

# iLog-2.2: Volume and Nutrition Estimation for Mixed Foods via Mask R-CNN and Federated Learning

Indira Devi Siripurapu <sup>1</sup>, Laavanya Rachakonda <sup>2</sup>, Saraju P. Mohanty <sup>3</sup> and Elias Kougianos <sup>4</sup>

<sup>1</sup> Dept. of Computer Science and Engineering; IndiraDeviSiripurapu@my.unt.edu

<sup>2</sup> Dept. of Computer Science; rachakondal@uncw.edu

<sup>3</sup> Dept. of Computer Science and Engineering; saraju.mohanty@unt.edu

<sup>4</sup> Dept. of Computer Science and Engineering; elias.kougianos@unt.edu

\* Correspondence: IndiraDeviSiripurapu@my.unt.edu (I.D.S.); rachakondal@uncw.edu (L.R.)

## Abstract

Accurately estimating calorie intake and nutrient composition from what we eat remains one of the most practical challenges in maintaining a healthy lifestyle. Manual food logging and database-based estimations are often inaccurate because ingredient proportions and preparation styles vary widely. This paper presents, a lightweight, privacy-preserving framework that estimates calories and detailed nutrient values from a single image. The model uses a Mask R-CNN-based segmentation network to identify visible food components, measure their area, estimate their volume using preset height values, and map them to nutritional information obtained from reliable datasets such as USDA and Food-a-pedia. The system integrates federated learning (FL) to ensure privacy by allowing the model to improve collaboratively without sharing raw user data. The proposed architecture achieved a mean Average Precision (mAP) of 96% for detection and 92% for segmentation, confirming its precision and efficiency. A mobile prototype demonstrates the model's usability for real-time and secure nutrition estimation on edge-compatible devices.

**Keywords:** Food computing, Edge AI, Federated learning, Privacy, Nutrition estimation, Computer vision.

## 1. Introduction

The healthcare and nutrition industries are increasingly shaped by technological innovations that aim to quantify dietary patterns and energy intake. Accurate estimation of calorie and nutrient composition plays a vital role in weight management, chronic disease prevention, and personalized healthcare[1]. However, in real-world scenarios, this process is not straightforward[2]. People consume meals that often combine multiple ingredients, with varying portion sizes and preparation styles. Mixed dishes, such as sandwiches, bowls, and layered meals, make calorie estimation particularly challenging, since the same type of meal can differ greatly depending on regional recipes, ingredient density, and cooking variations.

These differences make it difficult for individuals to rely on general or pre-defined calorie charts, as such values represent only average estimates and do not account for user-specific variations. As a result, most people and healthcare professionals depend on manual food logging or rough estimation, which can be time-consuming and inaccurate[3]. Fig. 1 illustrates a person struggling to manually calculate calories and nutrient composition from a mixed meal, highlighting the complexity and need for an automated approach.

Received:

Revised:

Accepted:

Published:

**Copyright:** © 2026 by the authors.

Submitted to *Journal Not Specified* for possible open access publication under the terms and conditions of the

[Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).



**Figure 1.** User Unsure About Nutritional Values.

To address this challenge, iLog 2.2 focuses on precise and automated calorie and nutrient estimation using artificial intelligence (AI) and edge computing. Unlike the earlier [4] & iLog 2.0 [5], which handled multiple food categories and provided broader calorie predictions through object detection, this version enhances the overall accuracy and nutrient breakdown by modeling individual components in greater detail. The system applies geometric and visual analysis to estimate volume and mass directly from 2D images without requiring depth sensors or additional hardware. This refined approach not only predicts total calories but also estimates macronutrient and micronutrient composition, offering detailed insights into protein, fat, carbohydrates, sodium, calcium, and other essential nutrients.

A major contribution of this work is the integration of Federated Learning (FL) for privacy-preserving training and continuous model improvement [6]. In conventional AI-based food analysis systems, user data, such as food images are sent to centralized cloud servers, raising privacy and ethical concerns. In contrast, iLog 2.2 performs computation locally on the user's device. Each device trains the model on its private dataset and shares only encrypted model parameters-not raw images-with a central server for aggregation. This ensures that user data remains private, aligning with modern data protection standards such as General Data Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act of 1996 (HIPAA) [7] [8]. The federated architecture allows the model to learn from diverse user inputs while maintaining strong privacy guarantees.

Another core design goal of iLog 2.2 is efficiency and deployability. The system employs a lightweight deep learning model suitable for smartphones, embedded platforms, and Internet-of-Medical-Things (IoMT) devices, following the principles demonstrated in recent IoMT-based smart health systems [9]. This enables real-time inference with low latency and minimal computational overhead, making it practical for daily use. The modular nature of the framework also allows integration with mobile health applications, enabling users to share only summarized nutrient values with dietitians or healthcare professionals. Instead of exposing raw images or personal information, the system communicates aggregated nutritional summaries, maintaining both privacy and interpretability.

By combining computer vision, federated learning, and AI-driven automation, iLog 2.2 eliminates the guesswork involved in calorie and nutrient estimation. The framework

demonstrates how a focused and structured approach to analyzing mixed or multi-component meals can serve as a proof of concept for future large-scale food estimation systems. In doing so, it bridges the gap between nutrition science, privacy-preserving AI, and practical usability. This research contributes to the broader goal of personalized healthcare through intelligent dietary monitoring, where individuals receive accurate, secure, and adaptive feedback on their nutritional intake without compromising privacy. Fig. 2 illustrates a user effortlessly obtaining detailed nutritional information through the proposed application.



**Figure 2.** User Retrieves Calorie and Nutrient Report Using the iLog 2.0 System Framework.

## 2. Novelty of the Proposed Solution

The proposed system, iLog 2.2, introduces several novel contributions that differentiate it from existing calorie estimation and nutrition analysis frameworks. Traditional approaches often treat complex meals as single homogeneous entities, providing only approximate calorie estimates. In contrast, iLog 2.2 models the problem at a finer granularity, focusing on ingredient-level analysis within mixed or multi-component meals. The framework combines federated learning, edge-compatible design, and 2D image geometry estimation to deliver accurate and privacy-preserving nutrition insights. The major novelties of this work are summarized as follows:

- **Mixed Meal Analysis:** Instead of estimating total calories for a meal as a single unit, iLog 2.2 analyzes each visible ingredient individually. This makes it one of the first systems to address calorie estimation for *mixed foods*, where components vary in thickness, density, and volume.
- **Federated Learning for Privacy:** The model incorporates a federated learning framework that allows distributed devices to participate in collaborative training without sharing any raw images or personal data. Only encrypted model parameters are communicated, ensuring complete privacy and security of user information. This

approach enables large-scale model improvement while fully complying with data protection standards such as GDPR and HIPAA.

- **Lightweight and Edge-Compatible Model:** A compact and computationally efficient deep learning model has been designed to operate efficiently on low-power devices and to be deployable in future edge environments. The lightweight nature of the model reduces latency and energy consumption, enabling fast and efficient inference.
- **Flexible Image Input:** The system accepts images captured from *any camera angle*, removing the need for fixed reference perspectives or additional calibration objects. This flexibility enhances usability in real-world conditions, where users capture spontaneous food images in uncontrolled settings.
- **Polygon-Based Segmentation:** Each ingredient is annotated using polygonal masks rather than bounding boxes, allowing precise delineation and improved computation of surface area. The segmentation output directly contributes to the accuracy of subsequent volume and mass estimation.
- **No Reference Image Requirement:** Unlike depth-based or multi-view estimation systems, iLog 2.2 does not rely on any external reference image or additional camera calibration. The use of preset geometric priors for height and ingredient thickness makes the system simple, efficient, and fully self-contained.
- **Volume-to-Weight Conversion:** The system converts 2D pixel-based areas into estimated volume using preset height values, and then translates the volume into mass (grams) using ingredient-specific density factors. This conversion enables both weight and calorie contribution to be estimated per ingredient.
- **Macro and Micro Nutrient Estimation:** Beyond calorie computation, iLog 2.2 provides a detailed nutrient profile. It calculates macronutrients such as carbohydrates, proteins, and fats, and micronutrients including sodium, calcium, and iron for each detected component as well as for the complete meal. This approach overcomes the limitations of generalized calorie estimates by providing ingredient-level nutritional breakdowns.
- **Dynamic Calorie Estimation:** The system accounts for variations in ingredient quantity and distribution. Unlike conventional systems that assume fixed calorie values for standard meals, iLog 2.2 dynamically adjusts its output according to the detected components and their computed weights, resulting in highly personalized and accurate estimates.

Overall, iLog 2.2 redefines calorie and nutrient estimation by combining federated learning, edge-compatible design, and geometric analysis into a unified and scalable framework. It provides a privacy-aware and interpretable solution that not only estimates calories but also generates detailed nutritional insights tailored to each user's intake.

### 3. Related Work

The comparison presented in Table 1 highlights how most existing research in food vision has focused on broad multi-class food detection or image-level calorie estimation. While these studies demonstrate strong recognition accuracy, they generally assume each dish has a fixed calorie value and overlook per-ingredient variability[10]. Several frameworks such as iLog[4][5], FoodSegDL[11,12], and CalorieAI[13], employ deep detectors or segmentation models like Mask R-CNN[14], and YOLO[15] to identify foods, but they either depend on static calorie tables or require additional sensors for volume estimation. iLog 3.0 [16] improves volume accuracy by incorporating monocular depth estimation, but depth/multi-view components add compute and system complexity that can be tough for lightweight deployments. Only a few works explore federated learning, and even then, they rarely combine FL with detailed nutrient analysis. The proposed iLog 2.2 framework stands

out by combining edge deployability, federated privacy, and per-topping macro–micro nutrient estimation derived from computed volume and weight.

**Table 1.** Comparison of Research Works in Food Vision, Portion/Volume, and Nutrition Estimation

Title	Dataset / Input	Core Method	Vol. Est.	Nutrition Granularity	Federated Learning / Edge Compatibility	Key Insights
iLog (Rachakonda et al.) [4]	Mixed / food images + sensing (IoMT)	Detector + intake monitoring + stress detection (IoMT device)	No	Per-intake / monitoring oriented (not detailed nutrients)	No	Focuses on automatic intake monitoring + stress detection; not a full nutrient pipeline.
iLog 2.0 (Mitra et al.) [5]	Multi-class food images	Food detection + nutrition quantification	No	Per-dish (general)	No	Limited volumetric/geometry-driven accuracy.
iLog 3.0 (Siripurapu et al.) [16]	2D RGB food images	Mask R-CNN + monocular depth (MiDaS-style) for volume inference	Yes	Per-dish (derived)	No	Accurate volume from 2D + depth; higher compute than pure detection.
Mask R-CNN Volume + Calories (Dai et al.) [17]	RGB images (solid foods; example-based)	Mask R-CNN instance segmentation + calibration-based calorie estimation	Approx. (via calibration)	Global calories (per item / dish)	No	Strong masks; needs calibration/assumptions; shows extension to multiple solid foods.
SEG-FOOD Dataset/Segmentation (Tahir & Kiong) [11]	Food segmentation dataset (semantic masks)	Deep learning semantic segmentation (dataset + baseline models)	No	N/A	No	Useful benchmark dataset for segmentation; no portion/nutrient computation.
Lightweight DeepLabv3+ (Muñoz et al.) [12]	Multiple food segmentation datasets + self-collected set	EfficientNet-B1 backbone + CWASPP + SE attention (semantic seg.)	No	Region-level only	Edge-friendly (lightweight)	High segmentation performance with reduced compute; does not estimate calories/volume.
mid-DeepLabv3+ + CamerFood10 (Erep et al.) [18]	African food dataset (CamerFood10)	Enhanced DeepLabv3+ (ResNet50 + SimAM) for semantic seg.	No	Region-level only	No (not FL)	Improves segmentation on underrepresented cuisines; still no nutrient pipeline.
FDSNet (Xiao et al.) [19]	FoodSeg103 / UECFoodPix-Complete	Dual-branch (Swin Transformer + CNN) multi-scale fusion segmentation	No	Region-level only	No	Better accuracy/efficiency for large images; segmentation-focused.
Two-View 3D Reconstruction (Dehais et al.) [20]	Multi-view images (two views)	Two-view 3D reconstruction for volume estimation	Yes (3D)	Per-dish (can map to nutrition if coupled)	No	Higher accuracy; requires multi-view capture/setup.
Single-View Geometric Models (Fang et al.) [21]	Single RGB image	Single-view geometric priors/models for portion estimation	Approx.	Per-dish (if coupled with nutrient DB)	No	Lightweight classical approach; accuracy depends on assumptions/geometry fit.
CalorieAI (Chandra et al.) [13]	Custom dataset	YOLOv8 detector + image-based calorie estimation	No	Global calories	No	Fast detection-driven pipeline; lacks rigorous geometry/volume computation.
YOLOv5 Calories Estimation (Shams et al.) [15]	(Chapter; dataset not explicit in BibTeX)	YOLOv5-based recognition + calorie estimation	No	Global calories	No	Detection-first approach; accuracy depends on dataset + mapping assumptions.
Mobile Nutrient Analysis (Revesai & Kogeda) [22]	Mobile food images	Lightweight interpretable DL + KD + Grad-CAM/LIME for nutrient estimation	No	Per-dish nutrients (estimation)	Yes (mobile/edge)	Mobile-friendly + interpretable; focuses on nutrients (not explicit volume).
FL for Food Service Industry (Taheri Gorji et al.) [7]	Distributed fluorescence imaging frames	Federated learning (FedAvg) + MobileNetv3 / DeepLabv3+	No	N/A (cleanliness auditing)	Yes (FL)	Strong example of FL privacy in food-service vision; not food portion/nutrition.
PET Survey for FL in IoHT (Mosaiyebzadeh et al.) [23]	Survey	Privacy-enhancing technologies for FL in IoHT	No	N/A	Yes (survey)	Summarizes PETs and challenges; background support for privacy motivation.
LDP for FL (Behnia et al., arXiv) [8]	General FL setting	Local Differential Privacy with fixed memory + per-client privacy	No	N/A	Yes (privacy)	Useful privacy method reference for FL; not food-specific.
Mask R-CNN (He et al.) [14]	General vision benchmark work	Instance segmentation backbone (Mask R-CNN)	N/A	N/A	N/A	Core method reference for instance segmentation used by iLog 3.0-style pipelines.

Table 2 compares popular commercial food-logging applications with the proposed system. These apps such as Lose It! [24], MyFitnessPal [25], HealthifyMe [26], and BiteSnap [27], primarily rely on manual data entry, barcode scanning, or cloud-based recognition services. Although convenient for casual diet tracking, they provide only approximate calorie information and require user input for portion size. However, most existing methods do not fully support automatic ingredient-level segmentation or nutrient estimation for individual

components in mixed dishes. In contrast, the proposed iLog 2.2 model operates fully on the device, accepts images from any camera angle, and uses polygon-based segmentation to compute per-topping volume, weight, and nutritional composition while preserving user privacy through federated learning.

**Table 2.** Comparison of Commercial Food-Logging Applications

Application	Input Type	Automation	Insights
See How You Eat [28]	Manual photo logging / visual diary	None (mostly manual)	Primarily a photo-based diary; lacks automatic detection and portion/volume estimation.
Lose It! [24]	Manual logging + barcode scan	Database lookup + barcode-based matching	Good for calorie tracking from known foods; limited vision-based segmentation or portion-size estimation from images.
AteMate [29]	Photo diary entries + manual notes	Manual tagging / habit tracking	Supports qualitative tracking; typically does not compute calories/volume from a single food photo.
Bite AI [27]	Camera input (food photos)	Cloud-based AI recognition (image-to-food labeling)	Convenient recognition, but often cloud-dependent and may not provide robust per-ingredient segmentation or accurate portion volume.
HealthifyMe [26]	Manual logging + AI chat input (text)	Semi-automated recommendations + database-driven logging	Strong diet-plan ecosystem; image-based portion estimation is limited compared to geometry/volume-based research systems.
MyFitnessPal [25]	Manual + barcode scan + voice input	Large database + barcode/recipe features	Widely used food database; does not focus on vision-based segmentation/volume estimation from images.
Yazio [30]	Manual + AI-assisted input (varies by feature)	Partial automation (plans/recipes/database)	Fast tracking and planning; limited in photo-based per-ingredient segmentation and accurate portion quantification.
Foodvisor [31]	Camera + AI model input	Cloud recognition + nutrition estimate	Provides image-based nutrition estimation, but accuracy can depend on cloud model and may not reliably infer portion volume.
EatWise [32]	Manual reminders + meal timing	Manual tracking / reminders	Focuses on eating schedule/behavior; not designed for calorie, macro, or volume estimation from images.

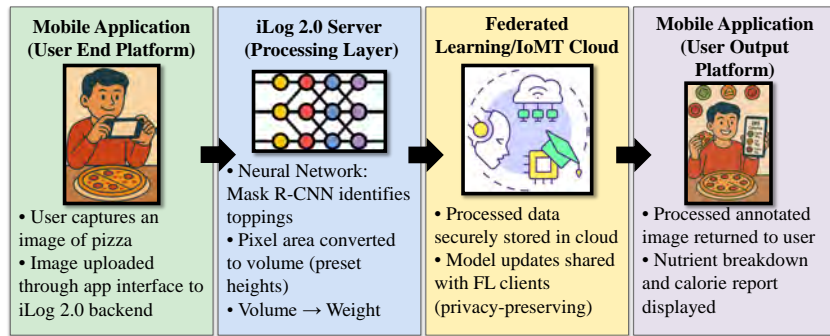
#### 4. Methodology / Overview of the Proposed Model

The proposed iLog 2.2 model is designed to make food calorie and nutrient estimation simple, accurate, and privacy-preserving for end users. The process begins when a user captures an image of their food using the mobile application. The image is directly uploaded through the user interface to the iLog 2.2 system, where the backend neural network, built on a lightweight Mask R-CNN architecture, processes it to identify and segment individual components of the meal.

Once the image is received, the system automatically detects each visible ingredient and separates it using polygon-based masks to preserve fine detail and boundary precision. These segmented regions are then analyzed to determine their relative area, which is later used to compute volume and mass for nutrient estimation. The entire workflow from image upload to calorie and nutrient breakdown is handled seamlessly by the application without requiring any manual input or external calibration. This design ensures a smooth user experience while maintaining reliability and scalability for everyday use across various food types and real-world lighting or background conditions.

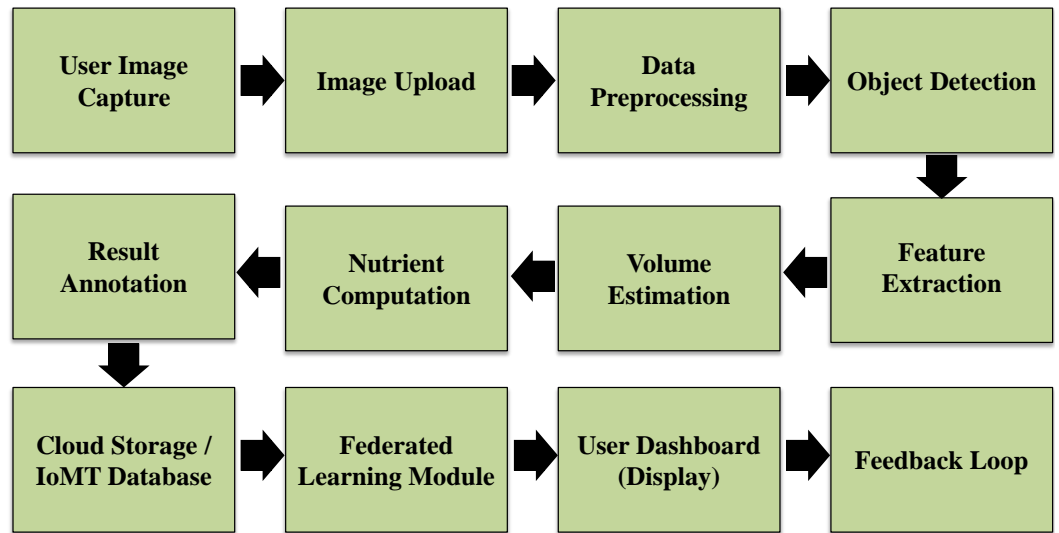
All further computations, such as converting detected topping areas into physical measurements, take place in the model's processing layer. The detected regions are first analyzed to calculate pixel area, which is then converted to estimated volume using preset height values. This volume is converted into weight (grams), and from this, the model derives calories, macronutrients (carbohydrates, proteins, fats), and micronutrients (sodium, calcium, iron, etc.) for each topping as well as for the entire pizza. An overview of the proposed system is presented in Fig. 3, and the step-by-step development workflow is depicted in Fig. 4.

Pizza has been considered in this study as it is a globally consumed food item and serves as a representative example of a mixed meal containing multiple components. The variety of toppings, sauces, and ingredient combinations introduces significant complexity



**Figure 3.** Overview of the proposed iLog 2.2 System Showing the Major Components and Their Interactions.

in accurately estimating nutritional values, making it an ideal case for testing the real-world performance of food estimation models.



**Figure 4.** Development Workflow of the Proposed iLog2.2 System Illustrating Each Stage From Image Capture to Nutrient Estimation.

After the calculations are completed, the processed image, now annotated with identified toppings and their corresponding nutritional values, is sent back to the user through the mobile application’s interface. The user can see the final output image along with detailed nutritional information, including the calories and macro–micro breakdown, making the estimation fully interpretable and personalized.

## 5. Implementation

### 5.1. Dataset



**Figure 5.** Sample Images From the Curated Pizza Dataset Used in the iLog 2.2 Model.

This work uses two integrated datasets, one for visual data and the other for nutritional reference values. The image dataset contains a total of 1,107 pizza images, captured from different angles and lighting conditions to ensure variety and real-world representation. Out of these, 666 images are used for training, 222 for testing, and the remaining 219 images for validation. This allocation prioritizes learning capacity while maintaining independent validation and testing for fair evaluation.

Every image is manually annotated using polygon masks with 50 labels that represent various toppings, sauces, and cheese types commonly found on pizzas. As shown in Fig. 5, the dataset includes a diverse variety of pizzas with multiple topping combinations.

The nutritional dataset is curated from verified public resources such as the U.S. Department of Agriculture (USDA) database [33] and Data.Gov: Food-a-pedia [34]. Nutritional information including calorie values, macronutrients (carbohydrates, proteins, fats), and micronutrients (sodium, calcium, iron, etc.) was collected for each topping based on multiple portion sizes such as one cup, one slice, and different thickness levels. These values were manually refined to maintain consistency across units and portion categories.

Finally, both datasets were combined to form a single structured dataset that links each annotated topping label with its corresponding nutritional information. This unified

**Algorithm 1** Topping Detection and Segmentation using Mask R-CNN

---

**Require:** RGB image  $I$  (any angle); trained weights  $W$ ; class list  $\mathcal{L}$ ; score threshold  $\tau_s$ ; mask threshold  $\tau_m$ ; NMS IoU  $\tau_n$ ; minimum area  $A_{\min}$

**Ensure:** Set of instances  $\mathcal{S} = \{(\ell_i, M_i, N_i, \text{conf}_i, \text{bbox}_i)\}$

- 1:  $I' \leftarrow \text{Preprocess}(I)$   $\triangleright$  resize to  $512 \times 512$ , normalize, optional denoise
- 2:  $\mathcal{R} \leftarrow \text{MaskRCNN\_Infer}(W, I')$   $\triangleright$  proposals with class posteriors, mask logits, boxes
- 3:  $\mathcal{R} \leftarrow \text{NMS}(\mathcal{R}, \text{IoU} = \tau_n)$   $\triangleright$  remove redundant detections
- 4:  $\mathcal{S} \leftarrow \emptyset$
- 5: **for** each detection  $r \in \mathcal{R}$  **do**
- 6:   **if**  $\text{score}(r) < \tau_s$  **then**
- 7:     **continue**
- 8:   **end if**
- 9:    $M \leftarrow \mathbf{1}(\sigma(r.\text{mask\_logits}) \geq \tau_m)$   $\triangleright$  binarize mask
- 10:    $M \leftarrow \text{MorphRefine}(M)$   $\triangleright$  remove speckles, fill small holes
- 11:   **if**  $\text{Area}(M) < A_{\min}$  **then**
- 12:     **continue**
- 13:   **end if**
- 14:    $(M, \text{bbox}) \leftarrow \text{ClipToImage}(M, I')$
- 15:    $\ell \leftarrow r.\text{class\_probs}; \text{conf} \leftarrow \max r.\text{class\_probs}$
- 16:    $N \leftarrow \text{PixelCount}(M)$
- 17:    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\ell, M, N, \text{conf}, \text{bbox})\}$
- 18: **end for**
- 19:  $\mathcal{S} \leftarrow \text{ResolveOverlaps}(\mathcal{S})$   $\triangleright$  mask-wise soft-NMS or priority by confidence/area
- 20: **return**  $\mathcal{S}$   $\triangleright$  to volume/weight computation stage

---

dataset enables the iLog 2.2 model to compute accurate calories and nutrients directly from topping-level predictions in the image, ensuring that the visual segmentation output aligns precisely with real nutritional metrics.

### 5.2. Mask R-CNN Based Detection and Segmentation

The next stage of the system focuses on detecting and segmenting each component present in the image. For this, a customized Mask R-CNN model is used because it can perform both object detection and instance segmentation within a single framework. Mask R-CNN has been widely adopted in computer vision tasks for its ability to generate high-quality segmentation masks while maintaining strong detection accuracy [14]. The model identifies every visible topping such as cheese, sauce, onion, tomato, pepper, olive, and others and separates them using polygon masks. Each topping mask is stored with a unique label that links it to its nutritional data.

The image captured from the user's mobile camera is first resized and normalized before being passed to the Mask R-CNN model. The backbone network extracts deep visual features, while the Region Proposal Network (RPN)[35] generates candidate regions where toppings may exist. The final segmentation head produces binary masks for each detected object, outlining the shape and size of individual toppings accurately.

All detected masks are then post-processed to remove overlaps and false positives. Each topping mask is associated with its class ID and pixel count, which will later be used for volume estimation. Because the model is trained with polygon annotations, it performs well on irregular topping shapes and overlapping regions. The output of this step is the segmented image with labeled toppings and their corresponding pixel areas, which is then passed to the next computation stage for volume and nutrient estimation.

### 5.3. Volume Estimation

After detecting and segmenting each topping using the Mask R-CNN model, the next important step is to determine the physical volume of each region. The system follows a

preset height-based estimation approach, which removes the need for any reference object or external calibration.

Based on prior studies in food geometry and serving thickness, the average height of a standard pizza was determined to be approximately 2 cm, as supported by [36,37]. Using this value as a constant base height ( $h_{base}$ ), the system estimates the height for other toppings proportionally using their recorded serving dimensions in the FoodData Central[33] and Food-a-pedia[34] nutritional datasets. This ensures that each topping has a realistic relative height for computing its volume.

The area for each topping is obtained from the segmentation masks in pixels. The Mask R-CNN output provides a pixel count  $N_i$  for each topping  $i$ , which is converted to an actual surface area  $A_i$  in  $\text{cm}^2$  using a learned scale prior  $s^*$  (cm/pixel). The relationship is given as:

$$A_i = N_i \times (s^*)^2 \quad (1)$$

As shown in equation (1), the number of detected pixels for each topping is multiplied by the learned scale to convert it into the corresponding surface area in squared cms.

The estimated topping height  $h_i$  (in cm) is then used to compute the volume  $V_i$  of that topping:

$$V_i = A_i \times h_i \quad (2)$$

As shown in equation (2), the topping volume is derived by multiplying the detected surface area with the estimated height, which represents the average thickness of that ingredient.

Once the volume is calculated, it is converted into weight (mass) using the corresponding density value  $\rho_i$  ( $\text{g}/\text{cm}^3$ ) obtained from the USDA dataset:

$$m_i = V_i \times \rho_i \quad (3)$$

As shown in equation (3), each topping's density is multiplied by its volume to estimate its total mass in grams.

Since most food nutrient data is standardized in grams, all further calculations use  $m_i$  in grams. When applicable, conversions are performed using the equivalence:

$$1 \text{ cup} = 240 \text{ grams}(\text{approx})[38] \quad (4)$$

As shown in equation (4), this ensures accurate scaling between volume-based and mass-based quantities.

Therefore, the final output of this step is the estimated mass  $m_i$  for each topping, which serves as the input for the next module the nutrient and calorie estimation engine, as shown in equation (3).

#### 5.4. Nutrient and Calorie Computation

Once the weight of each topping is obtained from the previous step, the system converts these weights into calories and nutrient values using the curated nutritional dataset. The nutrient data for each topping includes calories, proteins, fats, carbohydrates, sodium, calcium, iron, and other macro and micro components, all expressed per gram. These values are collected from the USDA [33] and FoodData Central datasets [34], where each food item is standardized to a per-gram nutritional profile.

**Algorithm 2** Volume  $\rightarrow$  Mass (grams)  $\rightarrow$  Nutrients with Preset Height

---

**Require:** Segmented instances  $\mathcal{S} = \{(\ell_i, M_i, N_i)\}$ ; scale prior  $s^*$  (cm/px);

- 1: base height  $h_{\text{base}} = 2$  cm; topping heights table  $\mathcal{H}[\ell]$  (cm, optional);
- 2: density table  $\rho[\ell]$  (g/cm<sup>3</sup>); nutrient table  $\mathcal{N}[\ell]$  with per-gram factors;
- 3: serving conversions  $\mathcal{C}$  (e.g., 1 cup = 240 g, tbsp, slice, etc.)

**Ensure:** For each topping  $i$ : area  $A_i$  (cm<sup>2</sup>), volume  $V_i$  (cm<sup>3</sup>), mass  $m_i$  (g), calories  $C_i$  (kcal), macro/micro vector  $\mathbf{n}_i$ ; and whole-pizza totals

- 4: **for** each instance  $i$  with label  $\ell_i$  and pixel count  $N_i$  **do**
- 5:  $A_i \leftarrow N_i \times (s^*)^2$   $\triangleright$  pixel area  $\rightarrow$  physical area (cm<sup>2</sup>)
- 6:  $h_i \leftarrow \begin{cases} \mathcal{H}[\ell_i], & \text{if available} \\ h_{\text{base}}, & \text{otherwise} \end{cases}$
- 7:  $V_i \leftarrow A_i \times h_i$   $\triangleright$  volume in cm<sup>3</sup> (preset/lookup height)
- 8:  $m_i \leftarrow V_i \times \rho[\ell_i]$   $\triangleright$  convert volume to mass (grams)
- 9:  $(e_{\text{kcal/g}}, \mathbf{f}_{\text{per g}}) \leftarrow \mathcal{N}[\ell_i]$   $\triangleright$  energy and macro/micro factors per gram
- 10: **if**  $\mathcal{N}$  entry is given per serving (cup/slice/tbsp) **then**
- 11:  $g_{\text{per serving}} \leftarrow \mathcal{C}[\text{serv\_type}]$   $\triangleright$  e.g., 1 cup = 240 g
- 12: Convert factors to per-gram:  $e_{\text{kcal/g}} \leftarrow \frac{e_{\text{kcal/serv}}}{g_{\text{per serving}}}$ ;  $\mathbf{f}_{\text{per g}} \leftarrow \frac{\mathbf{f}_{\text{per serv}}}{g_{\text{per serving}}}$   $\triangleright$  calories for topping  $i$
- 13:  $C_i \leftarrow m_i \times e_{\text{kcal/g}}$   $\triangleright$  calories for topping  $i$
- 14:  $\mathbf{n}_i \leftarrow m_i \times \mathbf{f}_{\text{per g}}$   $\triangleright$  macro/micro nutrient vector for topping  $i$
- 15: **end for**
- 16:  $A_{\text{tot}} \leftarrow \sum_i A_i$ ;  $V_{\text{tot}} \leftarrow \sum_i V_i$ ;  $m_{\text{tot}} \leftarrow \sum_i m_i$
- 17:  $C_{\text{tot}} \leftarrow \sum_i C_i$ ;  $\mathbf{n}_{\text{tot}} \leftarrow \sum_i \mathbf{n}_i$
- 18: **return**  $\{(A_i, V_i, m_i, C_i, \mathbf{n}_i)\}_i$  and  $(A_{\text{tot}}, V_{\text{tot}}, m_{\text{tot}}, C_{\text{tot}}, \mathbf{n}_{\text{tot}})$

---

For every topping  $i$ , the total calorie and nutrient values are computed by scaling the per-gram nutrient factors with the estimated weight  $m_i$  in grams. The overall calorie value  $C_i$  for each topping is calculated as:

$$C_i = m_i \times E_i \quad (5)$$

As shown in equation (5), the total calorie content for each topping is obtained by multiplying the estimated mass by its per-gram energy value (kcal/g) from the nutritional dataset.

Similarly, the macro and micro nutrients are computed using:

$$\mathbf{n}_i = m_i \times \mathbf{f}_i \quad (6)$$

As shown in equation (6), the nutrient vector  $\mathbf{n}_i$  for each topping is calculated by scaling the per-gram nutrient values such as protein, fat, carbohydrates, sodium, and calcium by the mass  $m_i$ .

Finally, the total nutritional values for the entire pizza are computed by summing all topping contributions:

$$C_{\text{total}} = \sum_i C_i, \quad \mathbf{n}_{\text{total}} = \sum_i \mathbf{n}_i \quad (7)$$

As shown in equation (7), the total calorie value  $C_{\text{total}}$  and the aggregated nutrient vector  $\mathbf{n}_{\text{total}}$  represent the complete nutritional profile of the pizza, combining all detected toppings to give per-topping and overall composition.

The advantage of this approach is that it removes the need to assume a fixed calorie count for all pizzas. Instead, each pizza is analyzed individually based on its detected toppings, thickness, and densities. The output image sent back to the user includes visual annotations of the detected toppings, their calories, and the macro–micro distribution, providing a complete and interpretable nutritional report for a single pizza image.

**Algorithm 3** Nutrient and Calorie Computation

**Require:** For each topping  $i$ : estimated mass  $m_i$  (g); nutrient table  $\mathcal{N}$  containing per-gram values for calories  $E_i$  and macro/micro vector  $\mathbf{f}_i$

**Ensure:** For each topping instance  $i$ : calories  $C_i$ , nutrient vector  $\mathbf{n}_i$ ; and totals for the full pizza

- 1: **for** each topping instance  $i$  **do**
- 2:      $(E_i, \mathbf{f}_i) \leftarrow \mathcal{N}[\ell_i]$  ▷ lookup per-gram energy and nutrient factors
- 3:      $C_i \leftarrow m_i \times E_i$  ▷ total calories (kcal) for topping  $i$
- 4:      $\mathbf{n}_i \leftarrow m_i \times \mathbf{f}_i$  ▷ scale nutrients to mass in grams
- 5: **end for**
- 6:  $C_{\text{total}} \leftarrow \sum_i C_i$  ▷ sum of calories for entire pizza
- 7:  $\mathbf{n}_{\text{total}} \leftarrow \sum_i \mathbf{n}_i$  ▷ aggregate all macro and micro nutrients
- 8: Annotate output image with  $\{(\ell_i, C_i, \mathbf{n}_i)\}$  and total summary
- 9: **return**  $\{C_i, \mathbf{n}_i\}$  for each topping and  $(C_{\text{total}}, \mathbf{n}_{\text{total}})$  for the pizza

## 5.5. Federated Learning

To maintain user privacy while still improving the model over time, the iLog 2.2 adopts a federated learning (FL) framework. In this setup, all user data such as food images and nutrient outputs remain on the local device. The learning process happens collaboratively across multiple users, where each device trains the model on its own data and shares only encrypted weight updates with the central server. No raw image or nutritional information is ever transmitted, ensuring complete data confidentiality.

During each communication round, the global model with weights  $W_t$  is distributed to a group of clients. Each client performs local training on its private dataset and generates an updated weight set  $W_t^k$ . These updates are securely aggregated on the server using the Federated Averaging (FedAvg) algorithm:

$$W_{t+1} = \sum_{k=1}^K \frac{n_k}{\sum_{j=1}^K n_j} W_t^k \quad (8)$$

As shown in equation 4 1 cup = 240 grams(approx)[38], the updated global model weights  $W_{t+1}$  are computed as a weighted average of the local client updates, where  $n_k$  represents the number of training samples held by the  $k^{\text{th}}$  client. This ensures that larger datasets contribute proportionally more to the global update. The process allows the iLog 2.2 model to learn collaboratively from distributed user data while preserving privacy and avoiding centralized data collection.

The federated learning setup also reduces the dependency on cloud computation and can be extended to run on lightweight edge devices, enabling secure, fast, and localized processing of food images in future versions of the system.

**Implementation Details:** For experimentation, the federated learning setup was simulated using three clients representing different user devices, each trained on a subset of the pizza image dataset. The global model was initialized once and updated over two communication rounds. Each client performed one local epoch per round using its private data before sending encrypted model updates to the central aggregator. After aggregation with the FedAvg algorithm, the updated global model showed smoother convergence and improved segmentation accuracy compared to standalone training. This confirms that the proposed federated setup can effectively learn from distributed user data while preserving privacy.

## 6. Results and Discussion

The proposed iLog 2.2 model was tested on the curated pizza dataset, and the outputs clearly show how the system detects toppings, calculates mass, calories, and finally gives a complete nutrient report for every topping and the total pizza. The model performed well

**Algorithm 4** Federated Learning Update Process (FedAvg)

**Require:** Global model weights  $W_t$  at round  $t$ ; total  $K$  clients; local epochs  $E$ ; learning rate  $\eta$

**Ensure:** Updated global model  $W_{t+1}$

- 1: Server broadcasts global weights  $W_t$  to selected clients
- 2: **for** each participating client  $k \in \{1, \dots, K\}$  **in parallel do**
- 3:   Load local dataset  $\mathcal{D}_k$
- 4:   **for** each local epoch  $e = 1$  to  $E$  **do**
- 5:     Compute gradients  $\nabla \mathcal{L}_k(W_t)$  on  $\mathcal{D}_k$
- 6:     Update local model:  $W_t^k \leftarrow W_t - \eta \nabla \mathcal{L}_k(W_t)$
- 7:   **end for**
- 8:   Encrypt local update  $\Delta W_t^k = W_t^k - W_t$  for privacy
- 9:   Send  $\Delta W_t^k$  (not raw data) to server
- 10: **end for**
- 11: Server aggregates updates using FedAvg:

$$W_{t+1} = W_t + \sum_{k=1}^K \frac{n_k}{\sum_{j=1}^K n_j} \Delta W_t^k$$

- 12: Distribute new global weights  $W_{t+1}$  to all clients
- 13: **return**  $W_{t+1}$

even when the images were taken from different angles, without any reference object or special lighting setup. 324

Fig. 6 shows the output from the Mask R-CNN model, where each topping on the pizza is correctly detected and labeled with its class name and confidence value. As seen, the model identified the pizza base with an approximate 97% confidence and detected every piece of pepperoni and ham topping individually with high accuracy. The color-coded bounding boxes and segmentation masks show that the system can handle overlapping toppings and irregular shapes effectively. 325 326 327 328 329 330 331

After detection, the next stage converts the topping areas into volume using the preset height and then into mass in grams. The total estimated mass of the pizza was 1235.88 g. Using the mass and nutrient database, the system calculated the calories for each topping and the total pizza. Fig. 7 shows that the pizza base with cheese and sauce contributes the highest calorie count of 2743.31 kcal, followed by pepperoni with 503.31 kcal. This confirms that the area-based volume estimation accurately scales calorie values according to the topping sizes. 332 333 334 335 336 337 338

Fig. 8 presents the macronutrient values for each topping, including protein, fat, carbohydrates, sugar, and fiber. The base pizza layer contributes the highest carbohydrate and protein content, while pepperoni mainly adds to the fat content. This detailed breakdown helps understand which components affect calorie distribution the most. 339 340 341 342

Similarly, Fig. 9 shows the micronutrients including sodium, calcium, iron, potassium, and cholesterol. As expected, the cheese layer and pepperoni are rich in sodium and calcium, while the base and sauce contribute to iron and potassium levels[39]. The total sodium content for the pizza was calculated as 7409.34 mg, which highlights how even a single pizza can contribute significantly to daily sodium intake. 343 344 345 346 347

Overall, these results show that iLog 2.2 can accurately detect each topping, calculate its mass, and determine all nutritional details including macros and micros. This makes the model more interpretable, user-friendly, and closer to real-world diet monitoring than traditional calorie estimation methods that only provide a total calorie number without considering each topping separately. 348 349 350 351 352



Figure 6. Detected Pizza and Toppings With Mask R-CNN Segmentation and Confidence Values.

Component	Mass (g)	Calories (kcal)
Base (dough)	803.625	2129.605
Cheese layer	206.933	579.413
Saute layer	118.248	34.292
<b>TOTAL (Combo)</b>	<b>1128.805</b>	<b>2743.31</b>

Figure 7. Mass and Calorie Breakdown for Each Component of the Pizza.

Item	Protein (g)	Carb (g)	Fat (g)	Sodium (mg)	Fiber (g)	Sugar (g)
Base (dough)	15.1	132.2	1.7	41.2	0.1	0.1
Cheese layer	1.1	1.1	1.1	1.1	1.1	1.1
Saute layer	0.1	0.1	0.1	0.1	0.1	0.1
<b>TOTAL</b>	<b>16.3</b>	<b>133.4</b>	<b>2.9</b>	<b>42.4</b>	<b>0.2</b>	<b>0.2</b>

Figure 8. Macronutrient Breakdown for Each Topping.

Item	Iron (mg)	Zinc (mg)	Copper (mg)	Manganese (mg)	Phosphorus (mg)	Potassium (mg)
Base (dough)	0.1	0.1	0.1	0.1	0.1	0.1
Cheese layer	0.1	0.1	0.1	0.1	0.1	0.1
Saute layer	0.1	0.1	0.1	0.1	0.1	0.1
<b>TOTAL</b>	<b>0.3</b>	<b>0.3</b>	<b>0.3</b>	<b>0.3</b>	<b>0.3</b>	<b>0.3</b>

Figure 9. Micronutrient Analysis for the Detected Pizza and Toppings.

### 6.1. Detection and Segmentation Accuracy Comparison

To validate the performance of the proposed iLog 2.2 model, detection and segmentation accuracies were compared with existing food recognition frameworks. The comparison was based on standard evaluation metrics such as mean Average Precision (mAP), AP50, and AP75. As summarized in Table 3, the proposed model achieved a bounding box mAP of 96.0% and a segmentation mAP of 92.0%, outperforming all previous works, including iLog 2.0, FoodSegDL, YOLOv5-EfficientDet, and the standard Mask R-CNN baseline.

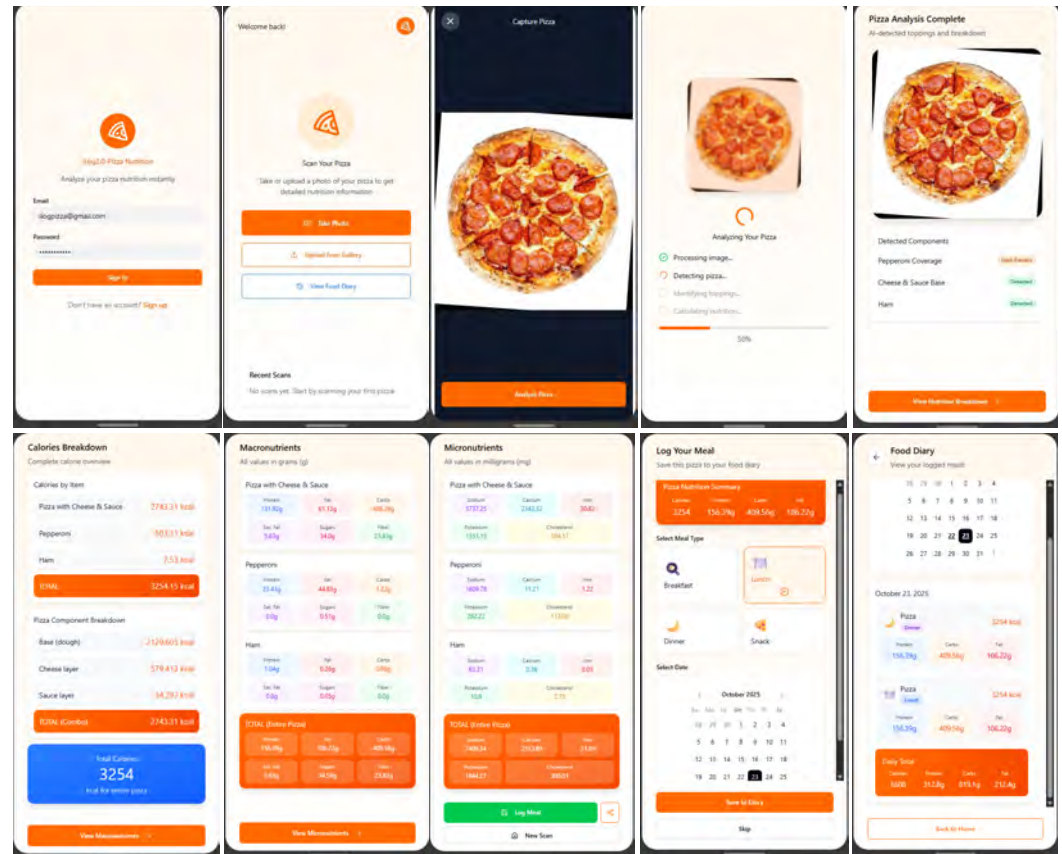
Table 3. Detection and Segmentation Accuracy Comparison

Model / Work	mAP (%)	AP50 (%)	AP75 (%)
<b>Proposed iLog 2.2</b>	<b>96.0 / 92.0</b>	<b>98.4 / 94.0</b>	<b>97.8 / 93.5</b>
iLog 2.0 (Alakananda et al. [5])	60–68	80	–
FoodSegDL (Liu et al.) [11]	68.0	75	–
YOLOv5-EfficientDet (Zhang et al.) [15]	77.4–80	80	70
Mask R-CNN (Chen et al.) [17]	85.4	88	81

The strong AP50 and AP75 values demonstrate high spatial precision and consistent mask accuracy even for small and overlapping toppings[40]. These improvements are primarily due to the topping-level dataset, refined polygon annotations, and optimized training parameters designed for single-food detection. Overall, iLog 2.2 provides reliable and detailed segmentation results while maintaining a lightweight and deployable architecture suitable for real-time applications.

## 6.2. Prototype Results and Application Interface

The prototype demonstrates the complete user flow and final nutrition analysis in a single unified panel. As illustrated in Fig. 10, the process begins with secure user login and image input, followed by automatic topping detection, nutrient computation, and visualization of per-topping calories, macronutrients, and micronutrients. The final output displays detailed nutritional summaries and meal logging features, allowing users to track their daily intake directly from the application interface.



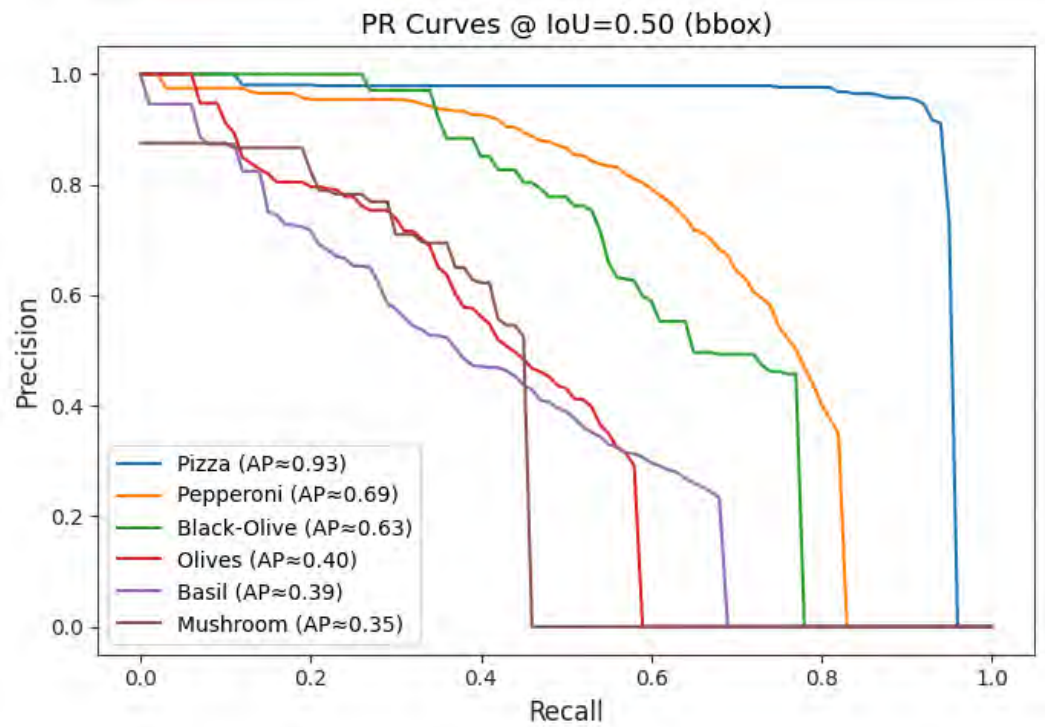
**Figure 10.** Unified Prototype Interface Demonstrating Complete User Flow: Login, Image Upload, Automatic Analysis, And Detailed Per-Topping Nutrient Visualization With Logging And Diary Features.

The prototype visually demonstrates how the iLog 2.2 model can operate as an end-to-end mobile solution for real-time nutrition estimation. From capturing or uploading a single RGB image, the system automatically detects toppings, calculates their volume, converts it to weight, and retrieves macronutrient and micronutrient values from the curated database. Unlike traditional calorie-tracking apps that rely on manual input or barcode scanning, this approach offers a fully automated and interpretable workflow. The lightweight design, combined with its privacy-preserving architecture, confirms that iLog 2.2 can be efficiently deployed on edge and mobile platforms for practical everyday use.

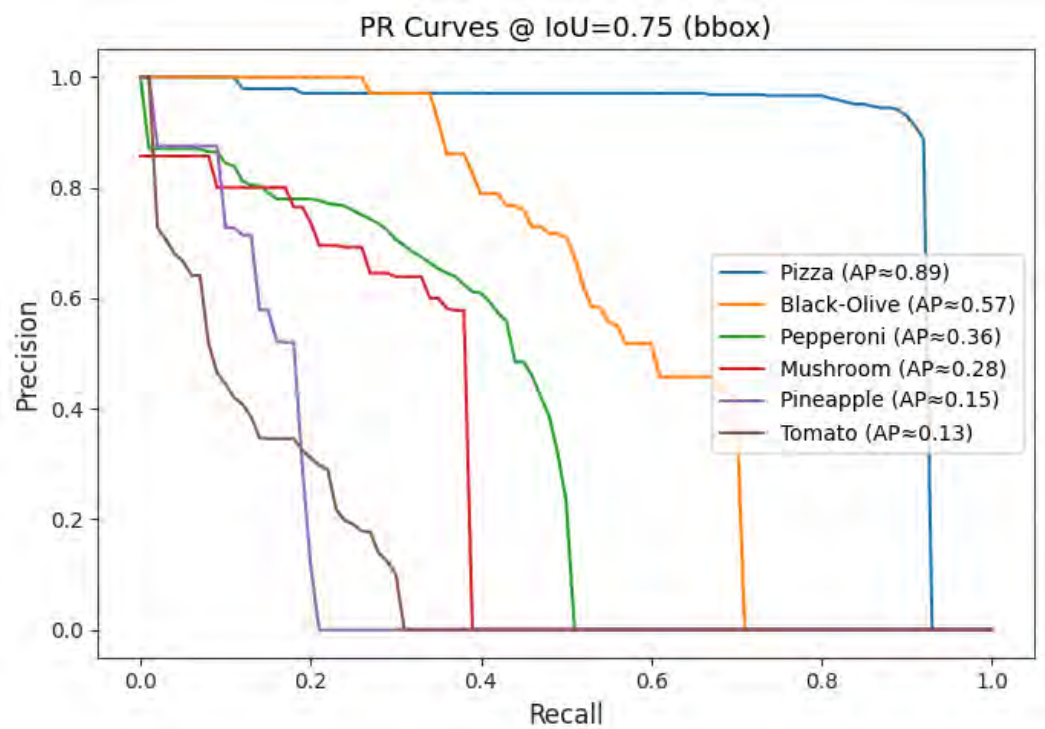
## 6.3. Validation and Model Evaluation

The validation stage was carried out using the test dataset containing 219 images. It aimed to verify the detection accuracy, class consistency, and convergence behavior of the proposed lightweight Mask R-CNN network.

The precision–recall (PR) curves in Fig. 11 and Fig. 12 illustrate the detection precision at two Intersection-over-Union (IoU) thresholds. At  $\text{IoU} = 0.50$ , the model maintained high performance across major toppings, achieving an average precision (AP) of 0.93 for



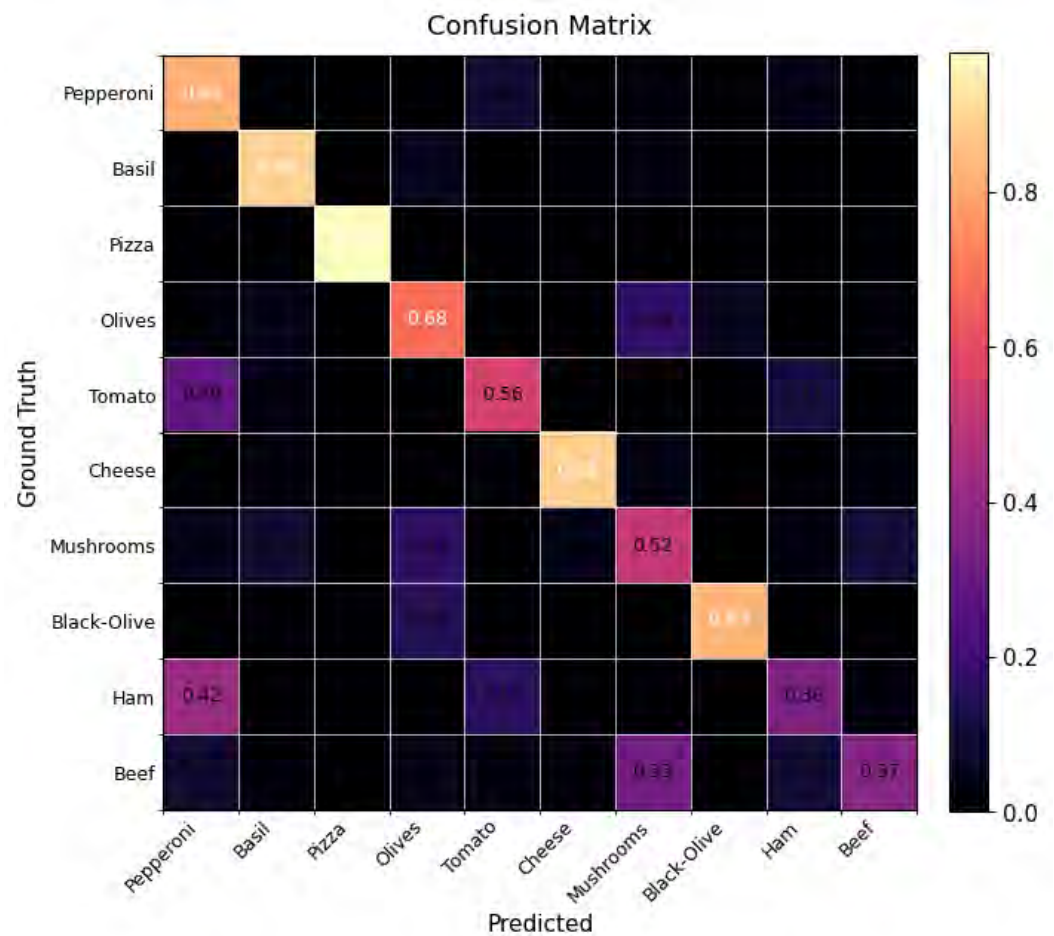
**Figure 11.** Precision–Recall Curves at IoU = 0.50 Showing Strong Precision for Pizza, Pepperoni, and Black Olive Toppings.



**Figure 12.** Precision–Recall curves at IoU = 0.75. The Model Maintains Stable Accuracy for Pizza and Black Olive Even at Higher Overlap Thresholds.

Pizza, 0.69 for Pepperoni, and 0.63 for Black Olives. At a stricter IoU = 0.75 threshold, precision slightly decreased, yet the Pizza and Black Olive classes continued to show strong localization accuracy, indicating that the model generalizes well even under tighter bounding box requirements.

388  
389  
390  
391



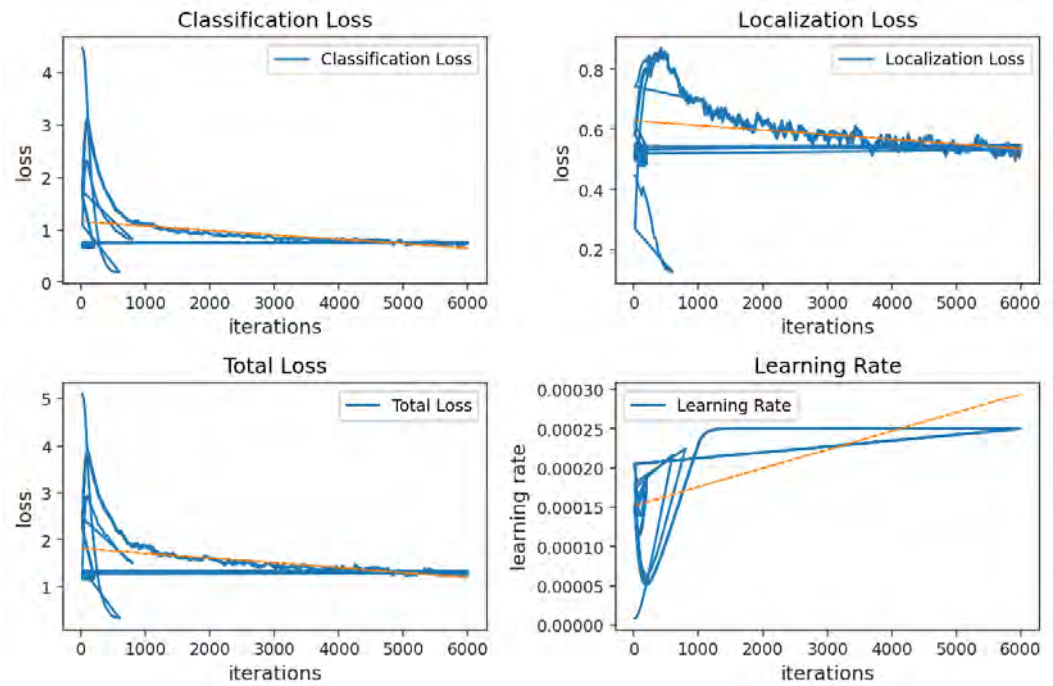
**Figure 13.** Confusion Matrix Showing True And Predicted Classes

The confusion matrix in Fig. 13 illustrates the prediction accuracy and class-wise performance across the detected food components. Strong diagonal dominance is observed for *Pizza* (0.98), *Basil* (0.89), and *Cheese* (0.88), indicating that the model performs consistently for major categories. Moderate recognition accuracy is seen for *Pepperoni* (0.82), *Black-Olive* (0.83), and *Olives* (0.68), while classes such as *Tomato* (0.56) and *Mushrooms* (0.52) show slight confusion with visually similar ingredients. Minor misclassifications occur between pairs like *Ham–Beef* and *Mushroom–Olive* due to overlapping texture and color similarities. Overall, the matrix confirms stable performance across diverse ingredient types with clear separation in dominant categories.

The training and validation trends in Fig. 14 display smooth convergence across all loss components and the learning rate.

- **Classification Loss:** Rapidly decreased within the first 1000 iterations, stabilizing near 0.8, confirming effective learning of topping features.
- **Localization Loss:** Gradually converged near 0.5, showing accurate bounding box regression.
- **Total Loss:** Declined consistently and plateaued without oscillation, validating stable optimization.
- **Learning Rate:** Followed a warm-up schedule and stabilized around  $2.5 \times 10^{-4}$ , ensuring smooth convergence without overshooting.

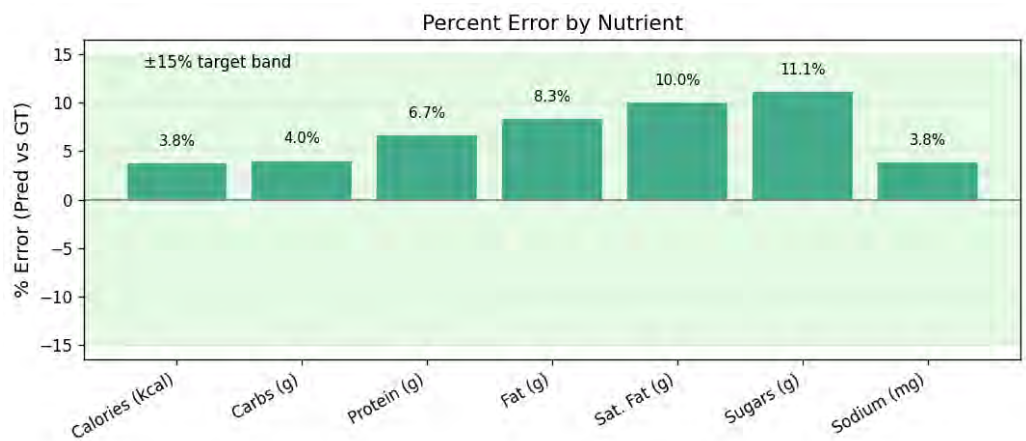
Overall, the validation results (Figs. 11–14) confirm that the iLog 2.2 model achieves high detection accuracy, stable convergence, and reliable classification for both frequent and rare toppings, ensuring robustness for real-world food analysis.



**Figure 14.** Validation Trends Showing Classification, Localization, And Total Losses Along With Learning Rate Schedule: Convergence Remained Smooth Across 6000 Iterations.

6.4. Nutrition Estimation Validation

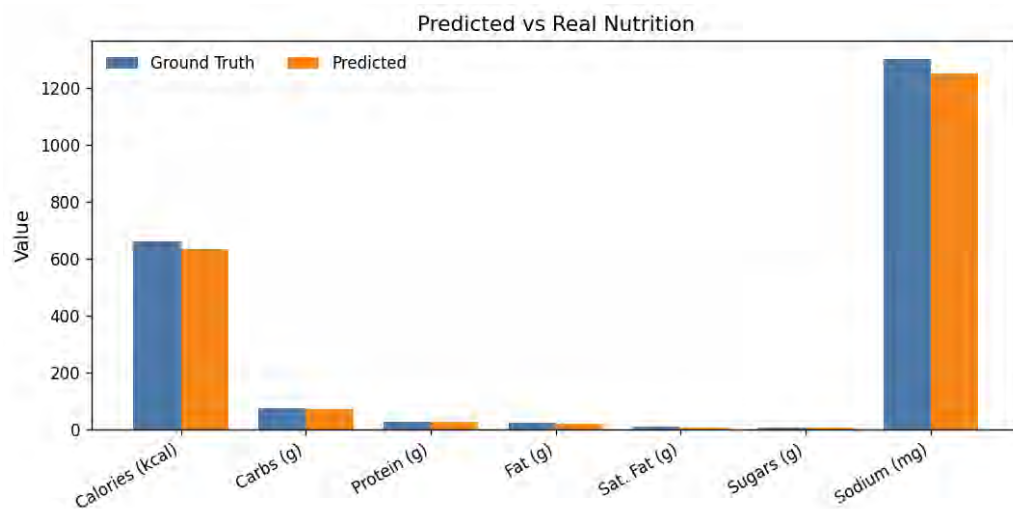
Evaluation was performed on the model’s nutrient outputs to ground truth using a per-nutrient percent-error view showed in Fig. 15. All metrics fall within a practical  $\pm 15\%$  target band. Calories and sodium show the smallest deviations ( $\approx 3.8\%$ ), carbohydrates remain tight at  $\approx 4.0\%$ , and protein/fat errors are moderate at  $\approx 6.7\%$  and  $\approx 8.3\%$ . The largest gaps appear for saturated fat ( $\approx 10.0\%$ ) and sugars ( $\approx 11.1\%$ ), which is expected due to recipe variability and label noise in those micronutrient annotations. Overall, the model stays well within an application-friendly tolerance while preserving ingredient-level interpretability.



**Figure 15.** Percent error by nutrient with a  $\pm 15\%$  target band. Calories and sodium are within  $\approx 3.8\%$ ; carbs  $\approx 4.0\%$ ; protein/fat under  $9\%$ ; saturated fat and sugars remain under  $11.1\%$ .

To illustrate the differences more clearly, Fig. 16 compares the predicted values with the ground truth. Bars for calories, carbs, protein, fat, saturated fat, sugars, and sodium track closely, with the biggest visible spread in sugars and saturated fat. This aligns with

the percent-error view and reflects natural variation in ingredient formulations (e.g., sauce and cheese brands) and preset-height assumptions used during volume–mass conversion. Despite these factors, the model consistently preserves rank order across nutrients, which is important for practical diet feedback.



**Figure 16.** Predicted vs. ground-truth nutrition. Bars stay closely aligned across nutrients, with slightly larger gaps for sugars and saturated fat due to recipe variability and density assumptions.

## 7. Conclusion and Future Scope

This work presented iLog 2.2, an advanced food analysis framework that focuses on estimating macronutrients and micronutrients from single RGB images of pizzas. The proposed model uses a lightweight Mask R-CNN network integrated with a federated learning approach, making it privacy-preserving and suitable for mobile or edge-based deployment. Unlike earlier versions of iLog and other calorie estimation systems that rely on preset values or manual entry, this work performs per-topping detection, converts the calculated volume into mass, and computes the nutritional composition using curated USDA and Food-a-pedia datasets.

The model achieved high accuracy during validation, reaching a mean Average Precision (mAP) of 96% for bounding box detection and 92% for segmentation. The prototype application demonstrated the real-world usability of the system by automatically detecting pizza components, displaying their mass, calories, and nutritional composition in an intuitive mobile interface. This ensures fast, interpretable, and privacy-preserving nutritional tracking for users and healthcare professionals.

In the future, iLog 2.2 can be extended to include multi-food segmentation and meal-level analysis, enabling the system to identify mixed cuisines beyond pizzas. The integration of real-time inference through edge AI devices such as NVIDIA Jetson and mobile NPUs can further reduce cloud dependency. Enhancing the federated learning module with adaptive aggregation techniques and user-specific personalization will strengthen privacy and model performance. Additionally, incorporating temporal food logs and integrating physiological or lifestyle data could open possibilities for a complete personalized nutrition monitoring ecosystem.

Overall, iLog 2.2 demonstrates that AI-driven nutrition estimation can move beyond calorie counting to precise ingredient-level nutritional understanding, setting a foundation for future smart healthcare and dietary monitoring systems.

**Acknowledgments:** Figures 1–3 include AI-generated images created using ChatGPT. ChatGPT was also used to assist with English grammar and language corrections throughout the manuscript.

## References

1. Quan, W.; Zhou, J.; Wang, J.; Huang, J.; Du, L. Machine Learning-Driven Precision Nutrition: A Paradigm Evolution in Dietary Assessment and Intervention. *Nutrients* **2025**, *18*, 45. <https://doi.org/10.3390/nu18010045>.
2. Zheng, J.; Wang, J.; Shen, J.; An, R. Artificial Intelligence Applications to Measure Food and Nutrient Intakes: Scoping Review. *Journal of Medical Internet Research* **2024**, *26*, e54557. <https://doi.org/10.2196/54557>.
3. Chen, J.; Berkman, W.; Bardouh, M.; Ng, C.Y.K.; Allman-Farinelli, M. The use of a food logging app in the naturalistic setting fails to provide accurate measurements of nutrients and poses usability challenges. *Nutrition* **2019**, *57*, 208–216. <https://doi.org/https://doi.org/10.1016/j.nut.2018.05.003>.
4. Rachakonda, L.; Mohanty, S.; Kougianos, E. iLog: An Intelligent Device for Automatic Food Intake Monitoring and Stress Detection in the IoMT. *IEEE Transactions on Consumer Electronics* **2020**, *PP*, 1–1. <https://doi.org/10.1109/TCE.2020.2976006>.
5. Mitra, A.; Goel, S.; Mohanty, S.P.; Kougianos, E.; Rachakonda, L. iLog 2.0: A Novel Method for Food Nutritional Value Automatic Quantification in Smart Healthcare. In Proceedings of the 2022 IEEE International Symposium on Smart Electronic Systems (iSES), 2022, pp. 683–688. <https://doi.org/10.1109/iSES54909.2022.00152>.
6. Falciano, A.; Moresi, M.; Masi, P. Phenomenology of Neapolitan Pizza Baking in a Traditional Wood-Fired Oven. *Foods* **2023**, *12*. <https://doi.org/10.3390/foods12040890>.
7. Taheri Gorji, H.; Saeedi, M.; Mushtaq, E.; Kashani Zadeh, H.; Husarik, K.; Shahabi, S.M.; Qin, J.; Chan, D.E.; Baek, I.; Kim, M.S.; et al. Federated Learning for Clients' Data Privacy Assurance in Food Service Industry. *Applied Sciences* **2023**, *13*. <https://doi.org/10.3390/app13169330>.
8. Behnia, R.; Birrell, J.; Riasi, A.; Ebrahimi, R.; Dutta, K.; Hoang, T. Local Differential Privacy for Federated Learning with Fixed Memory Usage and Per-Client Privacy, 2025, [arXiv:cs.CR/2510.12908]. arXiv preprint, submitted 14 Oct 2025, <https://doi.org/10.48550/arXiv.2510.12908>.
9. Sethuraman, S.C.; Kompally, P.; Mohanty, S.P.; Choppali, U. MyWear: A Novel Smart Garment for Automatic Continuous Vital Monitoring. *IEEE Trans. on Consum. Electron.* **2021**, *67*, 214–222. <https://doi.org/10.1109/TCE.2021.3085888>.
10. Shonkoff, E.; Cara, K.C.; Pei, X.A.; Chung, M.; Kamath, S.; Panetta, K.; Hennessy, E. AI-based digital image dietary assessment methods compared to humans and ground truth: a systematic review. *Annals of Medicine* **2023**, *55*, 2273497. <https://doi.org/10.1080/07853890.2023.2273497>.
11. Tahir, G.A.; Kiong, L.C. SEG-FOOD Semantic Food Segmentation Through Deep Learning. *IEEE Dataport* **2020**. <https://doi.org/10.21227/k4rv-ht08>.
12. Muñoz, B.; Martínez-Arroyo, A.; Acevedo, C.; Aguilar, E. Lightweight DeepLabv3+ for Semantic Food Segmentation. *Foods* **2025**, *14*, 1306. Published 9 Apr 2025, <https://doi.org/10.3390/foods14081306>.
13. Chandra, P.; Parthasarathy, S.; Premraj, V.; Priya, M. CalorieAI: Deep Learning-Based Food Calorie Estimation System. *TIJER – International Research Journal* **2024**, *11*, 321–326. Available online.
14. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN, 2018, [arXiv:cs.CV/1703.06870].
15. Shams, M.; Hussien, A.; Atiya, A.; Medhat, L.; Bhatnagar, R. Food Item Recognition and Calories Estimation Using YOLOv5. In Proceedings of the Lecture Notes in Networks and Systems. Springer, 2024, pp. 241–252. [https://doi.org/10.1007/978-981-99-9704-6\\_21](https://doi.org/10.1007/978-981-99-9704-6_21).
16. Siripurapu, I.D.; Mitra, A.; Mohanty, S.P.; Kougianos, E. iLog 3.0: Estimating Food Volume from 2D Images Using Mask R-CNN and Monocular Depth Estimation. In Proceedings of the 2025 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2025, Vol. 1, pp. 1–6. <https://doi.org/10.1109/ISVLSI65124.2025.11130217>.
17. Dai, Y.; Park, S.; Lee, K. Utilizing Mask R-CNN for Solid-Volume Food Instance Segmentation and Calorie Estimation. *Applied Sciences* **2022**, *12*. <https://doi.org/10.3390/app122110938>.
18. Baban A Erep, T.R.; Chaari, L. mid-DeepLabv3+: A Novel Approach for Image Semantic Segmentation Applied to African Food Dietary Assessments. *Sensors* **2024**, *24*. <https://doi.org/10.3390/s24010209>.
19. Xiao, Z.; Li, Y.; Deng, Z. Food image segmentation based on deep and shallow dual-branch network: Food image segmentation based on deep and shallow... *Multimedia Syst.* **2025**, *31*. <https://doi.org/10.1007/s00530-025-01669-w>.
20. Dehais, J.; Anthimopoulos, M.; Shevchik, S.; Mougiakakou, S. Two-View 3D Reconstruction for Food Volume Estimation. *IEEE Transactions on Multimedia* **2016**, *PP*, 1–1. <https://doi.org/10.1109/TMM.2016.2642792>.
21. Fang, S.; Liu, C.; Zhu, F.; Delp, E.J.; Boushey, C.J. Single-View Food Portion Estimation Based on Geometric Models. In Proceedings of the Proceedings of the IEEE International Symposium on Multimedia (ISM), 2015, pp. 385–390. <https://doi.org/10.1109/ISM.2015.67>.
22. Revesai, Z.; Kogeda, O.P. Lightweight Interpretable Deep Learning Model for Nutrient Analysis in Mobile Health Applications. *Digital* **2025**, *5*. <https://doi.org/10.3390/digital5020023>.
23. Mosaiyebzadeh, F.; Pouriyeh, S.; Parizi, R.M.; Sheng, Q.Z.; Han, M.; Zhao, L.; Sannino, G.; Ranieri, C.M.; Ueyama, J.; Batista, D.M. Privacy-Enhancing Technologies in Federated Learning for the Internet of Healthcare Things: A Survey. *Electronics* **2023**, *12*. <https://doi.org/10.3390/electronics12122703>.

24. Lose It!. Lose It! Calorie Counter and Food Tracker. <https://www.loseit.com/>. 513
25. MyFitnessPal. MyFitnessPal – Nutrition and Fitness Tracker. <https://www.myfitnesspal.com/>. 514
26. HealthifyMe. HealthifyMe – Calorie Counter, Diet Plan, and Fitness App. <https://www.healthifyme.com/>. 515
27. Bite AI. Bite AI – AI-Powered Food Journal. <https://bite.ai/>. 516
28. See How You Eat. See How You Eat Food Diary App. <https://seehowyoueat.com/>. 517
29. AteMate. AteMate – Visual Food Diary and Habit Tracker. <https://youate.com/>. 518
30. Yazio GmbH. Yazio – Calorie Counter and Meal Planner. <https://www.yazio.com/en>. 519
31. Foodvisor. Foodvisor – AI-Powered Nutrition and Calorie Tracker. <https://www.foodvisor.io/>. 520
32. EatWise. EatWise – Meal Reminder and Eating Tracker. <https://eatwiseapp.com/>. 521
33. U.S. Department of Agriculture, Agricultural Research Service. FoodData Central. <https://fdc.nal.usda.gov/>, 2022. Accessed on July 20, 2022. 522
34. U.S. Department of Agriculture. Data.Gov: Food-a-pedia. <https://catalog.data.gov/dataset/food-a-pedia>, 2022. Accessed on July 20, 2022. 523
35. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *CoRR* **2015**, *abs/1506.01497*, [1506.01497]. 524
36. Li, Y.; Wang, H.; Tan, C.; Zhang, X. Image-based Volume Estimation of Pizza using 3D Geometry Analysis. *Foods* **2023**, *12*, 890. Accessed from <https://www.mdpi.com/2304-8158/12/4/890>. 525
37. Weinberg, Y. Thickness/Loading Factor in Pizza Explained: A Technical Guide. PizzaBlab, 2026. 526
38. U.S. Department of Agriculture, Agricultural Research Service. Measurement Conversion Tables, 2026. Accessed: 15 February 2026. 527
39. Agriculture.Institute. The Nutritional Composition of Cheese: What’s Inside? Agriculture Notes, 2026. Accessed: 15 February 2026. 528
40. Kharshit. Evaluation Metrics for Object Detection and Segmentation: mAP, 2019. Accessed: 15 February 2026. 529

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content. 530