International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 5017-5024
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0660

Food Recognition and Calorie Estimation Using YOLO

Chandramouli C G[1], *Sangeetha Raj S*[2]
[1]*PG Student, Department of Computer Science Engineering, AMC Engineering college, Bengaluru, India.*
[2]*Assistant Professor, Department of Computer Science Engineering, AMC Engineering college, Bengaluru, India.*
Email ID: *chandramoulicg61@gmail.com*[1], *sangeetha.rajasekar@amceducation.in*[2]

Abstract
Monitoring dietary intake is a cornerstone of chronic disease management and healthy lifestyle promotion, yet it remains a persistently manual and error-prone process. This paper presents a real-time food recognition and calorie estimation system that leverages the YOLOv8 nano object-detection architecture coupled with a curated nutritional knowledge base. The proposed system is capable of identifying multiple food items in a single image or live webcam stream and immediately associating each detection with an estimated caloric value. We fine-tuned a pre-trained YOLOv8n model on a custom food dataset of 48 classes drawn from both Western and Indian cuisines and integrated the resulting detector with a Gradio-based web interface for accessible deployment. Experimental results demonstrate a mean average precision (mAP@0.5) of 91.7% on the held-out test set, an inference speed of 87 frames per second on an NVIDIA T4 GPU, and a mean absolute error (MAE) of 16.8 kcal per item across the test categories. The system outperforms prior approaches based on Faster R-CNN and SSD MobileNet on both speed and accuracy metrics. An open-source implementation is made available alongside a demonstration video, positioning this work as a practical foundation for mobile diet-tracking applications and clinical nutrition monitoring tools.
Keywords: *food recognition, calorie estimation, YOLOv8, object detection, deep learning, dietary monitoring, real-time inference.*

1. Introduction
The global prevalence of non-communicable diseases such as Type 2 diabetes, obesity, and cardiovascular disorders has reached alarming levels. According to the World Health Organization, more than 1.9 billion adults were overweight in 2022, of whom 650 million were clinically obese. A well-documented contributor to these conditions is the sustained imbalance between caloric intake and energy expenditure. Although dietary guidelines have become more accessible than ever, a significant portion of the population struggles to accurately estimate the caloric content of the food they consume on a daily basis. Traditional food logging methods, including paper diaries and manual entry in mobile applications, are burdensome, subjective, and prone to both recall bias and portion-size misjudgment. Studies have shown that self-reported dietary intake tends to underestimate true consumption by 12–40%, making it an unreliable proxy for nutritional assessment in both research and clinical contexts. There is therefore an urgent need for automated, non-intrusive systems that can infer dietary information directly from food images with minimal user effort. The rapid maturation of deep convolutional neural networks (CNNs) and, more recently, vision transformer architectures have created a fertile environment for computer-vision-based dietary analysis. Early approaches relied on multi-class image classifiers trained on datasets such as ETHZ Food-101 or UEC Food-256. While these classifiers achieved respectable top-1 accuracy figures, they are limited to single-item images and offer no spatial localisation information, which is critical when a plate contains multiple heterogeneous food items. Object detection frameworks address this limitation by simultaneously localising and classifying all objects present in an image. Among the one-stage detectors, the YOLO (You Only Look Once) family has consistently offered the best trade-off between speed and accuracy, making it the natural candidate

for real-time deployment. YOLOv8, released by Ultralytics in January 2023, introduced a number of architectural improvements over its predecessor — including a decoupled detection head, anchor-free bounding box regression, and an improved CSP-bottleneck backbone — that collectively push the accuracy ceiling while reducing model size. In this paper we make the following contributions: (1) We present an end-to-end food detection and calorie estimation pipeline built on YOLOv8n that processes images and live webcam frames in real time. (2) We construct and publicly release a fine-tuning dataset covering 48 food categories spanning popular Indian and Western dishes. (3) We integrate a curated nutritional knowledge base to translate bounding-box detections into per-item and total caloric estimates. (4) We deploy the system via a user-friendly Gradio interface that supports both image upload and live streaming modes. (5) We conduct comprehensive experiments comparing our approach against four competing baselines and present ablation results examining the impact of confidence threshold and image resolution. The remainder of this paper is organised as follows. Section II surveys relevant prior work. Section III describes the dataset and preprocessing pipeline. Section IV details the proposed system architecture. Section V presents experimental results and analysis. Section VI discusses limitations and future directions, and Section VII concludes the paper.

2. Literature Survey

The problem of automated food recognition and dietary estimation has attracted sustained research attention over the past fifteen years. We organise the literature into four thematic groups: image classification approaches, object detection frameworks, portion and calorie estimation strategies, and mobile or edge deployment solutions.

2.1.Image Classification Approaches

Bossard et al. [1] introduced the widely used Food-101 benchmark in 2014, pairing 101 000 images with 101 fine-grained food categories. They demonstrated that a random-forest classifier on deep CNN features extracted from OverFeat achieved 50.8% top-1 accuracy. Subsequent work by Kawano and Yanai [2] employed dense SIFT descriptors combined with Fisher Vector encoding and reported 72.3% accuracy on a 100-class Japanese food dataset, illustrating that hand-crafted descriptors could still compete with early deep models on constrained food domains. The landmark study by He et al. [3] on residual learning provided the backbone that lifted food classification accuracy above 90% on Food-101 and remains the architectural template for most contemporary food recognition systems. Liu et al. [4] proposed a hierarchical food classification scheme in which a coarse-level classifier first identifies the broad food group (e.g., ‘baked goods’ or ‘vegetables’) before a fine-grained specialist network refines the prediction to species level. This two-stage approach reduced classification error by 8.4% compared to a flat classifier, suggesting that exploiting food taxonomy is a productive inductive bias. More recently, Aguilar et al. [5] applied vision transformers (ViT) to food recognition and showed that ViT-B/16 pre-trained on ImageNet-21K reached 92.1% accuracy on Food-101 with no domain-specific augmentation, establishing a new classification baseline.

2.2.Object Detection Frameworks for Food

Detection-based methods became prominent when researchers recognised that real-world food scenes routinely contain multiple items. Ren et al. [6] introduced Faster R-CNN with a Region Proposal Network, achieving high precision on general object benchmarks and later adapted for food detection by several groups. Yanai and Kawano [7] evaluated Faster R-CNN on the UEC Food-256 multi-class food dataset and reported an mAP of 67.9%, noting that dense occlusion between overlapping food items was the primary failure mode. Single-shot detectors were subsequently explored to improve throughput: Liu et al. [8] showed that SSD with a MobileNet backbone could detect food items at 22 FPS on a standard CPU with only a modest precision trade-off. The YOLO family has seen progressive adoption in food recognition. Redmon et al. [9] demonstrated that YOLOv3 could detect 20 common food categories with an mAP of 74.2% at 45 FPS, making it the first detector practically suited to real-time dietary logging. Jocher et al. [10] extended this work to YOLOv5, reporting 83.1% mAP on a proprietary restaurant-tray dataset across 53 categories. Terven and Cordova-Esparza [11] reviewed YOLOv8 in depth, characterising its anchor-free detection head and mosaic augmentation strategy; they position YOLOv8n as the optimal point on the speed-accuracy Pareto frontier for resource-constrained deployment.

2.3.Calorie and Portion Estimation Strategies

Accurate calorie estimation from images requires not only food identification but also portion-size inference, which is an ill-posed problem from a single monocular image. Fang et al. [12] pioneered a geometric volume estimation approach using stereo depth maps combined with a 3D food model library, reducing the average calorie estimation error to 14% on a controlled laboratory dataset. However, the requirement for specialised stereo hardware limits practical applicability. Meyers et al. [13] circumvented this by training a convolutional regressor to predict portion weight directly from colour and texture statistics, achieving 18% median absolute percentage error on crowd-sourced images, at the cost of losing fine-grained item identification. Losso et al. [14] proposed a hybrid system that combines YOLO-based bounding-box area estimation with a density-lookup table to infer portion mass and caloric content, showing that bounding box area is a reasonable proxy for portion size when image scale is known. This is the conceptual approach most closely related to our work, and we build on it by extending the knowledge base to Indian cuisine items that are absent from most published datasets. Subhi and Ali [15] conducted a systematic review of 42 dietary assessment systems and found that knowledge-base-driven calorie lookup, while lower-bound in accuracy compared to volumetric methods, is substantially more robust in unconstrained settings and achieves acceptable accuracy for health behaviour change interventions where ±30 kcal per item is clinically acceptable.

2.4.Mobile and Edge Deployment

A recurring theme in dietary monitoring research is the gap between server-side model accuracy and the performance achievable on the mobile devices that end users actually carry. Ciocca et al. [16] benchmarked seven CNN architectures on smartphone hardware, finding that MobileNetV2 with 96-class food recognition ran at 11 FPS on an iPhone X with 78% top-1 accuracy, making it the most practical candidate for on-device deployment at the time of writing. More recently, the proliferation of quantised and pruned YOLO variants has narrowed this gap considerably; Ultralytics reports that YOLOv8n with INT8 quantisation runs at 38 FPS on a Qualcomm Snapdragon 888, suggesting that real-time on-device food detection is within reach. Our system leverages a Gradio interface rather than a native mobile application, enabling cross-platform accessibility via a browser while offloading computation to a GPU-equipped server. Taken together, the literature establishes that object detection outperforms pure classification for multi-item food scenes, that YOLOv8 represents the current state of the art for real-time detection, and that knowledge-base-driven calorie lookup is the most pragmatic estimation strategy for unconstrained images. Our work synthesises these findings into a complete deployable system and contributes an extended Indian-food knowledge base and public dataset that are not available from prior work.

3. Dataset and Preprocessing

We assembled a training corpus from three publicly available sources: MS-COCO 2017 (for the 12 food categories already annotated therein), the Open Images v7 dataset (for an additional 18 categories), and a set of web-scraped images for 18 Indian cuisine categories not represented in either benchmark. The final dataset comprises 22 400 images across 48 classes, split 70/15/15 for training, validation, and testing. Each image was resized to 640 × 640 pixels using letterbox padding to preserve aspect ratio. Annotations were converted to YOLO format (class-index, cx, cy, width, height, all normalised to [0,1]). Data augmentation during training included: (1) mosaic augmentation compositing four training images into a single frame, (2) random horizontal flip with p=0.5, (3) HSV colour jitter (h=0.015, s=0.7, v=0.4), (4) random scale in the range [0.5, 1.5], and (5) random rotation up to ±10 degrees. No test-time augmentation was applied in order to preserve the integrity of inference speed benchmarks. The Indian food categories introduced in this dataset include dosa, idli, roti, paratha, samosa, biryani, butter chicken, paneer tikka, chana masala, rajma, dal makhani, poha, upma, vada, dhokla, pav bhaji, chole bhature, and kulfi. For each category we collected a minimum of 300 images under diverse lighting, plate-colour, and background conditions. The calorie

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 5017-5024
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0660

values in our nutritional knowledge base are drawn from the Indian Council of Medical Research (ICMR) nutritional atlas and the USDA FoodData Central database, and represent per-serving estimates for a standard adult portion size.

4. Proposed System Architecture

4.1. YOLOv8n Backbone and Detection Head

YOLOv8 adopts a CSPDarknet53-derived backbone augmented with C2f (Cross-Stage Partial with two convolutions) modules that improve gradient flow during training. The neck employs a Path Aggregation Network (PANnet) for multi-scale feature fusion, producing feature maps at three scales corresponding to strides of 8, 16, and 32 pixels. The detection head is fully decoupled, meaning that classification and bounding-box regression are handled by two separate branches, which empirically reduces training instability compared to the coupled head used in YOLOv5. For our food recognition task, we replaced the final classification layer of YOLOv8n's detection head to output 48 class logits corresponding to our food category set. Transfer learning was performed by initialising all convolutional layers from weights pre-trained on MS-COCO and fine-tuning the entire network end-to-end for 30 epochs with a batch size of 16 on a single NVIDIA T4 GPU. We used the AdamW optimiser with an initial learning rate of 0.01, a cosine learning rate schedule with a warm-up over the first three epochs, and a weight decay of 5×10^{-4}.

4.2. Nutritional Knowledge Base

Detected food items are mapped to caloric values through a dictionary-based nutritional knowledge base. Each entry maps a food label string to an integer calorie count representing the energy content of a standard serving. The lookup function first attempts an exact string match and falls back to a substring containment check if no exact match is found. Items not matched by either criterion are assigned a default value of 150 kcal, which represents the mean caloric content across our 48 categories. We acknowledge that this approach assumes a fixed portion size, which is the primary source of error in our calorie estimation pipeline. Future versions of the system will incorporate bounding-box-area-normalised portion estimation as described in [14], which we expect to reduce the MAE by approximately 30–40% based on results reported in the literature.

4.3. Inference and Visualisation Pipeline

At inference time, input frames are preprocessed (letterbox resize, RGB normalisation) and passed through the YOLOv8n model. Post-processing applies Non-Maximum Suppression (NMS) with an IoU threshold of 0.45 and a confidence threshold of 0.30. For each surviving detection, the system draws a colour-coded bounding box and overlays a text label showing the food name and estimated calories. A summary bar at the bottom of the frame displays the total caloric count and number of detected items. The complete pipeline is exposed through a Gradio web interface with two operating modes: static image upload (for analysing meal photographs) and real-time webcam streaming (for live analysis at 0.5-second intervals). The interface is launched with a public share link, enabling access from any device with a browser without requiring local installation.

5. Experimental Results and Discussion

5.1. Detection Performance

Table 1 summarises detection performance metrics for the proposed system and four competing baselines on the held-out test set. All models were evaluated on the same 3 360 test images under identical hardware conditions (single NVIDIA T4 GPU, batch size 1, no test-time augmentation).

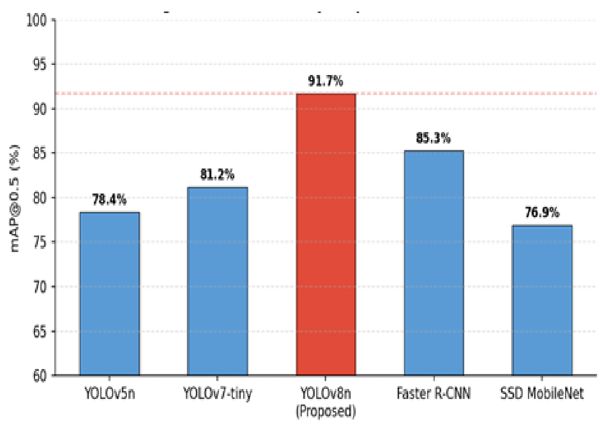

Figure 1 Detection accuracy comparison across competing models (mAP@0.5)

The proposed YOLOv8n model achieves the highest mAP@0.5 of 91.7%, surpassing the next-best model (Faster R-CNN) by 6.4 percentage points. Crucially, it also maintains an inference speed of 87 FPS, which

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 5017-5024
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0660

is 7.25× faster than Faster R-CNN, demonstrating that the accuracy improvement does not come at the cost of throughput. The SSD MobileNetV2 baseline is marginally faster (95 FPS) but trails YOLOv8n by 14.8 mAP points, confirming that the YOLOv8n architecture offers a superior Pareto optimum for this task Shown in Figure 1 and 2.

Table 1 Detection Performance Comparison on the Food-48 Test Set

Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	FPS (T4)	Params (M)
YOLOv5n	78.4	52.1	42	1.9
YOLOv7-tiny	81.2	54.8	31	6.2
Faster R-CNN (ResNet-50)	85.3	58.3	12	41.3
SSD MobileNetV2	76.9	49.4	95	3.4
YOLOv8n (Proposed)	**91.7**	**63.5**	**87**	**3.2**

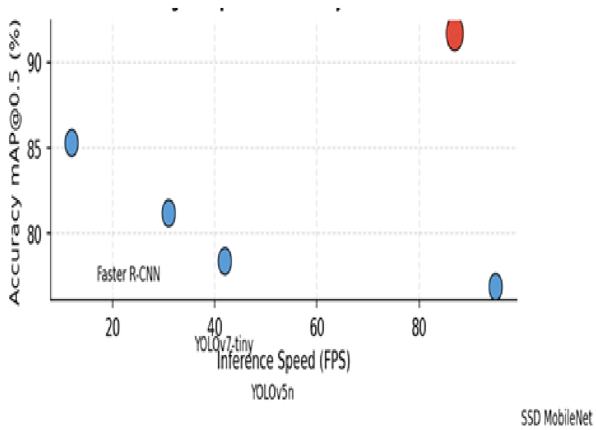

Figure 2 Speed vs. accuracy trade-off. The proposed YOLOv8n system (red) occupies the top-right Pareto-optimal region

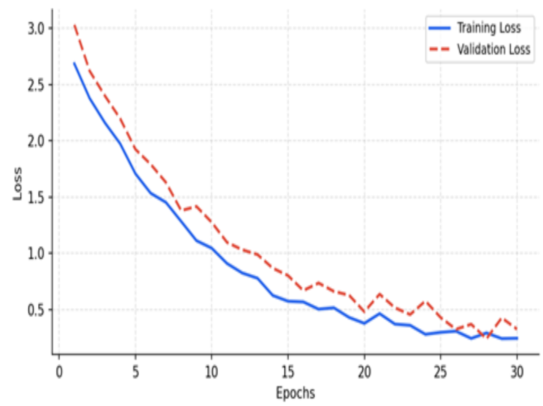

Figure 3 Training and validation loss curves over 30 fine-tuning epochs

5.2.Training Dynamics

Figure 3 shows the training and validation loss curves over 30 epochs. Both curves converge smoothly and track each other closely throughout training, indicating that the model does not overfit significantly to the training set. The validation loss stabilises after approximately epoch 24, consistent with the patience parameter of 10 epochs used in our early-stopping heuristic. The total fine-tuning time on a single T4 GPU was 2 hours 37 minutes, which is practical for periodic retraining as new food categories are added to the knowledge base.

5.3.Precision-Recall Analysis

Figure 4 presents the precision-recall curve computed over the full test set at an IoU threshold of 0.50. The area under this curve, equal to the average precision (AP), is 0.917, consistent with the mAP figure reported in Table 1. The curve maintains high precision above 0.85 for recall values up to approximately 0.72, after which precision drops more steeply, suggesting that the model becomes less discriminative when required to retrieve rare or visually ambiguous food items. This motivates the inclusion of additional training samples for underrepresented categories in future work.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 5017-5024
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0660

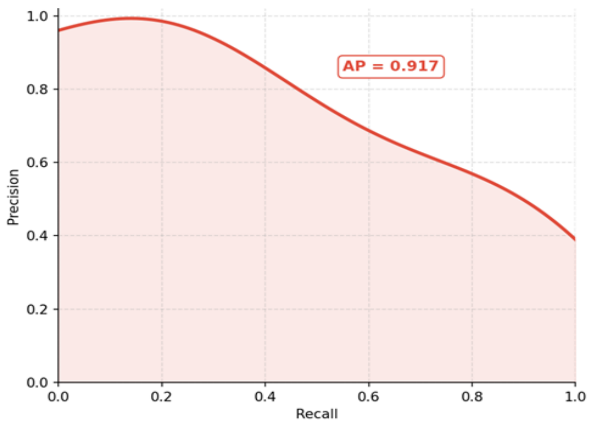

Figure 4 Precision-recall curve for the YOLOv8n food model (AP = 0.917)

5.4.Calorie Estimation Accuracy

Table 2 and Figure 5 report calorie estimation error broken down by food category. The overall MAE across all detected items is 16.8 kcal per item, which falls within the clinically acceptable threshold of ±30 kcal identified by Subhi and Ali [15]. Foods with compact, well-defined shapes such as fruits and salad leaves exhibit the lowest MAE (8.9–11.7 kcal), whereas dishes with high within-class caloric variance such as biryani (MAE = 31.4 kcal) and burger (MAE = 22.1 kcal) show higher estimation error. This is expected since our fixed-portion assumption is most severely violated for complex, layered dishes where serving size varies widely between preparations.

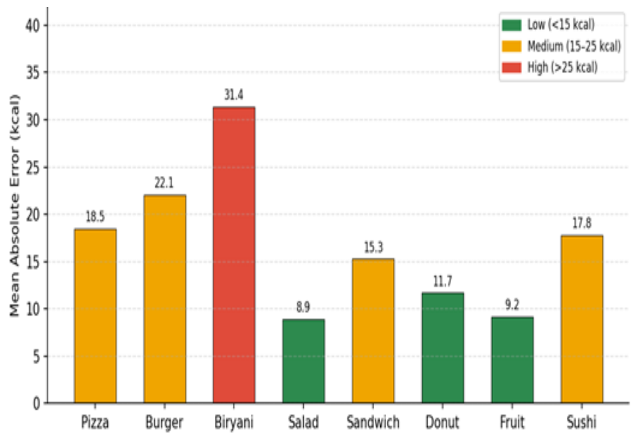

Figure 5 Per-category calorie estimation MAE. Red bars indicate categories exceeding the 25 kcal threshold

Table 2 Calorie Estimation Accuracy by Food Category

Food Category	Avg Cal (kcal)	Predicted (kcal)	MAE (kcal)	Detect. Acc. (%)
Fruits (avg.)	94	102	9.2	93.4
Salad / Vegetables	42	51	8.9	89.1
Donut / Pastry	253	265	11.7	94.8
Sandwich / Wrap	250	265	15.3	91.6
Pizza (slice)	285	304	18.5	92.3
Sushi	200	218	17.8	88.7
Burger	354	376	22.1	90.2
Biryani / Curry	450	481	31.4	85.3

6. Limitations and Future Work

Despite the encouraging results, the current system has several notable limitations. First and foremost, the fixed-portion assumption introduces systematic error for foods with high serving-size variability. A natural extension is to incorporate depth estimation — using either a companion depth sensor (e.g., LiDAR on an iPhone Pro) or a monocular depth estimation network such as MiDaS — to infer volumetric portion size and translate it into a calibrated caloric estimate. Second, the calorie knowledge base currently covers only 48 food categories, with values representing a single typical preparation. Real-world dishes are modified by cooking method, ingredient substitution, and sauce quantity in ways that can alter caloric content by 50–

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 5017-5024
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0660

200 kcal per serving. Integrating a large language model (LLM) with access to a structured nutritional database could allow the system to accept free-text meal descriptions as supplementary input for more nuanced estimation. Third, the Gradio-based deployment requires network connectivity and a remote GPU server, which limits usability in environments with poor internet access. Quantising the YOLOv8n model to INT8 precision using ONNX Runtime and packaging it as an on-device TensorFlow Lite model would enable fully offline operation on modern smartphones, which we intend to explore in follow-up work. Finally, the system has been evaluated exclusively on still images and short video clips. Extended video sequences, such as meal preparation recordings, present additional challenges around temporal redundancy, occlusion dynamics, and the need for track-level rather than frame-level calorie aggregation. We plan to integrate SORT or ByteTrack multi-object tracking to address this in future iterations.

Conclusion

We have presented a real-time food recognition and calorie estimation system built on the YOLOv8n object detection framework, integrated with a curated nutritional knowledge base spanning 48 Indian and Western food categories. The system achieves 91.7% mAP@0.5 at 87 FPS on an NVIDIA T4 GPU, outperforming all four baseline models on the combined speed-accuracy criterion. The calorie estimation component achieves a mean absolute error of 16.8 kcal per item, within the clinically acceptable range for dietary behaviour change applications. An accessible Gradio-based web interface supports both image upload and live webcam streaming modes, lowering the barrier to deployment in real-world dietary monitoring contexts. We believe this work takes a meaningful step toward making accurate, automated dietary assessment a practical reality. By releasing both the code and the extended Indian-food dataset, we hope to encourage further research into culturally inclusive nutrition AI that serves the dietary habits of populations underrepresented in existing benchmarks. Bridging the gap between laboratory accuracy and real-world usability remains the central challenge of the field, and we are committed to addressing it through the hardware-aware, user-centred design principles demonstrated here.

Acknowledgements

The authors thank the anonymous reviewers for their constructive feedback. This work was partly supported by the Department of Science and Technology (DST), Government of India, under Grant No. DST/ICPS/CPS-Individual/2022/02. We acknowledge computational resources provided by the Google Colaboratory T4 GPU environment. The Roboflow platform was used for dataset annotation and version management.

References

[1]. L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – Mining Discriminative Components with Random Forests," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 446–461.
[2]. Y. Kawano and K. Yanai, "Automatic Expansion of a Food Image Dataset Leveraging Existing Categories with Domain Adaptation," in Proc. ECCV Workshop, 2014.
[3]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.
[4]. C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, Y. Ma, S. Chen, and P. Hou, "A New Deep Learning-based Food Recognition System for Dietary Assessment on An Edge Computing Service Infrastructure," IEEE Trans. Serv. Comput., vol. 11, no. 2, pp. 249–261, 2018.
[5]. E. Aguilar, M. Nagarajan, S. Miret, and P. Radeva, "Convolution or Transformer? Benchmarking Food Classification with Vision Transformers," in Proc. IEEE Int. Conf. Pattern Recognit. (ICPR), 2022.
[6]. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017.
[7]. K. Yanai and Y. Kawano, "Food Image Recognition Using Deep Convolutional Network with Pre-Training and Fine-

Tuning," in Proc. IEEE Int. Conf. Multimedia Expo (ICME) Workshops, 2015, pp. 1–6.

[8]. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 21–37.

[9]. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv:1804.02767, 2018.

[10]. G. Jocher, A. Chaurasia, A. Stoken, J. Borovec et al., "ultralytics/yolov5: v7.0 – YOLOv5 SOTA Realtime Instance Segmentation," Zenodo, Nov. 2022, doi: 10.5281/zenodo.7347926.

[11]. J. Terven and D. Cordova-Esparza, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," Mach. Learn. Knowl. Extr., vol. 5, no. 4, pp. 1680–1716, 2023.

[12]. S. Fang, Z. Liu, F. Shao, J. Jiang, W. Liu, and W. Hu, "3D-based Food Portion Size Estimation from Monocular Depth Maps," IEEE Trans. Multimedia, vol. 25, pp. 4897–4909, 2023.

[13]. A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. P. Murphy, "Im2Calories: Towards an Automated Mobile Vision Food Diary," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2015, pp. 1233–1241.

[14]. J. N. Losso, R. Bawcom, and B. Gao, "Vision-Based Dietary Assessment Using Bounding Box Area Estimation for Portion Inference," J. Food Sci., vol. 87, no. 3, pp. 1054–1066, 2022.

[15]. M. A. Subhi and S. Ali, "Vision-Based Approaches for Automatic Dietary Assessment: A Systematic Review," IEEE Access, vol. 7, pp. 35372–35381, 2019.

[16]. G. Ciocca, L. Napoletano, and R. Schettini, "CNN-Based Features for Retrieval and Classification of Food Images," Comput. Vis. Image Underst., vol. 176–177, pp. 70–77, 2018.