive. Optuna, a hyperparameter optimization framework based on the Tree-structured Parzen Estimator (TPE), offers a more flexible and efficient approach [7].

Several previous studies have attempted to develop fish freshness detection systems to address the weaknesses of manual identification. Research by Emelia et al . (2023) demonstrated that extracting specific visual features from the eye area of fish is highly valid for use as a freshness determinant [8]. Rong et al, (2021) proved that TPE is

able to improve the accuracy of DNN models [9]. Santosa et al , (2025) confirmed the superiority of MobileNetV2 over other architectures for image classification [10]. However, the integration of Optuna TPE with MobileNetV2 specifically for tuna freshness classification based on eye images has not been explored previously, becoming a research gap that forms the basis of this study.

The objectives of this study are: (1) to build a MobileNetV2-based Transfer Learning model for fresh classification of tuna fish through eye images, (2) to optimize hyperparameters automatically using Optuna TPE, and (3) to analyze the effectiveness of optimization by comparing the performance of the baseline model with the optimized model.

2. Research methods

2.1. Dataset and Data Sharing

A dataset of 2,277 images of tuna fish eyes was split using an 80:10:10 ratio using the splitfolders library, resulting in 1,820 training images, 226 validation images, and 231 test images. Seed = 42 was used to ensure deterministic and reproducible division. The data used in this study are primary data in the form of digital images of tuna fish eyes. Before being used in the computational stage, all raw images have undergone an initial manual pre-processing stage, namely cropping to focus exclusively on the fish eyeball area. This step is crucial to eliminate irrelevant background elements (noise), so that the Deep Learning model can specifically learn feature extraction from the physical condition of the cornea and pupil. Based on the freshness level, the entire dataset is categorized into three main classes, namely "Fresh", "Less Fresh", and "Not Fresh".

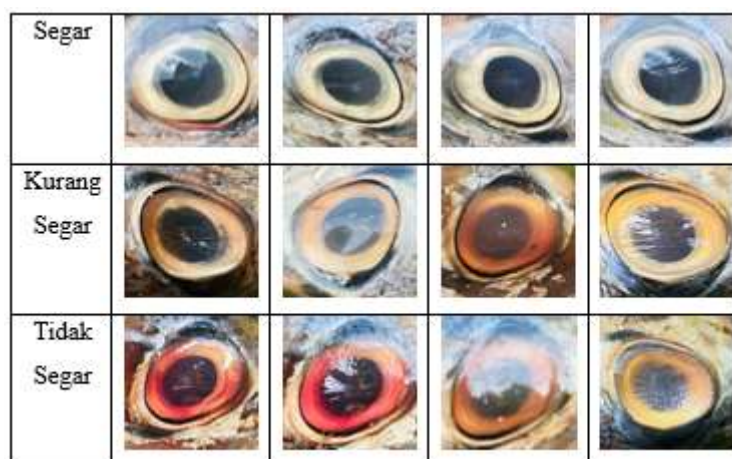


Figure 1 Dataset sample grouped by class

Overall, this study used 2,277 images of tuna fish eyes. To ensure objectivity in the model training and evaluation stages, the dataset was divided into three independent subsets using the split-folders library with a split ratio of 80:10:10. The breakdown of this distribution resulted in 1,820 images as the training set, 226 images as the validation set, and 231 images as the testing set.

2.2. Data Preprocessing and Augmentation

All images were resized to 224×224 pixels as per MobileNetV2 input standard and pixel values were normalized to the range [-1, 1]. On-the-fly data augmentation was applied using: RandomFlip horizontal, RandomRotation (10%), RandomZoom (10%), and RandomBrightness (10%) to prevent overfitting and improve model generalization.

2.3. MobileNetV2 Model Architecture

The MobileNet architecture is a type of Deep Convolutional Neural Network (CNN) with a total of 32 layers specifically designed for low-latency, resource-constrained devices such as smartphones. This compact yet deep design allows the architecture to perform effectively and efficiently under various hardware conditions without overloading the computational memory [11]. MobileNetV2 was chosen as the base model with the configuration include_top=False to remove the default ImageNet classification layer. Pre-trained weights from ImageNet were used to leverage its mature visual feature extraction capabilities. Custom Head layers were added consisting of: GlobalAveragePooling2D, Dropout for regularization, and Dense with Softmax activation function for the three

target classes. A partial fine-tuning strategy was applied by opening a number of final layers automatically determined by Optuna.

2.4. Optuna TPE Hyperparameter Optimization

The Hyperparameter Optimization (HPO) task aims to find the best combination of hyperparameters by searching this space in a way that optimizes a predefined objective function. In supervised learning, this function is usually a loss function, which measures the error between the model's predictions and the actual values [12]. Optuna is a novel hyperparameter optimization framework that uses a tree-structured Bayesian algorithm to model the objective function. This approach offers greater flexibility and efficiency than traditional Bayesian Gaussian optimization processes [7]. In this study, optuna is implemented using a Tree-structured Parzen Estimator (TPE) sampler [13]. TPE is a Bayesian Optimization method prevalent in many open-source software [14]. TPE models the objective function by creating probability distributions for favorable and unfavorable hyperparameter values. This distribution is iteratively refined during optimization, guiding the search towards regions with a higher probability of success [15]. The Optuna framework with the Tree-structured Parzen Estimator (TPE) sampler is used to automatically optimize four hyperparameters:

- (1) Fine-tune Depth: range 0-50 of the last unfrozen layers of MobileNetV2.
- (2) Dropout Rate: range 0.2-0.5 for model regularization.
- (3) Learning Rate: logarithmic scale 10^{-5} to 10^{-2} for convergence speed.
- (4) Optimizer: selection between Adam, SGD, and RMSprop.

The optimization process was run for 35 iterations (trials). TFKerasPruningCallback was used to stop poorly performing iterations early, while HyperbandPruner strategically allocated resources to promising trials. The model was trained for a maximum of 25 epochs per trial with EarlyStopping (patience=3).

2.5. Model Evaluation

Classification model performance evaluation is a crucial step in measuring the model's ability to correctly recognize and differentiate target classes. One method that can be used to measure classification performance is the Confusion Matrix method. This method measures classification performance by comparing actual classification results with those predicted by the system [16]. The evaluation is conducted using four classification metrics: Accuracy, Precision, Recall, and F1-Score. The Confusion Matrix is visualized to analyze the distribution of predictions per class.

1. Accuracy : *Accuracy* is the ratio of correct predictions (positive and negative) to the total amount of data. Formula accuracy shown in Equation 2.7.

$$Accuracy = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n (TP_i + TN_i + FP_i + FN_i)} \quad (2)$$

Where:

- i. i : is an index that represents each class in the classification multiclass
 - ii. n : is the total number of classes in the classification
 - iii. TP_i : (True Positive): Number of correct predictions for the class
 - iv. TN_i : (True Negative): The number of correct predictions that the sample is not from class i when it's not.
 - v. FP_i (False Positive): The number of incorrect predictions that a sample is of class i when it is not.
 - vi. FN_i (False Negative): The number of incorrect predictions that a sample is not of class i when in fact it is [17].
2. Precision : Precision measure level the accuracy of the model in predicting the positive class . The precision value indicates how accurate the model is when states that the image is fresh fish. The formula precision shown in Equation 2.8.

$$Precision \ n_i = \frac{TP_i}{TP_i + FP_i} \quad (3)$$

3. *Recall* (Sensitivity): *Recall* or sensitivity measure the model's ability to retrieve information or detect all the positive data that actually exist in the dataset. Formula *recall* shown in Equation 2.9.

$$Recal \ l_i = \frac{TP_i}{TP_i + FN_i} \quad (4)$$

4. F1- Score : F1 -Score is the harmonic *mean* of mark precision and *recall* . These metrics provide description better performance objective , especially If there is inequality data distribution between classes. Formula *F1-Score* shown in Equation 2.10.

$$F1-Score_i = 2 \times \frac{(Precision_i \times Recall_i)}{(Precision_i + Recall_i)} \quad (5)$$

3. Results and Discussion

3.1. Baseline Model Evaluation

The baseline model implementation phase was conducted to build a basic classification framework as an initial benchmark for system performance. At this stage, the MobileNetV2 architecture was used as a base model, pre-trained on the ImageNet dataset. This allowed the system to leverage the model's advanced visual feature extraction capabilities. The baseline model was built using the MobileNetV2 architecture with manual parameters: a learning rate of 0.0001 (Adam optimizer), a dropout of 0.5, and the last 10 layers unfrozen. The model was trained for a maximum of 25 epochs with EarlyStopping (patience=7).

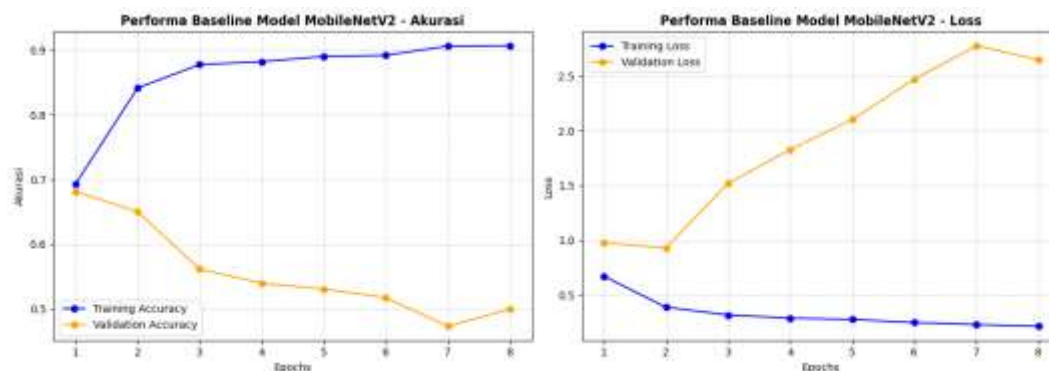


Figure 2 Baseline Model Training Performance Graph

The baseline model training graph shows signs of serious overfitting. Accuracy on training data steadily increased to 85%, but validation accuracy stagnated at around 60%. The validation loss experienced a sharp divergence after the fifth epoch, while the training loss continued to decline toward zero. This phenomenon demonstrates that the baseline model merely "memorized" the training data patterns without being able to generalize.

Table 1. Classification Report Model Baseline

Class	Precision	Recall	F1-Score	Support
Not Fresh	0.45	1.00	0.62	72
Fresh	1.00	0.17	0.29	72
Not Fresh	1.00	0.67	0.80	87
Accuracy			0.61	231

Based on Table 1, the baseline model only achieved an overall accuracy of 61%. The performance gap is clearly visible in the Precision and Recall values. The "Less Fresh" class has a perfect recall (1.00) because the model correctly guessed all "Less Fresh" fish (72 data points), but its precision is very low (0.45) because many fish from other classes were incorrectly guessed into this class. Conversely, for the "Fresh" class, the model has perfect precision (1.00) but very poor recall (0.17), which represents the model's failure to identify most fish that are actually fresh.

3.2. Optuna TPE Optimization Process

This stage is carried out to overcome the weaknesses of the baseline model by automatically finding the most optimal hyperparameter configuration. This study utilizes the Optuna framework integrated with the Tree-structured Parzen Estimator (TPE) and Hyperband Pruner algorithms. This approach allows the system to explore the parameter space extensively and prune unpromising iterations early, thus maintaining computational time efficiency. The optimization process is executed through the objective function (trial) with an allocation of 35 trials. The use of TFKerasPruningCallback ensures that any trial showing poor performance on the validation data is terminated early, while HyperbandPruner strategically allocates more resources to trials with the highest

potential accuracy. Of the 35 search iterations, Optuna TPE successfully identified the best hyperparameter combination on Trial 29 with a validation accuracy of 92.04%. Table 2 presents the 5 best trials from the Optuna search results.

Table 2Top 5 Best Optuna Trials

Trial	Accuracy	Fine-Tune Depth	Dropout Rate	Learning Rate	Optimizer	Status
29	92.04%	3	0.306	0.00376	SGD	FINISHED
4	91.59%	0	0.292	0.00324	SGD	FINISHED
32	90.71%	4	0.321	0.00267	SGD	FINISHED
21	89.38%	3	0.442	0.00053	SGD	FINISHED
17	88.50%	4	0.459	0.00045	SGD	FINISHED

From these results, *trials* with values accuracy highest become reference key in the assembly of the final model. The use of mechanism This automatic optimization has proven to be very effective . compared to manual parameter determination , because capable map connection complex between depth *fine-tuning* and the type of *optimizer* that is most compatible with the image dataset tuna fish eyes .

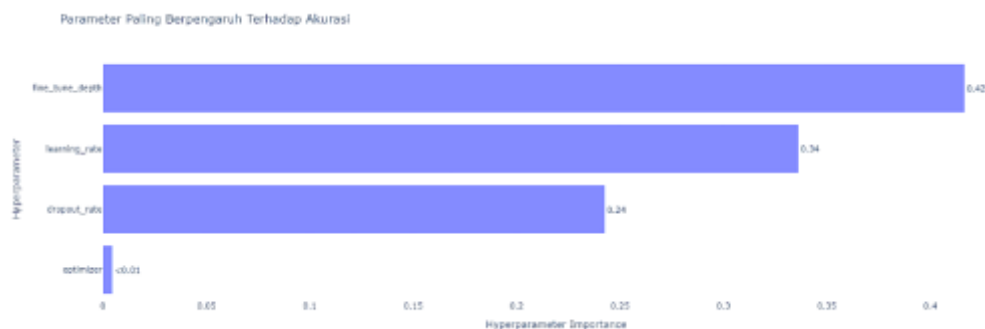


Figure 3Analysis of the Most Influential Parameters on Accuracy

Based on the visualization in Figure 3, it is revealed that the unfreeze layer depth (*fine_tune_depth*) is the most crucial parameter in this research architecture, with an influence level reaching 42%. This empirically proves that the balance between retaining the basic features of ImageNet (early layers) and learning specific fisheye features (late layers) is the main key to the model's success. The second most influential parameter is the *learning_rate* (34%), followed by the *dropout_rate* (24%). Interestingly, the choice of the weight update algorithm (*optimizer*) turned out to have the least significant influence (<0.01%) on the final results. This finding indicates that as long as the learning rate and layer depth range are set optimally, the model can still adapt and achieve good convergence without being too dependent on the type of optimizer used.

3.3. Final Model Evaluation

After the hyperparameter optimization process through the Optuna framework is completed, a set of the best parameter configurations is obtained which provide the highest validated accuracy performance.

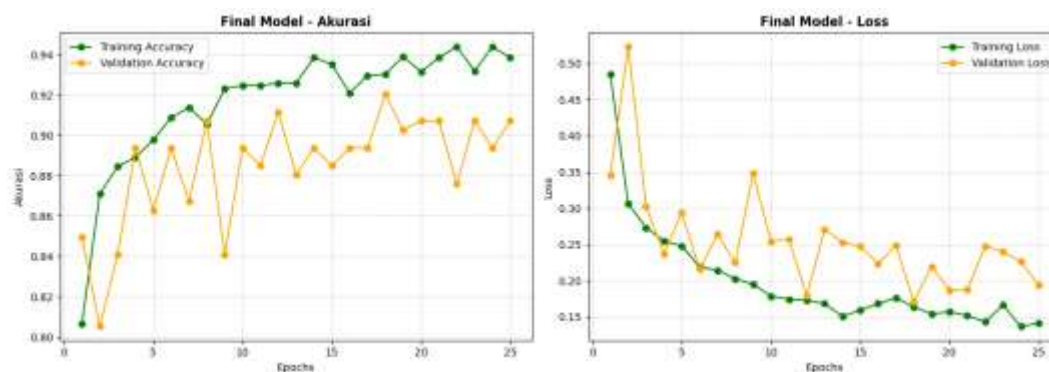


Figure 4Final Model Training Performance Graph Analysis

The final model training graph shows a drastic difference compared to the baseline. The training and validation accuracy curves move in sync, approaching 90%, without a significant gap. The validation loss curve consistently

decreases following the training loss without divergence, confirming that the overfitting issue has been completely addressed.

Table 3. Classification Report Final Model

Class	Precision	Recall	F1-Score	Support
Not Fresh	0.90	0.89	0.90	72
Fresh	0.99	0.96	0.97	72
Not Fresh	0.92	0.95	0.94	87
Accuracy			0.94	231

According to Table 3, the final model recorded a significant performance jump, achieving an overall accuracy of 94%, a significant increase from the baseline model's 61%. More importantly, the performance metrics are now distributed very evenly across classes.

The most impressive performance was demonstrated in the "Fresh" class, where the model achieved a precision of 0.99 and a recall of 0.96 (resulting in an F1-Score of 0.97). This is crucial in the context of real-world system implementation, as it ensures that if the system labels a fish as "Fresh," the guess is correct 99% of the time, and the system almost never misses a fresh fish.

Furthermore, the "Less Fresh" and "Not Fresh" classes also recorded very high F1-Scores (0.90 and 0.94, respectively). This excellent balance of metrics (consistent macro avg and weighted avg at 0.94) provides scientific legitimacy that the combination of hyperparameters optimized by Optuna has successfully addressed the performance imbalance issue and produced a highly reliable model ready for implementation in fish freshness detection.

3.4. Comparative Analysis of Models

The final stage of the evaluation process was a comprehensive performance comparison between the baseline model (MobileNetV2 with manual parameters) and the final model (MobileNetV2 optimized by Optuna). This comparative analysis examined two key dynamics: model behavior during the training (validation) process and final performance metrics during the testing phase. The fundamental differences between the two architectures were already significantly apparent during the learning process. A comparison of accuracy and loss metrics on the validation data is visualized in Figure 5 below.

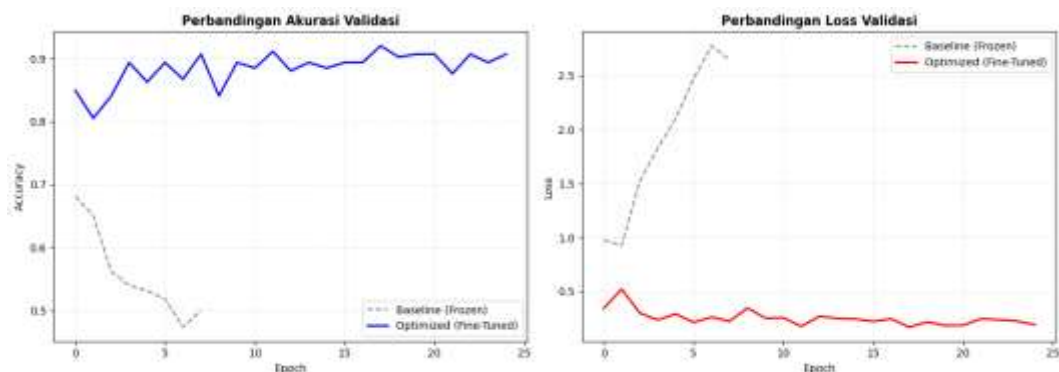


Figure 5 Comparison of Accuracy and Loss of Model Validation

Based on Figure 5, there are two crucial phenomena that can be analyzed:

- 1) Early Stopping and Overfitting Wall Phenomenon: It is clear that the baseline model curve (dotted line) stops much earlier (around the 7th or 8th epoch), while the final model (solid line) is able to continue training until further epochs. This baseline curve stoppage is triggered by the EarlyStopping mechanism that detects that the model has hit an "overfitting wall." This is confirmed by the loss graph (right), where the baseline validation loss value jumps drastically (divergence) to reach 2.5. The baseline model loses its ability to learn and simply memorizes the training data.
- 2) Final Model Convergence Stability: In contrast, the model optimized with Optuna shows a very stable, gentle, and continuously decreasing loss curve (solid red line on the right graph) in the range of 0.2 to 0.5. This stability proves that the combination of learning rate (0.0037) and dropout (0.30) found by Optuna successfully regulates the rhythm of weight updates precisely, preventing the model from basic feature damage (catastrophic forgetting), thus being able to reach a peak validation accuracy of 92.04% (an increase of +23.89% from the baseline of only 68.14%).

The success of stabilization during the validation phase is directly proportional to the model's inference capability when confronted with real test data (unseen data). A comparison of testing metrics is presented in Table 4.

Table 4Comparison of Baseline Model Performance vs. Final Model

Metric	Baseline	Final (Optuna)	Improvement
Accuracy	61.47%	93.51%	+32.04%
Precision	81.57%	93.64%	+12.07%
Recall	61.11%	93.37%	+32.26%
F1-Score	56.79%	93.49%	+36.70%

Based on Table 4, the final model recorded absolute superiority across all measurement indicators. Further analysis shows:

- 1) Class Bias Issue Resolved: The most significant improvement was seen in the F1-Score metric, which jumped sharply by +36.70% (from 56.79% to 93.49%). The F1-Score is the harmonic mean of Precision and Recall. This extreme jump is mathematical proof that the class imbalance prediction issue suffered by the baseline model (where the model predicted almost all fish as "Less Fresh") has been resolved. The model now has even sensitivity across all classes.
- 2) System Reliability: The overall system accuracy increased substantially from 61.47% to 93.51%. This +32.04% improvement transformed the system from being unusable (due to accuracy approaching random guessing) into a highly reliable intelligent system for detecting tuna quality in the field.

Overall, this comparison concludes that optimization intervention using the Tree-structured Parzen Estimator (TPE) algorithm not only artificially increases accuracy but also fundamentally improves the health of the model architecture. Optuna successfully found a parameter space that allows MobileNetV2 to perform optimal fine-tuning, overcome overfitting, and distribute precision evenly across target classes.

3.5. Interface Implementation

This phase aims to bridge user interaction with the Deep Learning model. The web-based interface is designed using HTML and Tailwind CSS for a modern and responsive appearance. The back-end uses the FastAPI framework connected via API, enabling real-time delivery of fisheye images and receipt of classification results from MobileNetV2.



Figure 6Main Page

The homepage features a clean design focused on user convenience. The top section features stage indicators (Upload, Process, Classify, Result) to guide the interaction flow. The middle section provides an interactive upload area (drag-and-drop, file selection, or camera capture), while the bottom section features three information panels on system characteristics: Fast, Accurate, and Secure.

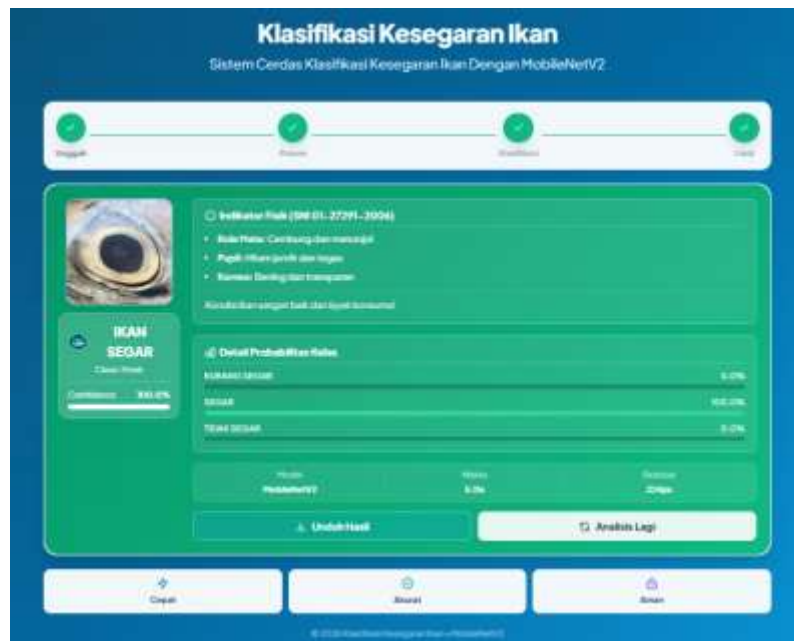


Figure 7 Classification Page

This page automatically displays the results after image processing is complete. The results card will dynamically change color based on the prediction (e.g., bold green for "FRESH FISH"), along with its confidence score. Information presented includes a three-class probability bar graph, physical indicators of fish freshness according to SNI 01-27291-2006, and system metadata. There are also buttons to download the report or repeat the analysis.

4. Conclusion

This study successfully demonstrated the effectiveness of integrating Optuna TPE with the MobileNetV2 Transfer Learning architecture for tuna freshness classification based on eye images. First, the model architecture was successfully designed by modifying MobileNetV2 using Custom Head (GlobalAveragePooling2D, Dropout, Dense-Softmax) and integrated into a Web API-based system using FastAPI. Second, the Optuna TPE algorithm proved highly effective in guiding hyperparameter search, successfully finding the best configuration (fine-tune depth=3, dropout=0.30, learning rate=0.0037, SGD optimizer) that stabilized convergence and overfitting. Third, the Optuna TPE optimization resulted in a surge in accuracy of +32.04% (from 61.47% to 93.51%), with significant improvements in Precision (+12.07%), Recall (+32.26%), and F1-Score (+36.70%). The resulting model was highly reliable and balanced in identifying the three tuna freshness classes.

Reference

- [1] H. Fitriyah, D. Syaquy, and FA Susilo, "Detection Freshness of Mackerel tuna (Euthynnus Affinis) Automatically Based on Eye Images Using Binary Similarity," *Journal of Information Technology and Computer Science*, vol. 7, no. 5, pp. 879–886, Oct. 2020, doi : 10.25126/jtiik.2020753839.
- [2] MFN Al Fatich, AI Setyastuti, D. Kresnasari, and S. Sarmin, " Identification of Freshness Level of Mackerel tuna (Euthynnus sp.) at Bumiayu Market , Bumiayu Regency Brebes ," *J. Mar. Res.*, vol. 12, no. 3, pp. 511–518, Jul. 2023, doi : 10.14710/ jmr.v 12i3.40444.
- [3] M. Muchtar, YP Pasrun, R. Rasyid, N. Miftachurohmah, and M. Mardawati, " Application of the Naïve Bayes Method in Classification Freshness of Fish Based on Color in Eye Area Images," *Journal of Informatics and Electrical Engineering Applied*, vol. 12, no. 1, Jan. 2024, doi : 10.23960/ jitet.v 12i1.3879.
- [4] VT Deco, A. Nilogiri, and Q. A'yun, " Classification The Freshness of Lemuru Fish Based on Eye Images Using Convolutional Neural Network Method . ," *Journal of Information Systems and Electronics Applications*, vol. 3, no. 1, pp. 14–25, 2021.
- [5] IR Alamsyah, SN Budiman, and U. Mawaddah, " Implementation of Deep Learning Using Convolutional Neural Network (CNN) for Classification of Goldfish Types (Carassius Auratus) Based on Characteristics Morphology .," *Infomedia Journal : Informatics Engineering, Multimedia, and Networks*, vol. 10, no. 2, pp. 119–124, 2025.
- [6] MAK Raiaan et al., "A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks," *Decision Analytics Journal*, vol. 11, p. 100470, Jun. 2024, doi : 10.1016/j.dajour.2024.100470.
- [7] J X. Liu, Y. Hu, 15, no. 1, p. 18, Dec. 2024, doi : 10.3390/agronomy15010018.
- [8] WON Emelia, I. Irawati, and N. Kurniati, " Classification Freshness of Lencam Fish (Lethrinus Atkinsoni Seale) Using the K-Nearest Neighbor Method Based on Digital Imagery," *Bulletin of Islamic Information Systems and Technology*, vol. 4, no. 3, pp. 274–285, Aug. 2023, doi : 10.33096/ busiti.v 4i4.1778.

- [9] G. Rong *et al.* , "Comparison of Tree-Structured Parzen Estimator Optimization in Three Typical Neural Network Models for Landslide Susceptibility Assessment," *Remote Sens. (Basel)* , vol. 13, no. 22, p. 4694, Nov. 2021, doi : 10.3390/rs13224694.
- [10] RR Santosa , A. Sasikirana , ZL Ramadhan, YPB Satria, and A. Nugroho, " Performance Comparison of MobileNetV2 and ResNet50V2 in Maturity Level Classification Tomatoes ," *Indonesian Journal of Science, Technology and Humanities* , vol. 3, no. 1, pp. 49–56, Jun. 2025, doi : 10.60076/ ijstech.v 3i1.1335.
- [11] YY Pane and JJ Sihombing , " Bird Species Classification Using the Transfer Learning Method," *Journal of Technology Integrated* , vol. 9, no. 2, pp. 89–94, Dec. 2023, doi : 10.54914/ jtt.v 9i2.744.
- [12] S. Sieradzki and J. Mańdziuk , "Modified Adaptive Tree-Structured Parzen Estimator for Hyperparameter Optimization," Feb. 2025.
- [13] G. Hariharan, R. Mallya, N. Kumar, D. Doreswamy , GM Chennegowda , and SK Bhat, " Optuna -Optimized Ensemble and Neural Network Models for Static Characteristics Prediction of Active Bearings with Geometric Adjustments," *Modeling* , vol. 6, no. 3, p. 98, Sept. 2025, doi : 10.3390/modelling6030098.
- [14] K. Abe, Y. Wang, and S. Watanabe, "Tree-Structured Parzen Estimator Can Solve Black-Box Combinatorial Optimization More Efficiently," Jul. 2025.
- [15] R. Kochnev , A.T. Goodarzi , Z.A. Bentyn , D. Ignatov, and R. Timofte , " Optuna vs Code Llama: Are LLMs a New Paradigm for Hyperparameter Tuning?, " Sep. 2025.
- [16] FM Fathoni , CA Putra, and AL Nurlaili , " Classification of Grape Leaf Diseases Using the K-Nearest Neighbor Method Based on the Gray Level Co-Occurrence Matrix," *Biner: Scientific Journal Informatics and Computers* , vol. 3, no. 1, pp. 8–15, Jan. 2024, doi : 10.32699/biner.v3i1.6332.
- [17] RA Kumala , CA Sari, and EH Rachmawanto , "A Comparison of MobileNetV2 and VGG16 Architectures with Transfer Learning for Multi-Class Image-Based Waste Classification," *Journal of Applied Informatics and Computing* , vol. 9, no. 4, pp. 1610–1624, Aug. 2025, doi : 10.30871/ jaic.v 9i4.9958.