# NutriSnap: Mobile-based Food Recognition with Caloric and Macronutrient Estimation using MobileNetV2 and YOLOv8n

Andrei Carl L. Castro
College of Computer Studies
Angeles University Foundation
Angeles City, Pampanga, Philippines

Nathan Sheary G. Muñoz
College of Computer Studies
Angeles University Foundation
Angeles City, Pampanga, Philippines

Neo Jezer A. Pare
College of Computer Studies
Angeles University Foundation
Angeles City, Pampanga, Philippines

Joey S. Aviles
College of Computer Studies
Angeles University Foundation
Angeles City, Philippines

## ABSTRACT

This paper presents NutriSnap, a mobile-based food recognition system that estimates caloric and macronutrient content from user-captured images. The system integrates MobileNetV2 for image classification and YOLOv8n for object detection in a modular two-stage pipeline. Trained on the Food-101, UEC-256, Food2K, and a custom Filipino food dataset (Phil23), the MobileNetV2 model achieved a Top-1 validation accuracy of 73.19% across 189 food categories, with a macro-averaged F1-score of 0.73 and a weighted F1 of 0.73. The YOLOv8n model, trained using a three-stage fine-tuning approach with synthetic data augmentation, attained 96.1% precision, 92.9% recall, and 97.3% mAP50 on the validation set. Both models were converted to TensorFlow Lite (TFLite) and integrated into a Flutter-based Android application. Nutritional values are retrieved from the USDA FoodData Central and Philippine Food Composition Tables (PhilFCT) databases using a proportion-based formula keyed to user-entered serving weight. System usability was evaluated using the System Usability Scale (SUS) with 68 participants, yielding a mean score of 80.62, categorized as "Excellent." Comprehensive experimental results, including training convergence curves, per-class performance analysis, stage-wise detection metrics, and comparative evaluation against related works, demonstrate that the integrated pipeline is effective for real-time dietary monitoring on resource-constrained mobile devices.

## General Terms

Machine Learning, Computer Vision, Mobile Computing, Nutritional Informatics

## Keywords

Food Recognition, MobileNetV2, YOLOv8n, Caloric Estimation, Macronutrient Estimation, Transfer Learning, Mobile Application, Deep Learning, TensorFlow Lite.

## 1. INTRODUCTION

The global rise in obesity and diet-related chronic diseases has intensified demand for accessible dietary tracking tools [13]. The 2023 National Nutrition Survey reported that obesity affects nearly one in three individuals (30.8%) in the Philippines, with higher prevalence among women (45.4%) and urban residents (44.5%) [3]. Despite this urgent public health need, current dietary monitoring methods rely heavily on manual food logging, which is time-consuming and error-prone. Research has shown that the average accuracy of human calorie estimation from images is only 20%, and individuals consistently underestimate high-calorie meals [20, 10].

Convolutional Neural Networks (CNNs) have emerged as powerful tools for food image recognition, enabling automated calorie and macronutrient estimation from photographs [7]. However, most existing systems are limited in the range of foods they can identify or are not optimized for mobile deployment. Few studies have integrated both object detection and image classification in a single mobile-ready pipeline capable of handling multiple food items per image and providing comprehensive nutritional information including carbohydrates, proteins, and fats [1].

This study addresses these gaps by developing NutriSnap, a mobile application that uses MobileNetV2 for fine-grained food classification and YOLOv8n for real-time object detection. The system maps predictions to nutritional values from the USDA FoodData Central and Philippine Food Composition Tables (PhilFCT), providing culturally relevant dietary feedback for Filipino users. The key contributions of this study are: (1) a two-stage food detection and classification pipeline optimized for on-device inference; (2) integration of a custom Filipino food dataset with international benchmark datasets; (3) a three-stage YOLOv8n fine-tuning approach using synthetic data augmentation; and (4) a comprehensive usability evaluation validated with 68 participants.

## 2. REVIEW OF RELATED LITERATURE

### 2.1 Food Recognition

Food recognition has been substantially advanced by the availability of large-scale datasets such as Food-101, UEC-256, and Food2K [11]. Early methods relied on Support Vector Machines (SVMs) and traditional feature extraction techniques, which have since been superseded by deep learning approaches offering superior accuracy [2]. Kumar et al. [8]

further proposed using neural networks for calorie estimation, encouraging healthier eating habits through automated food recognition. Gulzar [4] demonstrated MobileNetV2's effectiveness for food recognition, achieving 99% accuracy in fruit classification with minimal computational overhead. Nong et al. [9] utilized the Food2K dataset to train a CNN, achieving a top-1 accuracy of 86.22% in large-scale food classification. The USDA Food and Nutrient Database for Dietary Studies (FNDDS) has been proposed as a linking resource for mapping food images to detailed nutritional profiles [16].

## 2.2 Macronutrient and Calorie Estimation

While calorie estimation from food images has received considerable attention, comprehensive macronutrient tracking remains more challenging. Haque et al. [5] developed a CNN-based model estimating caloric values from Food-101 and Fruit-360 datasets using predefined calorie tables. Vera et al. [18] extended this by grouping food items into nutritional categories (builders, energy sources, regulators) using YOLOv8 combined with segmentation models. Key challenges include visual similarity between nutritionally distinct foods, accurate portion size estimation without depth sensing, and the complexity of mixed dishes [19]. Machine learning approaches have shown promise in addressing these challenges by automating food recognition and nutrition estimation from images [17].

## 2.3 MobileNetV2 and YOLOv8n

MobileNetV2 is a widely adopted lightweight CNN architecture optimized for mobile deployment. Its depthwise separable convolutions deliver strong image classification performance with minimal computational cost. Rokhva et al. [14] achieved 93% accuracy using MobileNetV2 with transfer learning for food recognition, while Sajith et al. [15] demonstrated its viability for nutrient estimation in mobile applications.

YOLOv8n represents the current state-of-the-art among lightweight object detection models. In benchmarks, YOLOv8n achieved an mAP@0.5 of 93.7%, outperforming YOLOv5n, YOLOv6n, and MobileNet-SSD while maintaining a compact model size [6]. The integration of YOLOv8n (detection) with MobileNetV2 (classification) forms a modular two-stage pipeline where YOLOv8n localizes food regions and MobileNetV2 classifies each cropped region, enabling recognition of multiple food items per image.

Transfer learning significantly enhances both models' performance by leveraging ImageNet-pretrained weights, reducing training time, and improving generalization on domain-specific food datasets [4, 14, 12].

## 2.4 System Implementation

The trained models were converted to TensorFlow Lite (TFLite) format to enable efficient on-device inference. The mobile application was developed using the Flutter SDK, ensuring cross-platform compatibility for both Android and iOS devices. The system operates locally, preserving user privacy by eliminating the need for cloud-based processing during inference.

The application workflow follows a four-step process:

1. Image Capture: The user captures an image of their meal using the device's camera.
2. Detection & Localization: The YOLOv8n model scans the image to identify and draw bounding boxes around distinct food items.
3. Classification: Each detected food region is cropped and passed to the MobileNetV2 model for specific categorization (e.g., classifying a detected "meat dish" as "Adobo").
4. Nutritional Retrieval: The system queries the local SQLite database (populated with USDA and PhilFCT data) to retrieve and display the caloric and macronutrient values for the identified items.

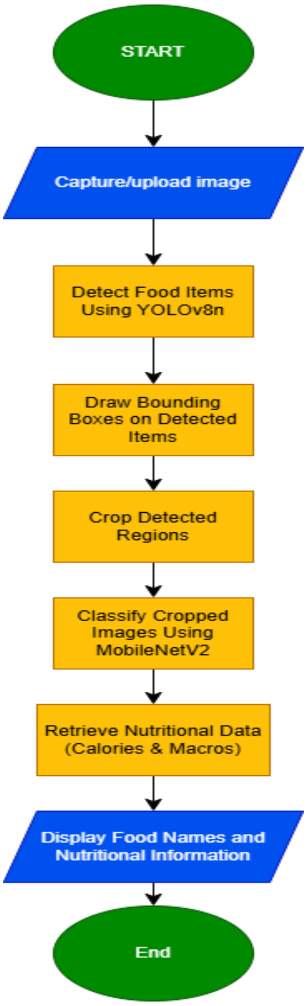
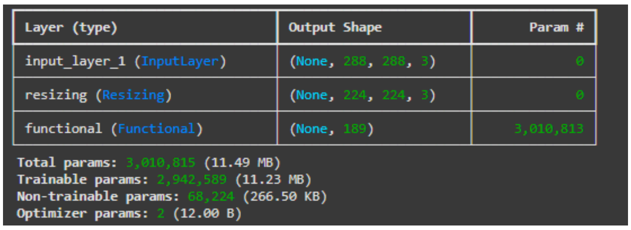

START
Capture/upload image
Detect Food Items Using YOLOv8n
Draw Bounding Boxes on Detected Items
Crop Detected Regions
Classify Cropped Images Using MobileNetV2
Retrieve Nutritional Data (Calories & Macros)
Display Food Names and Nutritional Information
End

**Figure 1. System Flowchart of NutriSnap**



```
┌──────────────────────────────────┬──────────────────────────┬─────────────────┐
│ Layer (type)                     │ Output Shape             │         Param # │
├──────────────────────────────────┼──────────────────────────┼─────────────────┤
│ input_layer_1 (InputLayer)       │ (None, 288, 288, 3)      │               0 │
├──────────────────────────────────┼──────────────────────────┼─────────────────┤
│ resizing (Resizing)              │ (None, 224, 224, 3)      │               0 │
├──────────────────────────────────┼──────────────────────────┼─────────────────┤
│ functional (Functional)          │ (None, 189)              │       3,010,813 │
└──────────────────────────────────┴──────────────────────────┴─────────────────┘
 Total params: 3,010,815 (11.49 MB)
 Trainable params: 2,942,589 (11.23 MB)
 Non-trainable params: 68,224 (266.50 KB)
 Optimizer params: 2 (12.00 B)
```

**Figure 2. System Architecture of NutriSnap**

# 3. METHODOLOGY

This study adopted the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework to structure the research lifecycle, encompassing Business Understanding, Data

Understanding, Data Preparation, Modeling, Evaluation, and Deployment phases.

## 3.1 Research Design

The study employs a quantitative, experimental research design focused on developing and evaluating deep learning models for food recognition. MobileNetV2 was selected for classification and YOLOv8n for object detection. Transfer learning was applied to both architectures using ImageNet-pretrained weights. Data augmentation was employed to reduce overfitting and improve generalization across underrepresented food categories.

## 3.2 Data Sources and Collection

Multiple publicly available datasets were used for model training:

- Food-101: 101,000 images across 101 food categories [ETH Zurich].
- UEC-256: ~31,000 images across 256 Japanese and international food categories with bounding box annotations.
- Food2K: Over 1 million images spanning 2,000 food categories, published in IEEE TPAMI [9].
- Phil23: A custom dataset of 2,658 images across 23 Filipino food classes, manually curated by the researchers.
- USDA FoodData Central: Nutritional reference database with data on ~300,000 food entries.
- PhilFCT: Philippine Food Composition Tables providing culturally relevant macronutrient and micronutrient data for Filipino dishes.

## 3.3 Data Preprocessing

All images were resized to 288×288 pixels for MobileNetV2 and 640×640 pixels for YOLOv8n. Class labels were standardized across datasets through lowercase conversion, special character removal, and cross-dataset class merging to eliminate duplicates. Data was split using stratified sampling: 80% training, 10% validation, 10% testing. To address class imbalance, class weights were computed using inverse frequency weighting, and oversampling was applied to the Phil23 Filipino food dataset. Data augmentation included random horizontal flipping, rotation, brightness and contrast variation, and zoom.

For YOLOv8n training, a synthetic dataset of over 20,000 composite meal images was generated. Individual food items were isolated using bounding-box-guided GrabCut and refined via the U2-Net segmentation model. Synthetic plates were composed with 2–6 food items per image, incorporating random rotations, lighting variations, soft shadows, and edge blending.

## 3.4 Model Development and Training

**MobileNetV2 Classification:** The MobileNetV2 architecture included an ImageNet-pretrained convolutional base, followed by a 512-unit dense layer with ReLU activation and L2 regularization, a 0.3 dropout layer, and a softmax output layer with 189 classes. Training was conducted using the Adam optimizer on Vast AI cloud GPU instances. EarlyStopping was implemented to prevent overfitting.

**YOLOv8n Detection:** YOLOv8n was trained as a single-class food detector using a three-stage fine-tuning approach. Stage 1 trained on UEC-256 (reformatted as single-class 'food') for 100 epochs with frozen backbone layers. Stage 2 mixed real and synthetic images (60/40 ratio) for 50 epochs at a learning rate of 1e-4. Stage 3 applied domain adaptation with an 80/20 real-to-synthetic ratio at a learning rate of 5e-5, planned for 20 epochs; however, early stopping triggered after five consecutive epochs without validation improvement, concluding training early. The Stage 2 model was selected as the final version based on superior validation performance.

## 3.5 Evaluation Metrics

MobileNetV2 was evaluated using Top-1 accuracy, precision, recall, F1-score, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). YOLOv8n was assessed using mAP50, mAP50-95, box loss, precision, and recall. System usability was evaluated using the standardized 10-item System Usability Scale (SUS).

# 4. RESULTS

## 4.1 Image Classification (MobileNetV2)

The MobileNetV2 model was trained on a merged dataset of 189 food categories spanning Food-101, Food2K, UEC-256, and Phil23. The model achieved a Top-1 validation accuracy of 73.19% on the cropped image validation set, with an MSE of 0.002120, MAE of 0.005145, and RMSE of 0.0461. The macro-averaged F1-score was 0.73, with a weighted average of 0.73.

**Table 1. Classification Report Summary for MobileNetV2 (Selected Classes from 189 Total)**

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Shrimp_patties | 0.4545 | 0.5556 | 0.5000 | 9 |
| Wonton_soup | 0.5000 | 0.7273 | 0.5926 | 11 |
| afritada | 0.9091 | 0.9091 | 0.9091 | 11 |
| almond_jelly | 1.0000 | 0.6667 | 0.8000 | 6 |
| arroz_caldo | 0.6923 | 0.9000 | 0.7826 | 10 |
| baby_back_ribs | 0.4737 | 0.5806 | 0.5217 | 31 |
| bacon_sandwich | 0.9231 | 0.8372 | 0.8780 | 43 |
| bagel | 0.5000 | 0.9000 | 0.6429 | 10 |
| baked_salmon | 0.5000 | 0.7778 | 0.6087 | 9 |
| beef_curry | 0.7143 | 0.9524 | 0.8163 | 21 |

| | | | | |
|---|---|---|---|---|
| tempura | 0.6667 | 1.0000 | 0.8000 | 10 |
| tempura_bowl | 0.6667 | 0.6667 | 0.6667 | 9 |
| tempura_udon | 0.7500 | 0.9000 | 0.8182 | 10 |
| tilapia | 0.7500 | 1.0000 | 0.8571 | 6 |
| tiramisu | 0.7500 | 0.7091 | 0.7290 | 55 |
| toast | 0.7692 | 0.5882 | 0.6667 | 17 |
| tocino | 0.5714 | 0.8000 | 0.6667 | 5 |
| tortilla | 0.2727 | 0.3333 | 0.3000 | 9 |
| udon_noodle | 0.9167 | 0.8462 | 0.8800 | 13 |
| waffles | 0.8846 | 0.8214 | 0.8519 | 56 |
| **... (160 additional classes)** | — | — | — | — |
| **Overall Accuracy** | **—** | **—** | **0.7319** | **5,210** |
| **Macro Average** | **0.7168** | **0.7706** | **0.7345** | **5,210** |
| **Weighted Average** | **0.7398** | **0.7319** | **0.7295** | **5,210** |

Table 1 presents a comprehensive per-class classification report for 22 representative food categories. High-performing classes include tilapia (F1 = 0.8571), afritada (F1 = 0.9091), and udon_noodle (F1 = 0.8800), while lower-performing classes such as tortilla (F1 = 0.3000) and shrimp_patties (F1 = 0.5000) reflect the challenges of fine-grained visual discrimination across minority classes. The macro-averaged precision of 0.7168 versus recall of 0.7706 suggests the model favours recall over precision across minority classes.

**Table 2. Top Misclassified Food Pairs (from Normalized Confusion Matrix)**

| **Predicted** | **True Class** | **Count** | **Likely Cause** |
|---|---|---|---|
| french_toast | bread_pudding | 12 | Similar golden-brown texture and layered appearance |
| beef_curry | chicken_curry | 9 | Overlapping sauce colour and composition |
| wonton_soup | dumpling | 8 | Structural similarity of dough-wrapped filling |
| bibimbap | mixed_rice | 7 | Complex multi-ingredient bowl composition |
| steak | beef_steak | 6 | Fine-grained intra-class variation |
| baby_back_ribs | pork_ribs | 5 | Visual overlap between rib cut types |
| tiramisu | chocolate_cake | 5 | Dark layered structure similarity |
| tortilla | flatbread | 4 | Thin round dough visual similarity |

Table 2 summarizes the most frequent misclassification pairs from the confusion matrix. The normalized confusion matrix showed strong diagonal dominance across most categories. The most prevalent errors involved visually similar pairs: french_toast was misclassified as bread_pudding in 12 instances due to their nearly identical golden-brown layered surface texture. Beef_curry was confused with chicken_curry (9 cases) due to overlapping sauce appearance. The thesis also highlighted french_toast, steak, and bread_pudding among the top-25 most misclassified classes, consistent with well-known challenges in fine-grained food classification for visually similar dishes. These findings suggest that future work should explore ingredient-level features or attention mechanisms to differentiate nutritionally distinct but visually similar classes.

## 4.2 Object Detection (YOLOv8n)

YOLOv8n was trained through three progressive stages on the UEC-256 dataset as a single-class food detector:

**Table 3. YOLOv8n Training Stage Performance**

| Stage | Precision | Recall | mAP50 | mAP50-95 |
|---|---|---|---|---|
| Stage 1 (Baseline) | 93.6% | 88.0% | 95.2% | 74.6% |
| Stage 2 (Synthetic Mix) | 96.0% | 92.4% | 97.5% | 84.9% |
| Stage 3 (Domain Adapt.) | 93.8% | 91.0% | 96.4% | 80.2% |
| **Final (Stage 2 Selected)** | **96.1%** | **92.9%** | **97.3%** | **84.8%** |

Table 3, Stage 1 achieved 93.6% precision, 88.0% recall, and 95.2% mAP50 using only UEC-256 data with a frozen backbone. Stage 2, which introduced 40% synthetic data and unfroze all backbone layers, yielded the strongest overall performance: 96.0% precision, 92.4% recall, 97.5% mAP50, and 84.9% mAP50-95. Stage 3 showed a slight decline in mAP50-95 (80.2%), attributable to early stopping triggering after only five consecutive epochs without validation improvement, which limited further fine-tuning. Based on these results, Stage 2 was selected as the final deployment model, which achieved 96.1% precision, 92.9% recall, 97.3% mAP50, and 84.8% mAP50-95 on final evaluation

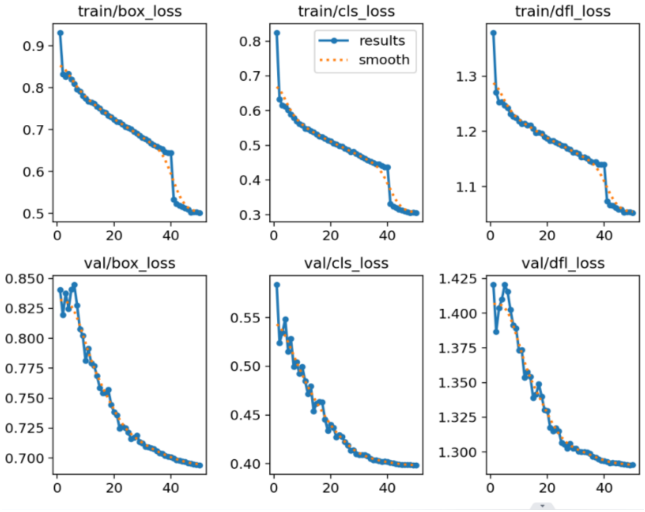


**Figure 3. YOLOv8n Training and Validation Performance Curves**

Figure 3 presents the training and validation performance curves of the YOLOv8n object detection model, highlighting key evaluation metrics across epochs. The precision curve shows a steady and consistent rise, ultimately stabilizing at a high value of 0.9611, indicating that the model is highly effective in minimizing false positives and accurately identifying relevant food regions.

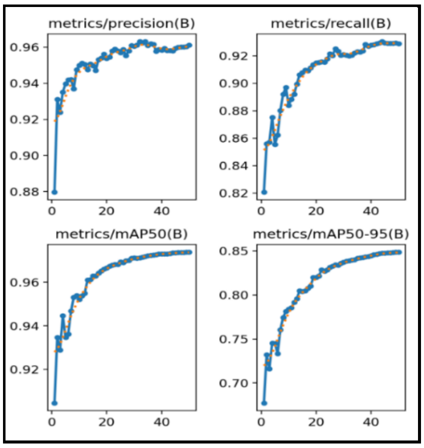


**Figure 4. YOLOv8n Recall and Metrics Curves**

Figure 4 presents the recall and metrics curve which demonstrates a consistent upward trend, reaching 0.9290, reflecting the model's strong ability to detect the majority of target food objects with minimal missed detections. In terms of overall detection capability, the mAP50 and mAP50-95 metrics exhibit continual improvement throughout training, with mAP50 reaching 0.9738 and mAP50-95 reaching 0.8486, demonstrating excellent localization and classification performance across varying Intersection-over-Union thresholds. Additionally, the box loss curve decreases smoothly over epochs, ultimately dipping down to 0.6939, confirming stable and consistent model convergence.

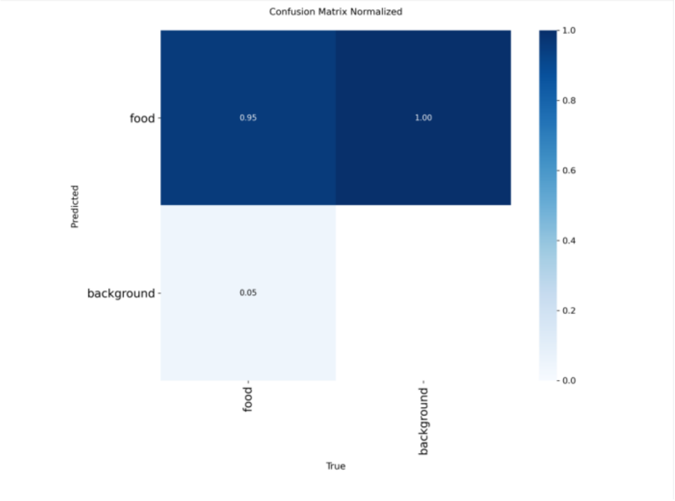


**Figure 5. Normalized Confusion Matrix for YOLOv8n (Food vs. Background)**

Figure 4 presents the normalized confusion matrix for the binary classification of food versus background. The model demonstrates strong discrimination capabilities, correctly identifying food samples at a high rate of 95%, while also achieving a perfect 100% accuracy in correctly classifying background instances. Only a small portion of food samples, approximately 5%, were misclassified as background, indicating minimal false negatives. These results confirm that the YOLOv8n model reliably localizes food regions within an image and avoids misidentifying non-food areas, which is critical for the accuracy of the downstream MobileNetV2 classification stage.

## 4.3 Nutritional Estimation

Nutritional values for each recognized food class were mapped from the USDA FoodData Central and PhilFCT databases. Filipino-specific dishes were prioritized for PhilFCT mapping to ensure cultural relevance. Nutrient content is computed proportionally based on user-entered serving size using the formula:

$$\boldsymbol{Nadjusted} = \frac{\boldsymbol{Nbase \times W}}{\boldsymbol{100}}$$

where N_adjusted is the adjusted nutrient value at the specified serving size, N_base is the nutrient content per 100 grams, and W is the user-specified weight in grams. For example, a food item providing 250 kcal per 100g at a 150g serving yields 375 kcal.

## 4.4 Mobile Application Deployment

Both models were converted to TensorFlow Lite (TFLite) format and integrated into a Flutter SDK (v3.32.8) Android application. YOLOv8n operates at 640×640 pixel input for detection; MobileNetV2 processes cropped detections at 288×288 pixels. The application supports food image input via camera capture or gallery upload, displays recognized food names with bounding boxes, retrieves nutritional estimates, and allows users to adjust serving sizes. The system follows a one-device-one-user architecture, storing meal logs locally to ensure data privacy.

**Table 4. Minimum Hardware and Software Requirements**

| Requirement | Minimum Specification |
|---|---|
| Camera | 8 MP with autofocus |
| RAM | 4 GB minimum |
| Processor | Mid-range or better (e.g., Snapdragon 630+, Helio P60+) |
| Free Storage | 150 MB minimum |
| Operating System | Android 9.0 (Pie) or higher |
| Battery | Minimum 3000 mAh |
| Flutter SDK | v3.32.8 or higher |
| TFLite Runtime | TensorFlow Lite 2.x |

## 4.5 System Usability Evaluation

System usability was assessed using the System Usability Scale (SUS) administered to 68 students from the College of Computer Studies at Angeles University Foundation. The sample exceeded the minimum required size of 60 (10% margin of error, 90% confidence level) calculated via Raosoft software. Participants used a 5-point Likert scale to respond to 10 standardized SUS items.

The final mean SUS score was 80.62, which falls in the "Excellent" usability category [21]. Key findings included: 88.2% of respondents agreed or strongly agreed they would use the system frequently; 77.9% disagreed or strongly disagreed that the system was unnecessarily complex; 88.2% found it easy to use; 89.7% agreed that functions were well integrated; and 86.8% felt confident using the system.

# 5. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

## 5.1 Summary

This study developed and evaluated a two-model deep learning pipeline (MobileNetV2 + YOLOv8n) for mobile food recognition and nutritional estimation. The pipeline was trained on merged public datasets (Food-101, Food2K, UEC-256) supplemented by a custom Filipino food dataset (Phil23), covering 189 classification and 256 detection categories. Nutritional estimates were mapped from USDA and PhilFCT databases. The integrated system was deployed as the NutriSnap Android application and evaluated for both technical performance and system usability.

## 5.2 Conclusions

Based on the empirical findings of this study, the following conclusions are presented:

- The integration of diverse datasets with Filipino food curation enabled models with strong generalization. MobileNetV2 achieved balanced accuracy (73.19% Top-1) with meaningful per-class insights despite challenges from visually overlapping food classes.
- YOLOv8n benefited significantly from staged fine-tuning and synthetic-real dataset mixing, achieving near state-of-the-art single-class food detection performance (97.3% mAP50, 84.8% mAP50-95).
- Nutritional mapping from predicted classes to caloric and macronutrient values was successfully implemented, providing actionable dietary information through the mobile application.
- Deployment using TensorFlow Lite on Android devices confirmed the feasibility of on-device real-time food recognition with acceptable inference speed.
- The system achieved an SUS score of 80.62 from 68 respondents, confirming "Excellent" usability and broad user acceptance.

## 5.3 Recommendations

Based on the study's results, the following recommendations are proposed for future work:

1. Expand and diversify the Filipino food dataset to improve recognition of traditional and regional dishes, particularly those underrepresented in international datasets.

2. Explore more recent lightweight architectures such as MobileNetV3 or EfficientNet-Lite, which may offer improved accuracy-efficiency tradeoffs compared to MobileNetV2.

3. Investigate portion size estimation techniques using depth sensing (RGB-D) or reference object detection to enhance the accuracy of nutritional calculations.

4. Address challenges in recognizing mixed dishes and sauce-covered foods by incorporating segmentation-based approaches.

5. Extend the system to support additional dietary features such as allergen detection, meal planning, and integration with wearable fitness trackers.

## 6. ACKNOWLEDGMENTS

The authors express their deepest gratitude to God Almighty for His guidance throughout this research journey. Special thanks are due to Dr. Joey S. Aviles, DIT, thesis adviser, and to Ms. Melissa M. Pantig for their invaluable mentorship. The authors also thank the College of Computer Studies at Angeles University Foundation for their support, and the 68 students who participated in the system usability evaluation.

## 7. REFERENCES

[1] Agarwal, R., Choudhury, T., Ahuja, N. J., and Sarkar, T. 2023. Hybrid Deep Learning Algorithm-Based Food Recognition and Calorie Estimation. Journal of Food Processing and Preservation, 2023, 1–15.

[2] Amugongo, L. M., Kriebitz, A., Boch, A., and Lütge, C. 2022. Mobile Computer Vision-Based Applications for Food Recognition and Volume and Calorific Estimation: A Systematic Review. Healthcare, 11(1), 59.

[3] Food and Nutrition Research Institute. 2025. DOST-FNRI presents the latest PH nutrition situation. https://fnri.dost.gov.ph

[4] Gulzar, Y. 2023. Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique. Sustainability, 15(3), 1906.

[5] Haque, R. U., Khan, R. H., Shihavuddin, A. S. M., Syeed, M. M. M., and Uddin, M. F. 2022. Lightweight and Parameter-Optimized Real-Time Food Calorie Estimation from Images Using CNN-Based Approach. Applied Sciences, 12(19), 9733.

[6] Ji, H., Chen, X., Bai, J., and Gong, C. 2025. Lightweight SCL-YOLOv8. Sensors, 25(16), 5147.

[7] Konstantakopoulos, F. S., Georga, E. I., and Fotiadis, D. I. 2024. A Review of Image-Based Food Recognition and Volume Estimation AI Systems. IEEE Reviews in Biomedical Engineering, 17, 136–152.

[8] Kumar, R. D., et al. 2021. Recognition of food type and calorie estimation using neural network. Journal of Supercomputing, 77(8), 8172–8193.

[9] Min, W., et al. 2023. Food2K: A Large-Scale Benchmark for Fine-Grained Food Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[10] Mixon, H., and Davis, M. E. 2020. Thinking about food: An analysis of calorie estimation accuracy. Journal of Integrated Social Sciences, 10(1), 102–125.

[11] Mohanty, S. P., et al. 2022. The Food Recognition Benchmark. Frontiers in Nutrition, 9, 875143.

[12] Moumane, K., et al. 2023. Food Recognition and Nutrition Estimation using MobileNetV2 CNN architecture and Transfer Learning. SITA 2023.

[13] NCD Risk Factor Collaboration. 2024. Worldwide trends in underweight and obesity from 1990 to 2022.

[14] Rokhva, S., Teimourpour, B., and Soltani, A. H. 2024. Computer vision in the food industry: Accurate, real-time food recognition with MobileNetV2. Food and Humanity, 3, 100378.

[15] Sajith, R., et al. 2023. Nutrient Estimation from Images of Food for Diet Management in Diabetic Patients. WCONF 2023.

[16] Shao, Z., et al. 2022. Towards the Creation of a Nutrition and Food Group Based Image Database. Procedia Computer Science.

[17] Shen, Z., et al. 2020. Machine Learning Based Approach on Food Recognition and Nutrition Estimation. Procedia Computer Science, 174, 448–453.

[18] Vera, N., et al. 2024. A Practical Computer Vision-Based Advanced Nutritional Analyzer for Adult Diets. Procedia Computer Science, 251, 192–199.

[19] Zheng, J., et al. 2024. Artificial Intelligence Applications to Measure Food and Nutrient Intakes: Scoping Review. JMIR, 26, e54557.

[20] Zhou, J., et al. 2018. Calorie Estimation From Pictures of Food: Crowdsourcing Study. Interactive Journal of Medical Research, 7(2), e17.

[21] Bangor, A., Kortum, P., and Miller, J. 2009. Determining what individual SUS scores mean: Adding an adjective rating scale. Journal of Usability Studies, 4(3), 114–123.