# NutriSave: An Integrated Mobile System for Food Waste Reduction Featuring High-Accuracy Image Classification

Maulana Ahmad Zahiri[*]
*5027231010@student.its.ac.id*

Athalla Barka Fadhil[*]
*5027231018@student.its.ac.id*

Danendra Fidel Khansa[*]
*5027231063@student.its.ac.id*

Harwinda[*]
*5027231079@student.its.ac.id*

[*] *Department of Information Technology, Sepuluh Nopember Institute of Technology, Surabaya, Indonesia*

***Abstract*—Household food waste contributes significantly to global economic losses and environmental degradation. To address this, we present NutriSave, an integrated mobile system for automated food inventory management. The system's novelty lies in a dual-AI backend that couples (1) a state-of-the-art ConvNeXt model for high-accuracy food identification, and (2) a deep neural network for predicting consumption likelihood before expiry. This paper details the system's AI components and provides a rigorous empirical validation of the core image classification engine, which achieves a state-of-the-art test accuracy of 96.38%. We also present a preliminary analysis of the consumption prediction model, which attains an AUC-ROC score of 0.73 on a synthetic dataset. Together, these components validate the technical feasibility of using a multi-faceted AI approach to power a low-friction, high-impact tool for reducing household food waste.**

***Index Terms*—Food Waste, Mobile Computing, Deep Learning, ConvNeXt, Computer Vision, Image Classification, Expiry Tracking.**

## I. Introduction

Food waste is a pressing global issue with profound environmental, economic, and social consequences [1], [2]. A substantial fraction of this waste occurs at the consumer level, where households discard perishable items due to improper inventory management and forgotten purchases [3]. The application of artificial intelligence offers a promising pathway to address such complex societal challenges, aligning with broader goals of sustainable development [4]. While numerous mobile applications exist, they often rely on manual data entry or barcode scanning [5], [6], creating significant user friction that hinders long-term adoption and effectiveness, a well-documented phenomenon in digital health and consumer applications [7].

This paper proposes **NutriSave**, a holistic mobile system designed to minimize user effort by automating the inventory process. The system is architected as a two-tier client-server application where a mobile frontend communicates with a backend API. The backend's novelty lies in its dual-AI engine:

1) A user photographs a food item with the mobile app.
2) The backend's high-accuracy ConvNeXt model classifies the item.
3) This classification, along with other metadata, feeds into a second neural network that predicts the likelihood of the item being used before its typical expiry date.

This workflow enables intelligent, proactive reminders and recipe suggestions for items at high risk of being wasted.

The primary contributions of this work are twofold: (1) we present the system concept and dual-AI approach of NutriSave; and (2) we provide a rigorous development and evaluation of its cornerstone image classification model, achieving a state-of-the-art test accuracy of **96.38%**, while also offering a preliminary validation of the novel consumption prediction classifier.

## II. Related Work

The development of NutriSave's recognition engine is informed by recent advancements in computer vision and deep learning. Early vision-based systems struggled with the high intra-class variance of food items [6], a classic challenge in fine-grained visual categorization (FGVC) [12]. The advent of deep Convolutional Neural Networks (CNNs), particularly landmark architectures like ResNet [13], has revolutionized the field. However, many studies report accuracies that may not be sufficient for a frictionless user experience. For instance, Soedomo and Heryanto [8] achieved 85.43% accuracy for fruit classification, while Sultana and Jahan [9] reached 85% with an InceptionV3 architecture. For deployment on mobile or edge devices, lightweight architectures such as MobileNet [14] and EfficientNet [15] have also been proposed, balancing accuracy with computational cost.

More advanced architectures have pushed performance higher. Wang et al. [11] used a Vision Transformer (ViT) to achieve 90% accuracy. A key challenge remains classifying freshness, where an optimized YOLOv4 model achieved a mean average precision (mAP) of only 50.4% [10], a difficult task also explored by others using spectral imaging [16], highlighting the difficulty of nuanced visual assessment.

Our approach builds on these advancements by leveraging **ConvNeXt** [17], a modern architecture designed to improve upon standard ConvNets by incorporating design principles from Transformers. As summarized in Table I, by employing a meticulous two-phase fine-tuning strategy, our work surpasses these prior benchmarks, demonstrating a state-of-the-art accuracy of **96.38%**. This result confirms that high-precision, automated food identification is a solved problem ready for integration into user-facing mobile systems.

TABLE I
COMPARATIVE ANALYSIS OF RECENT FOOD IMAGE CLASSIFICATION MODELS

| Reference | Year | Model Architecture | Primary Task | Reported Performance |
|---|---|---|---|---|
| Soedomo & Heryanto [8] | 2019 | Deep Learning (CNN) | Fruit Type Classification | 85.43% Accuracy |
| Sultana & Jahan [9] | 2023 | InceptionV3 | Organic Veg. Quality | 85% Accuracy |
| Chen & Wu [10] | 2022 | Optimized YOLOv4 | Freshness Classification | 50.4% (mAP) |
| Wang et al. [11] | 2023 | Vision Transformer (ViT) | Fruit Quality Assessment | 90% Accuracy |
| **This Work** | **2024** | **ConvNeXt Tiny** | **Fruit & Veg. Classification** | **96.38% Accuracy** |

*Note: Performance metrics are not directly comparable as each study utilizes a different dataset and, in some cases, different evaluation tasks.*

## III. HIGH-ACCURACY FOOD CLASSIFICATION

We first validated the cornerstone of the system: the food classification model.

### A. Methodology

The model was trained on the "Fruit and Vegetable Image Recognition" dataset, a public collection of 3,825 images across 36 classes, pre-divided into training (3,115), validation (351), and test (359) sets. We selected the **ConvNeXt Tiny** architecture [17], pre-trained on ImageNet, and implemented a two-phase fine-tuning strategy:

1) **Phase 1 (Head-Only Training):** The model's backbone was frozen, and only the classification head was trained for 5 epochs using the Adam optimizer (LR = $1 \times 10^{-3}$) [18].
2) **Phase 2 (Full Fine-Tuning):** All layers were unfrozen and trained with a low learning rate (LR = $1 \times 10^{-5}$), a Cosine Annealing scheduler to smoothly adjust the learning rate [19], and early stopping.

Aggressive data augmentation, a technique proven essential for generalization in deep models [20], and a simulated batch size of 64 were used during training on two NVIDIA T4 GPUs.

To ensure stable convergence during the full fine-tuning phase, the learning rate $\eta_t$ at epoch $t$ follows the Cosine Annealing schedule defined as:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min}) \left(1 + \cos\left(\frac{T_{cur}}{T_{max}}\pi\right)\right) \quad (1)$$

where $\eta_{max}$ and $\eta_{min}$ are the initial and minimum learning rates, respectively, $T_{cur}$ is the current epoch, and $T_{max}$ is the total number of epochs per cycle.

### B. Results

The training history in Fig. 1 shows the effectiveness of our strategy. The model reached a peak validation accuracy of **96.30%** at epoch 14, and early stopping halted training at epoch 19. The best model was then evaluated on the held-out test set, achieving a final test accuracy of **96.38%** (Table II). This exceptional result confirms the model's robustness.

## IV. CONSUMPTION LIKELIHOOD PREDICTION (NOVELTY COMPONENT)

The novelty of NutriSave lies in its second AI component, which predicts not an exact date, but the likelihood that an item will be consumed before spoiling.

TABLE II
FINAL CLASSIFICATION MODEL PERFORMANCE

| Metric | Value |
|---|---|
| Model Architecture | ConvNeXt Tiny |
| Dataset Split | Test Set (359 images) |
| Best Validation Accuracy | 96.30% |
| **Final Test Accuracy** | **96.38%** |

### A. Methodology

We developed a Deep Neural Network (DNN) for this binary classification task (`used_before_expiry`: 1 or 0). This approach draws parallels to user behavior prediction models seen in other domains [21].

- **Dataset:** We used a synthetic dataset of 500 food items with features such as `purchase_month`, `quantity`, `item_type`, and `storage_type`.
- **Preprocessing:** To handle outliers and class imbalance, we applied `RobustScaler` for feature scaling and computed class weights to penalize errors on the minority class during training.
- **Model:** A sequential DNN with three hidden layers (64, 32, 16 neurons), each regularized with Batch Normalization [22] and Dropout [23], was implemented in TensorFlow/Keras. The model was trained to optimize for the Area Under the Receiver Operating Characteristic Curve (AUC-ROC), a robust metric for imbalanced classification problems [24].

Given the imbalanced nature of the dataset, we optimize the network using a Weighted Binary Cross-Entropy loss function to increase the penalty for misclassifying the minority class (items consumed). The objective function $\mathcal{L}$ is formulated as:

$$\mathcal{L}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}[w_1 \cdot y_i \log(\hat{y}_i) + w_0 \cdot (1 - y_i)\log(1 - \hat{y}_i)] \quad (2)$$

where $N$ is the batch size, $y_i \in \{0, 1\}$ is the ground truth label, $\hat{y}_i$ is the predicted probability, and $w_1, w_0$ are the calculated class weights for the positive and negative classes, respectively.

### B. Preliminary Results

The model was trained and evaluated on stratified train, validation, and test splits. The final performance on the test set is summarized in Table III. The model achieved an AUC-ROC

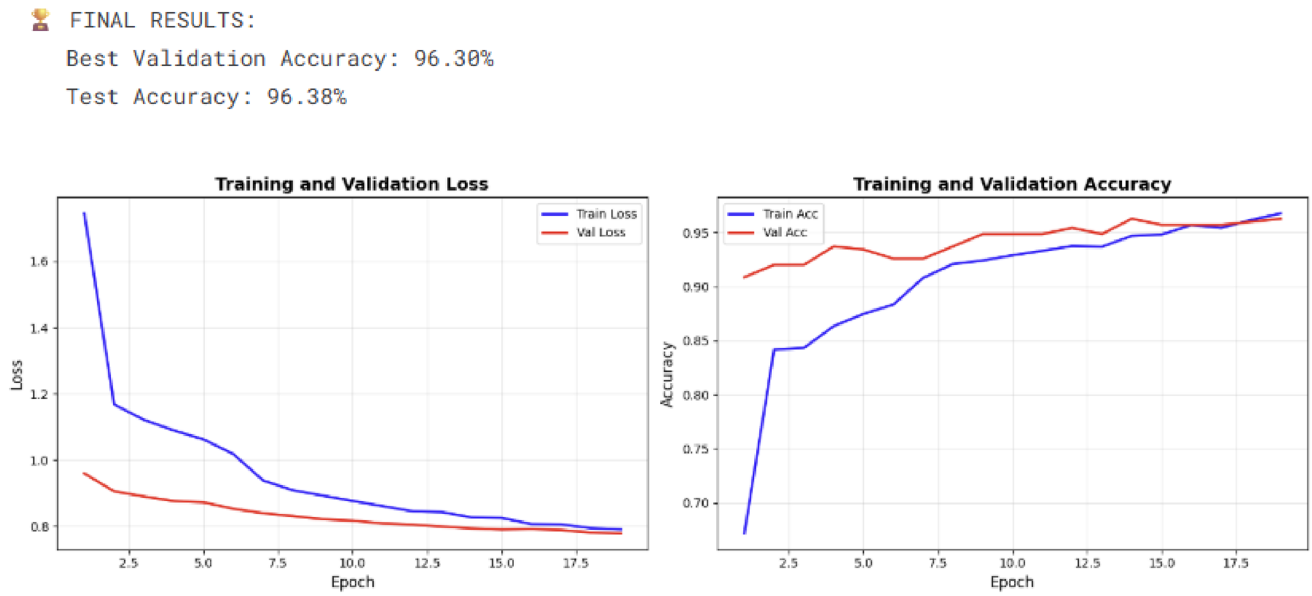


Fig. 1. Training and validation loss (left) and accuracy (right) curves for the classification model over 19 epochs. The model achieves its best performance around epoch 14 before training is halted by early stopping.

score of **0.7302**, demonstrating a predictive capability significantly better than random chance. The high recall (86.11%) indicates that the model is effective at identifying items that are likely to be consumed, which is crucial for a waste-reduction system.

TABLE III
PERFORMANCE OF THE CONSUMPTION PREDICTION CLASSIFIER (TEST SET)

| **Metric** | **Value** |
|---|---|
| Accuracy | 68.42% |
| Precision | 63.27% |
| Recall | 86.11% |
| F1-Score | 73.08% |
| **AUC-ROC** | **0.7302** |

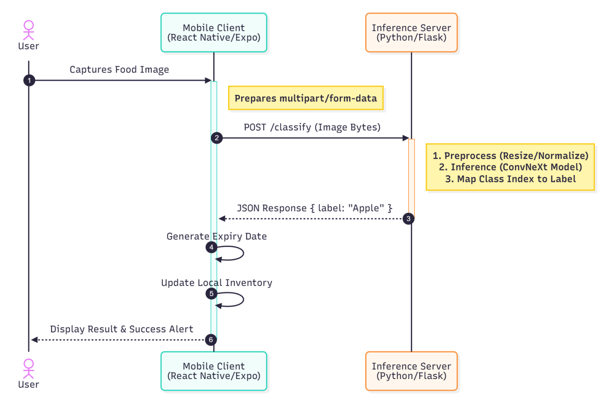


Fig. 2. System architecture diagram illustrating the data flow between the React Native client and the Flask inference backend.

## V. SYSTEM ARCHITECTURE AND IMPLEMENTATION

To ensure practical viability and low-friction user adoption, NutriSave is architected as a decoupled client-server system. This design separates the resource-intensive inference tasks from the user-facing mobile interface.

### A. High-Level Architecture

The system operates on a RESTful microservice pattern. The architecture consists of two primary components:

- **Mobile Client:** A cross-platform application developed using React Native and the Expo framework, responsible for image capture, user interaction, and local inventory management.
- **Inference Server:** A Python-based backend utilizing Flask, which hosts the deep learning models and exposes endpoints for image classification.

Communication between the client and server is handled via HTTP/1.1 POST requests carrying multipart/form-data, ensuring compatibility with standard network protocols. Furthermore, the system is designed to preserve user privacy; images transmitted to the backend are processed strictly in-memory using byte streams and are not permanently stored on the server's file system.

### B. Mobile Application (Frontend)

The frontend is built with React Native to target both Android and iOS platforms from a single codebase. Key implementation details include:

- **Image Capture:** We leverage the device's native camera modules to capture high-resolution images. To optimize network latency, images are downscaled to $512 \times 512$ pixels and compressed into JPEG format before transmission.
- **User Flow:** The user initiates the "Scan" flow, capturing a food item. The app displays a loading state while asynchronously awaiting the API response. To ensure system robustness, the client implements error handling for HTTP 4xx and 5xx response codes, prompting the user to retry the operation in the event of transient network instability. Upon receiving the label (e.g., "Tomato"), the user confirms the item, which is then added to the local SQLite database with an automatically calculated expiry risk.

### C. Backend Inference Engine

The backend is powered by a Flask server running on a GPU-accelerated environment. The implementation prioritizes modularity:

1) **Preprocessing Pipeline:** Incoming image bytes are converted to RGB format and normalized using ImageNet statistics ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$) to match the ConvNeXt input requirements.
2) **Inference:** The system loads the fine-tuned ConvNeXt Tiny model using the `timm` library. The inference logic runs in a `torch.no_grad()` context to minimize memory overhead.
3) **API Response:** The server returns a JSON payload containing the predicted class label, which the mobile app parses to update the UI. The response adheres to a strict JSON schema (e.g., `"label": "Apple"`), which allows the mobile client to deterministically map the classification result to the local inventory database.

Incoming image bytes are converted to RGB tensors and normalized channel-wise. For a given input pixel $x$, the normalized value $x'$ is computed via:

$$x'_c = \frac{x_c - \mu_c}{\sigma_c}, \quad c \in \{R, G, B\} \tag{3}$$

where $\mu = [0.485, 0.456, 0.406]$ and $\sigma = [0.229, 0.224, 0.225]$ represent the channel-wise mean and standard deviation of the ImageNet dataset, respectively. Following normalization, the image is explicitly resized and reshaped into a tensor of dimensions *(1,3,256,256)* to strictly align with the input layer requirements of the ConvNeXt architecture.

## VI. Discussion

The empirical results and architectural design of NutriSave validate the feasibility of an automated food waste reduction system.

### A. Performance Analysis

The 96.38% accuracy of the ConvNeXt classifier provides a robust foundation for automated food logging. In the context of user experience, this high precision significantly reduces the "cognitive load" on the user, as the need to manually correct the system's predictions is minimized. The consumption prediction model, with an AUC of 0.73, offers a heuristic baseline that performs significantly better than random chance, providing actionable intelligence to the user regarding potential spoilage.

### B. Architectural Trade-offs: Cloud vs. Edge

A key design decision was to host the inference engine on a remote server rather than on-device (Edge AI). While edge deployment offers privacy benefits and zero latency, it imposes significant constraints on model complexity and battery life. By offloading inference to the cloud, NutriSave ensures that the high-accuracy ConvNeXt model performs consistently regardless of the user's device capabilities. Furthermore, this decoupled architecture allows for the continuous deployment of improved AI models on the server without requiring users to download frequent application updates.

### C. Limitations and Future Work

The primary limitation of the current work is the reliance on synthetic data for the consumption prediction model. Future iterations will focus on collecting real-world longitudinal data from pilot deployments. As suggested by our analysis during development, exploring gradient-boosted decision trees like XGBoost, which are renowned for their performance on tabular data [25], could yield significant improvements over the current deep neural network approach for this specific task.

To further minimize user friction, future system updates will integrate auxiliary input methods. Specifically, we aim to incorporate barcode scanning for packaged goods [5] and Optical Character Recognition (OCR) to automatically extract "best by" dates from product packaging [26], thereby creating a fully seamless inventory logging experience.

## VII. Conclusion

This paper introduced NutriSave, an integrated mobile system designed to combat household food waste through intelligent automation. We detailed a rigorous methodology for fine-tuning a ConvNeXt architecture, achieving a state-of-the-art accuracy of 96.38% on food classification tasks. Furthermore, we demonstrated the technical implementation of a decoupled client-server architecture that balances model performance with mobile usability. By integrating high-precision computer vision with proactive consumption tracking, NutriSave establishes a scalable framework for technology-driven sustainability.

## References


[1] Food and Agriculture Organization of the United Nations (FAO), *The State of Food and Agriculture 2019. Moving forward on food loss and waste reduction*. Rome: FAO, 2019.

[2] G. Farr-Wharton, J. H. Choi, and M. Foth, "Food talks back: exploring the role of mobile applications in reducing domestic food wastage," in *OzCHI '14*, 2014, p. 352–361.

[3] S. Mamidala, "The SLED (Shelf Life Expiration Date) Tracking System: Using Machine Learning Algorithms to Combat Food Waste and Food Borne Illnesses," Jan 2023.

[4] R. Vinuesa, H. Azizpour, I. Leite, M. Balaam, V. Dignum, S. Domisch, A. Felländer, S. D. Langhans, M. Tegmark, and F. Fuso Nerini, "The role of artificial intelligence in achieving the sustainable development goals," *Nature Communications*, vol. 11, no. 1, p. 233, 2020.

[5] M. Pajpach *et al.*, "Exspiro - Mobile Application for Food Sustainability," in *2023 15th International Conference on Mobile, Hybrid, and on-line Learning (MOSICOM)*, 2023, pp. 77–82.

[6] H. Almurashi, B. Sayed, M. Khalid, and R. Bouaziz, "Smart expiry food tracking system," in *Advances in Information and Communication*. Springer Singapore, 2021, pp. 541–551.

[7] G. Eysenbach, "The law of attrition," *Journal of medical Internet research*, vol. 7, no. 1, p. e11, 2005.

[8] A. R. Soedomo and F. Heryanto, "Automatic fruit classification using deep learning for industrial applications," in *2018 International Conference on Information Management and Technology (ICIMTech)*, 2019, pp. 445–449.

[9] R. Sultana and I. Jahan, "Deep transfer learning cnn based for classification quality of organic vegetables," *International Journal of Advanced and Applied Sciences*, vol. 10, no. 12, pp. 211–219, 2023.

[10] Z. Chen and C. Wu, "Improved classification approach for fruits and vegetables freshness based on deep learning," *Sensors*, vol. 22, no. 21, p. 8192, 2022.

[11] H. Wang, J.-P. Emond, L. Jonkman, and D. Rong, "Facilitated machine learning for image-based fruit quality assessment," *Food Control*, vol. 143, p. 109315, 2023.

[12] X.-S. Wei, J.-H. Cui, Y. Wang, J.-R. Liu, Z. Wang, and J.-M. Feng, "Fine-grained visual classification: A comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 8939–8957, 2021.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," in *arXiv preprint arXiv:1704.04861*, 2017.

[15] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[16] H. Zhu, F. Liu, and L. Wang, "Deep learning in food freshness and quality assessment: A systematic review," *Trends in Food Science & Technology*, vol. 118, pp. 12–25, 2021.

[17] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2017.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems 25*, 2012.

[21] Y. Aoyagi and J. Ota, "Predicting user engagement with app notifications using a recurrent neural network," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 3153–3158.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[25] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[26] W.-Y. Ong, C. W. Too, and K.-C. Khor, "Transfer Learning on Inception ResNet V2 For Expiry Reminder: A Mobile Application Development," in *Computational Science and Technology*. Springer International Publishing, 2021, pp. 149–160.