Lightweight Food Localization and Recognition via Multi-Branch Feature Learning and Enhanced Aggregation

Xiangyi Zhu, Yancun Yang, Pindan Cao, Guorui Sheng, and Bruce Denby, *Senior Member, IEEE*

***Abstract*— Food image localization and recognition on edge devices is a core task in food computing, enabling convenient dietary monitoring and efficient health management. However, food localization and recognition presents significant challenges due to inherent intra-class variability, inter-class similarity, and non-rigid characteristics. To address these challenges, we propose YOLO-Multi Feature Fusion, a novel multi-feature fusion model for food image localization and recognition. Building upon the YOLOv5 framework, YOLO-Multi Feature Fusion integrates several key components: the Ghost Bottleneck from the lightweight GhostNet, a newly designed Multi-Scale Feature Bottleneck , a Bidirectional Vision Transformer, and an Information Cross-Exchange module. These modules enable the model to comprehensively capture and fuse complex feature information from food images while simultaneously reducing both model parameters and computational load. Extensive evaluations on benchmark datasets (UEC Food100, UEC Food256, and ZSFooD) demonstrate that YOLO-Multi Feature Fusion outperforms existing lightweight detectors. Compared to YOLOv5, YOLO-Multi Feature Fusion achieves mAP improvements of 3.0%, 3.0%, and 0.3% on these datasets, respectively, with parameter reductions of 5.7M, 4.7M, and 4.9M, and computational load reductions of 44.6 GFLOPs, 42.0 GFLOPs, and 42.0 GFLOPs. The source code will be released upon the formal publication of the paper.**

***Index Terms*— Computer Vision, Deep Learning, Food Computing, Lightweight, Localization, Recognition**

I. Introduction

DRIVEN by socioeconomic advancement, global emphasis on dietary health is rapidly intensifying. This heightened awareness has brought critical aspects like nutritional analysis, ingredient traceability, and food source verification into sharp focus. Consequently, consumer demand is shifting towards health-conscious products such as organic and low-glycemic index foods, simultaneously stimulating innovation within the food technology sector to meet this demand. In this context, food image localization and recognition [1] has emerged as a pivotal technology within food computing. By enabling automated food identification through localization and classification, it provides a fundamental capability for diverse applications. These include nutritional assessment, clinical nutrition, food supply chain management [2]–[6], and personalized dietary recommendations [7]–[9]. Ultimately, these applications serve as integral components of preventive healthcare systems, playing a crucial role in dietary intervention and health management frameworks.

Food image localization and recognition pose a series of unique challenges that distinguish them from typical RGB image analysis, and detectors can be well adapted to this task. The inherent diversity and complexity of food images—spanning color, texture, shape, and ingredient composition—lead to a far richer, yet more challenging, feature landscape. First, the impact of cooking methods often results in high intra-class variability and low inter-class discriminability among food items [10]. For example, relying solely on local features can make it difficult to distinguish between highly similar food categories or to identify vastly different presentations within the same category. Consider cheesecake and ice cream, as shown in Fig. 2(a) and Fig. 2(b) respectively. Despite their similar colors and ingredients, they exhibit significant intra-class variation, with different cheesecakes and ice creams taking on diverse shapes. This contrasts sharply with images of animals like cats and birds (e.g., Fig. 1(a) and Fig. 1(b)), which generally display clear inter-class differences and higher intra-class consistency. Second, unlike animals such as birds or cats that possess independently discernible features (e.g., beaks, ears, tails) [11], food images typically feature complex and non-rigid structures. This inherent characteristic necessitates the extraction and utilization of multi-scale features for accurate fine-grained classification. Furthermore, accurately capturing long-range dependencies within food images is critical for robust identification [12]. This is particularly challenging given the random distribution of ingredients that often occurs during cooking processes like stir-frying or mixing.

Currently, widely used general models, such as two-stage object detectors like Faster R-CNN [13], Mask R-CNN [14], and Cascade R-CNN [15], achieve SOTA performance. However, due to their large number of parameters and computational demands, they are difficult to deploy effectively on resource-constrained edge devices. In contrast, one-stage localization and recognition models, such as the YOLO series [16] and SSD [17], have emerged in the field of object localization

(Corresponding author: Guorui Sheng.)

Xiangyi Zhu, Yancun Yang, Pindan Cao, and Guorui Sheng is with the School of Computer and Artificial Intelligence, Ludong University, Yantai, 264025, China (e-mail: zhuxiangyi@m.ldu.edu.cn; Harryyang@ldu.edu.cn; caopindan@m.ldu.edu.cn; shengguorui@ldu.edu.cn).

Bruce Denby was with Institut Langevin, ESPCI Paris, PSL University, CNRS, Sorbonne Université, Paris, 75005, France (e-mail: denby@ieee.org).

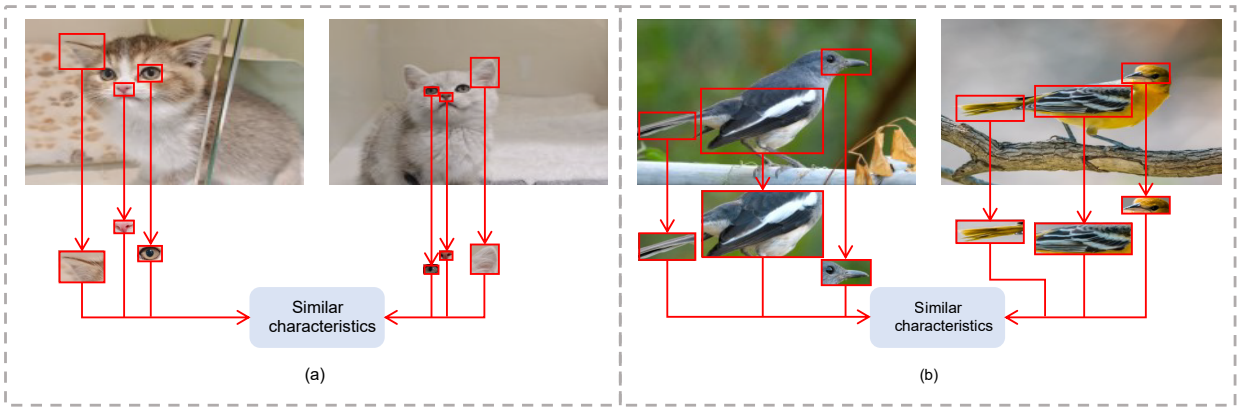

Fig. 1. Examples of general object images: (a) Cat; (b) Bird.

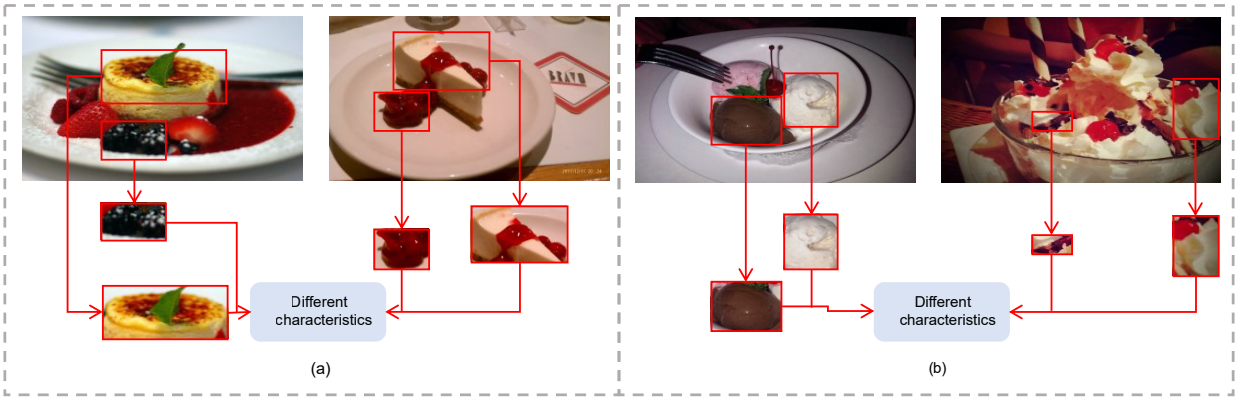

Fig. 2. Examples of food image: (a) Cheesecake; (b) Ice cream.

and recognition due to their unique localization and recognition mechanisms. These models enable an end-to-end object localization and recognition process. Thanks to this highly efficient localization and recognition paradigm, these models show clear advantages in speed while maintaining good localization and recognition accuracy, providing viable solutions for many practical application scenarios. For instance, to improve small object localization and recognition in UAV aerial images, the MFEL-YOLO [18] model implemented targeted feature enhancement optimizations across multiple stages, including feature extraction, feature fusion, and the localization and recognition head. Similarly, to boost localization and recognition accuracy for jujubes in complex orchard environments (especially differentiating cracked ones), the Jujube-YOLO [19] model significantly improved recognition precision by integrating new feature enhancement and attention modules into the network's backbone and neck. However, traditional YOLO localization and recognition models, limited by their initial design and architectural characteristics, often struggle to sufficiently and deeply extract the rich features contained within food images. This limitation prevents them from precisely and comprehensively grasping the essential characteristics of food images, ultimately affecting the accuracy of localization and recognition results.

We employ the GhostBottleneck (GB) from GhostNet [20] to capture fine-grained features in food images. To acquire multi-scale features from food images, we propose the Multi-Scale Feature Bottleneck (MSFB), which achieves progressive receptive field expansion through cascaded depthwise convolutions. This module is specifically designed to extract non-rigid multi-scale features inherent in food while ensuring lightweight computation. Additionally, we introduce the Bidirectional Vision Transformer (B-ViT), centered on our novel Bidirectional Self-Attention (BSA) mechanism. By independently computing attention scores along the image height and width dimensions, B-ViT efficiently captures global correlations. Unlike existing approaches, our B-ViT exhibits linear computational complexity and eliminates the need for positional encoding, thereby reducing both parameter count and computational overhead while effectively modeling global context. Finally, to achieve efficient fusion of the aforementioned heterogeneous features, we develop the Information Cross-Exchange (ICE) module. Through an innovative cross-recombination mechanism, ICE enables thorough integration of heterogeneous features. Experimental verification on the UEC Food100 [21], UEC Food256 [22], and ZSFooD [23] benchmark datasets proves the superiority of YOLO-MF2. Fig. 3, Fig. 4, and Fig. 5 demonstrate that YOLO-MF2 achieves comparable or even superior accuracy compared to other models in the YOLO series across three datasets, all while maintaining fewer parameters and lower computational complexity.

The principal contributions of this work are summarized as follows:

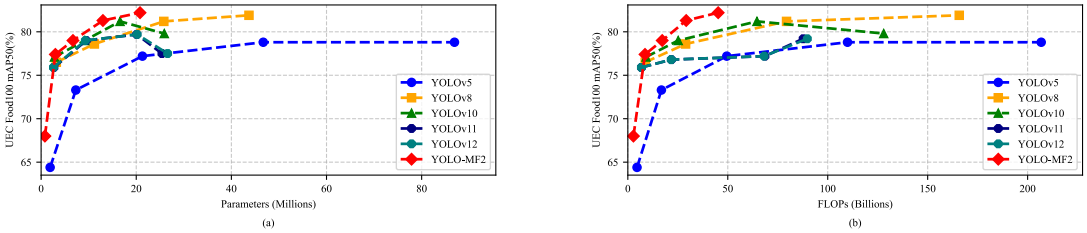

Fig. 3. Comparison of parameter counts and computational FLOPs between the proposed method and YOLO variants on the UEC Food100 dataset.

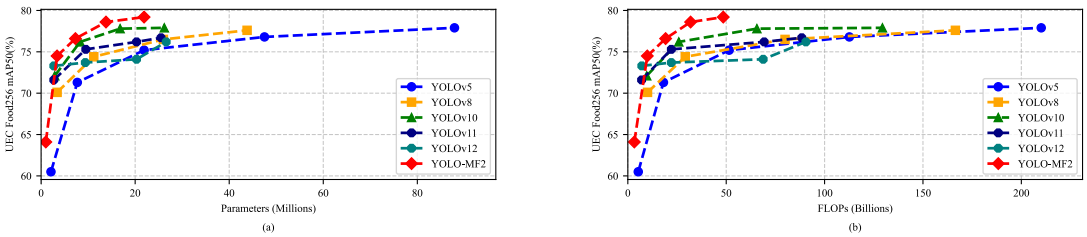

Fig. 4. Comparison of parameter counts and computational FLOPs between the proposed method and YOLO variants on the UEC Food256 dataset.

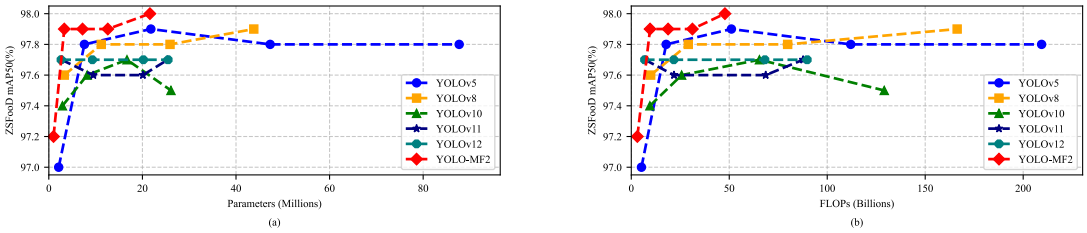

Fig. 5. Comparison of parameter counts and computational FLOPs between the proposed method and YOLO variants on the ZSFooD dataset.

- We propose a novel YOLO-Multi Feature Fusion (YOLO-MF2) that synergistically integrates local detail preservation, global context modeling and multi-scale feature handling. This design achieves enhanced performance in food localization and recognition while maintaining a lightweight architecture suitable for edge deployment.
- Multi-Scale Feature Extraction: We introduce the Multi-Scale Feature Bottleneck (MSFB) module, which employs depthwise convolutions and progressive receptive field expansion to extract discriminative multi-scale features from complex food images.
- Context Modeling: We propose a Bidirectional Vision Transformer (B-ViT) with Bidirectional Self-Attention (BSA), which computes correlations along both the width and height dimensions of the image to capture long-range dependencies while maintaining computational feasibility for edge devices.
- Feature Cross-Fusion: We developed an Information Cross-Exchange (ICE) module, which integrates heterogeneous features through cross-exchange reorganization, significantly improving localization and recognition accuracy.

II. Related work

A. General object detectors

Current general object detectors are broadly classified into two categories: two-stage and single-stage detectors. Two-stage detectors, such as Faster R-CNN [13], Mask R-CNN [14], and Cascade R-CNN [15], involve two phases: localizing candidate regions within an image and then classifying those regions. In contrast, single-stage detectors, like SSD [17] and the YOLO series [16], directly predict bounding boxes and classes, forming an end-to-end localization and recognition pipeline. Compared to two-stage detectors, single-stage detectors have greater potential for deployment on mobile devices while maintaining only a small loss in precision. For example, the YOLO series has progressively established its foundation in end-to-end object localization and recognition, from the groundbreaking reformulation of object localization and recognition as a single regression problem in YOLOv1 [24] to the introduction of Anchor Boxes and the expansion of localization and recognition categories to over nine thousand using the WordTree model in YOLOv2 [25]. YOLOv3 [26] significantly improved localization and recognition accuracy with the deeper Darknet53 network and multi-scale prediction techniques. YOLOv4 [27] optimized the DarkNet53 backbone, SPP and PAN structures for better performance. YOLOv5 [28] inherited the YOLOv4 framework, improving data augmentation strategies and expanding into a wider range of model

variants. YOLOX [29] is a high-performance, anchor-free object detector that introduces a Decoupled Head, advanced data augmentation, and the SimOTA dynamic label assignment strategy. YOLOv7 [30] introduces architectural innovations such as the Extended Efficient Layer Aggregation Network and a series of trainable optimization strategies. YOLOv8 [31] adopted a more optimized CSPNet backbone and a FPN-PAN neck structure, and shifted towards a more efficient Anchor-Free localization and recognition paradigm. However, these detectors often generate a large number of redundant bounding boxes that need to be filtered in the post-processing stage using Non-Maximum Suppression (NMS). YOLOv10 [32], developed by Tsinghua University in China, proposed a consistent dual assignment strategy for NMS-free training, which significantly improves localization and recognition efficiency. Moreover, YOLOv12 [33] pioneered an attention-centric YOLO framework by introducing an efficient Area Attention module, further enhancing model accuracy. While these methods perform well in general object localization and recognition, they are unable to fully exploit the complex features within food images.

B. Food object localization and recognition

Food localization and recognition [34] has garnered significant attention in the field of food computing [35] due to its potential applications in computer vision and multimedia [36]–[38]. However, this task faces significant challenges: on one hand, food itself exhibits complex characteristics in terms of ingredients, cuisine styles, and flavors; on the other hand, fine-grained food features suffer from significant intra-class variability and inter-class similarity [39], making it difficult to distinguish between different food objects. Currently, a common approach to food localization and recognition is to adapt general object localization and recognition frameworks. For example, [40] developed a mobile application system for food localization and recognition based on YOLOv2, while [41] proposed a weakly supervised method for generating food region proposals by training a fully convolutional network on web images. Although both of these works were designed to be lightweight for mobile or weakly supervised scenarios, their sole reliance on fully convolutional neural network architectures limits the models' ability to extract rich features such as texture, color, and shape from food images. The LOFI framework proposed by [11] addresses the challenge of distinguishing similar dishes through a multi-task fine-grained module that leverages lexical information. It also optimizes overlapping localization and recognition results with a graph confidence propagation module and employs strategies like Equilibrium Loss (EQLv2) to tackle the unique fine-grained and long-tail distribution challenges in food images. Despite achieving promising results in food image localization and recognition, the LOFI framework's large number of model parameters and computational demands prevent its effective deployment on edge devices.

C. Lightweight backbone for food localization and recognition

To achieve a lightweight food image localization and recognition model, a lightweight backbone network is essential. Recent advancements in lightweight architectures, such as MobileNetV4 [42] and RepViT [43], have significantly reduced model complexity while maintaining competitive performance. However, these general-purpose models often fail to adequately leverage food-specific features. GSNet [12] introduced Global Shuffle Convolution to efficiently capture long-range dependencies between dispersed ingredients in food images. AFNet [44] designed an Aggregation Block that efficiently captures global features of food images by averaging row and column information and combining it with convolutions. While both GSNet and AFNet effectively capture global information in food images and enhance model expressiveness while remaining lightweight, they do not account for the more complex multi-scale features present in food images.

III. Proposed method

This section details the YOLO-MF2 framework. We first present the overall architecture, then describe the baseline backbone components, and finally elaborate on three novel modules MSFB, B-ViT and ICE.

A. Overview of YOLO-MF2

The overall architecture of the model is illustrated in Fig. 6. The backbone network consists of five cascaded Multi-Feature Fusion (MF2) Blocks, which is named Multi-Feature Fusion Network (MF2Net). In the backbone network, there are three output branches from the last three layers. The Neck part of the model is identical to that of YOLOv5, but we've replaced the original Conv and C3 modules with the more lightweight GhostConv and C3Ghost. Additionally, we removed the SPPF module. These changes significantly reduce the model's parameters and computational load while maintaining its expressive power.

B. Multi-Feature Fusion Block

Food image localization and recognition is a fine-grained task characterized by its complex and rich features. To significantly boost a model's performance in this domain, it's crucial to comprehensively integrate the various features present in food images. This fundamental need led us to design the Multi-Feature Fusion (MF2) Block. The MF2 Block is composed of GB, MSFB, B-ViT, and ICE, as illustrated in the upper part of Fig. 6. Specifically, the GB module is employed to extract local features from food images, while the MSFB module is used to capture multi-scale features. The B-ViT module is responsible for acquiring global information within the food images. The ICE module is then utilized to enhance the information exchange among these local, multi-scale, and global features, subsequently fusing them. In a more detailed flow, the input first passes through parallel GB and MSFB modules to obtain local and multi-scale features, respectively. These features are then fused via an ICE module before

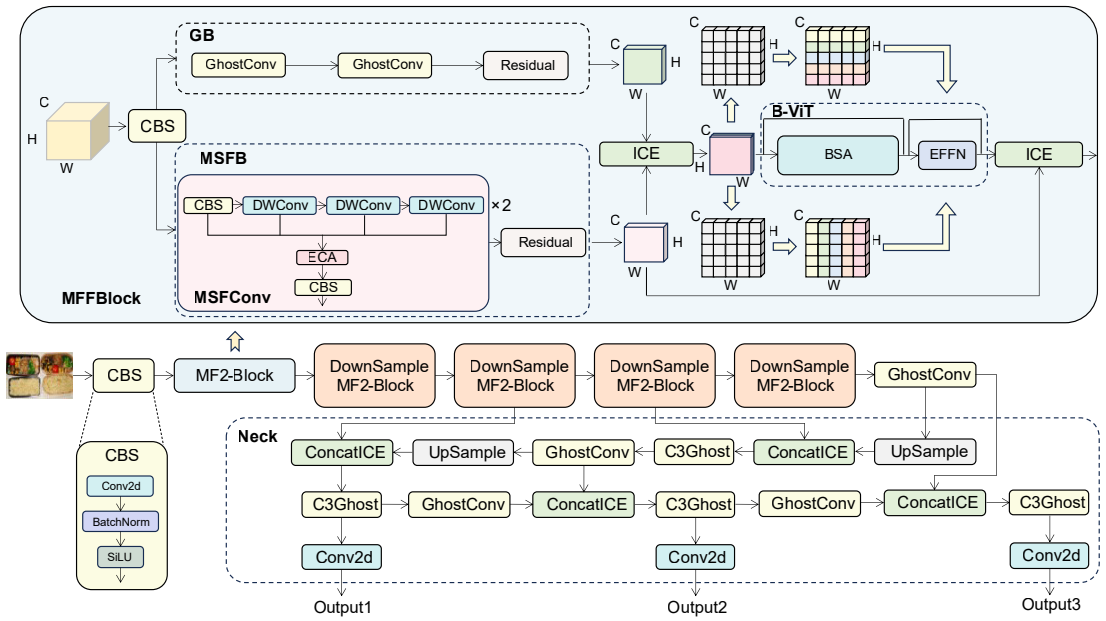

Fig. 6. Overall architecture of YOLO-MF2 depicting backbone and neck components.

proceeding to the B-ViT to acquire global information. Finally, a skip connection and another ICE module are used to fuse the local feature, multi-scale feature and global information, yielding the output.

C. Multi-Scale Feature Bottleneck

Food images often contain a variety of complex features, including color, texture, and shape. This complexity necessitates that a model be capable of extracting multi-scale features from food images to enhance localization and recognition accuracy. Therefore, this paper introduces the Multi-Scale Feature Bottleneck (MSFB). Inspired by InceptionNetv2 [45] and EfficientViT [46], we achieve the effect of large convolutional kernels by connecting (or stacking) convolutions with small kernels, and we use a cascaded structure to capture visual features at different scales. Specifically, the structure of the MSFB is illustrated in the middle-left portion of Fig. 6. It comprises two Multi-Scale Feature Convolution (MSFConv), with a final residual connection fusing the output with the input. An MSFConv consists of four convolutions connected in a cascaded manner. Initially, the input is processed by a CBS module. This is followed by three cascaded Depthwise Convolutions (DWConv), each with a kernel size of 3. This setup effectively captures feature maps at three different scales, equivalent to receptive fields of kernel sizes 3, 5 and 7. Subsequently, an ECA [47] module is used to compute the correlations between these different scale feature maps. Finally, another CBS module gives the ultimate output. Unlike the structure proposed in HRNet [48], MSFConv which is designed with Depthwise Separable Convolutions (DWConv) can extract multi-scale features while staying lightweight.

D. Bidirectional Vision Transformer

Food preparation often involves processes like stir-frying and mixing, which cause ingredients to be distributed dispersedly throughout an image. To enhance the accuracy of food image localization and recognition, models need the capability for global information modeling. This is why we designed the Bidirectional Vision Transformer (B-ViT). B-ViT is structured as shown in the upper right part of Fig. 6. It consists of a Bidirectional Spatial Attention (BSA) module and an Enhanced Feature Feedforward Network (EFFN). The BSA, designed based on Separable Self-Attention [49], calculates correlations along both the height and width dimensions of the image, thereby deriving the correlations among all pixels in the image. The EFFN enhances the original FFN by incorporating a Depthwise Convolution (DWConv) with a kernel size of 3, thereby strengthening the model's ability to acquire abstract features. Due to BSA's unique design, B-ViT can directly operate on images while preserving their original spatial positional information, obviating the need for operations such as patch embedding and positional embedding, which results in a smaller parameter count and computational cost for B-ViT. The workflow of B-ViT is illustrated in Fig. 7.

Specifically, we assume the input is a tensor $X \in \mathbb{R}^{c\times H\times W}$, where c is the channel count, H is the height, and W is the width. The query (q), key (k), and value (v) tensors are generated by applying a pointwise affine transformation to the feature vector at each spatial location of X. Each transformation uses a unique learnable weight matrix W and a bias vector b. For the BSA mechanism, the specific transformations are defined as:

$$\begin{aligned} q_h &= W_{q_h} X + b_{q_h} \\ q_w &= W_{q_w} X + b_{q_w} \\ k_h &= W_{k_h} X + b_{k_h} \\ k_w &= W_{k_w} X + b_{k_w} \\ v &= W_v X + b_v \end{aligned} \tag{1}$$

The dimensions of the weight matrices and bias vectors

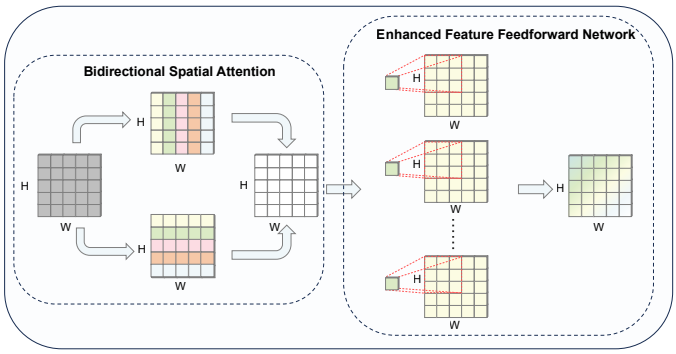

Fig. 7. Schematic diagram of the B-ViT module.

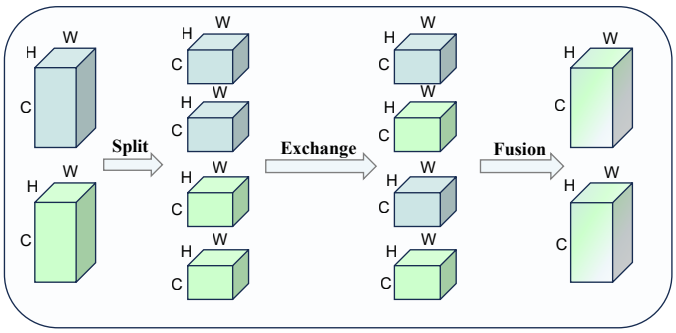

Fig. 8. Schematic diagram of the ICE module.

determine the output tensor shapes. For the transformations above:

- To generate the queries q_h and q_w, the weights and biases are $W_{q_h}, W_{q_w} \in \mathbb{R}^{1\times c}$ and $b_{q_h}, b_{q_w} \in \mathbb{R}^1$. This compresses the channel dimension to 1, yielding output tensors in $\mathbb{R}^{1\times H\times W}$.
- To generate the keys k_h, k_w and the value v, the weights and biases are $W_{k_h}, W_{k_w}, W_v \in \mathbb{R}^{c\times c}$ and $b_{k_h}, b_{k_w}, b_v \in \mathbb{R}^c$. This preserves the channel dimension, yielding output tensors in $\mathbb{R}^{c\times H\times W}$.

Then, the query for the H-direction and W-direction are applied with $Softmax$ along their respective directions to obtain the weights for both directions. Afterward, the resulting scores of the queries are multiplied by the corresponding keys, and the results are summed for both directions to obtain the context relevance vectors $c_h^{c\times H\times 1}$ and $c_w^{c\times 1\times W}$, as shown in the following formulas:

$$\begin{aligned} c_h &= \sum_i^W (Softmax(q_h) * k_h) \\ c_w &= \sum_i^H (Softmax(q_w) * k_w) \end{aligned} \tag{2}$$

Finally, the context relevance vectors for both directions are multiplied by the value to obtain the attention feature maps for the H-direction and W-direction, denoted as $M_h^{c\times H\times W}$ and $M_w^{c\times H\times W}$, respectively. Then, the attention feature maps from both directions are summed and fused to obtain the final attention feature map for the entire image, $Output^{c\times H\times W}$. The process can be expressed by the following formula:

$$\begin{aligned} M_h &= ReLU(v) * c_h \\ M_w &= ReLU(v) * c_w \\ Output &= M_h + M_w \end{aligned} \tag{3}$$

Since this process directly operates on the image, it retains the original spatial information of the image, thus eliminating the need for the positional encoding and Patch Embedding typically required in traditional ViT models. This makes the B-ViT module more efficient.

E. Information Cross-Exchange

Since YOLO-MF2 extracts three distinct types of features independently, a simple fusion of these features could lead to a loss of valuable information. Therefore, this paper introduces the Information Cross-Exchange (ICE) module to enhance the information flow and fusion among these different feature types, thereby improving the model's accuracy. The workflow of the Information Cross-Exchange (ICE) module is depicted in Fig. 8. It first averages and groups the incoming heterogeneous features, then performs a cross-exchange operation, and finally fuses them.

Although the ICE module shares superficial similarities with existing channel fusion techniques, such as the Channel Shuffle operation in ShuffleNet [50] and the Cross-Stitch networks in multi-task learning [51], it is fundamentally different in purpose and mechanism. ICE is specifically designed for exchanging and integrating heterogeneous features from multiple streams, rather than merely shuffling or linearly combining channels, thereby enabling deeper semantic interaction across feature types.

Specifically, we assume the input is a tensor $f_i \in \mathbb{R}^{c_i\times H\times W}$, where c_i is the channel count, H is the height, W is the width, $i \in \{0, ..., n-1\}$ and n is the number of features. The specific computational process for the first step of the ICE module is as follows:

$$f_{i_j} = f_i[s_i * j : s_i * (j+1), :, :] \tag{4}$$

Here, f_{i_j} represents the j-th segment when the i-th type of feature is equally divided into n parts, s_i equals c_i/n and $j \in \{0, ..., n-1\}$.

Next, the calculation process for the cross-exchange operation is as follows:

$$f_{c_i} = Concat(f_{0_i}, f_{1_i}, ..., f_{n-1_i}) \tag{5}$$

Here, f_{c_i} represents the features after cross-exchange operation.

Finally, these features are fused using a concatenation operation, which is the approach adopted in this paper.

$$f_o = Fuse(f_{c_0}, f_{c_1}, ..., f_{c_n-1}) \tag{6}$$

Here f_o denotes the final output feature.

IV. Experiments

This section details our experimental setup and evaluation results. We compared the localization and recognition performance of YOLO-MF2 against representative YOLO variants

across three benchmark datasets: UEC Food100 [21], UEC Food256 [22], and ZSFooD [23]. Comprehensive ablation studies were conducted to validate the effectiveness of our proposed modules. All experiments were implemented using Python 3.9.19 and PyTorch 1.12.1 on an NVIDIA A800 GPU (80GB).

A. Datasets

UEC Food100 [21] is a dataset comprising 14,361 images across 100 food categories, primarily featuring common Japanese dishes. A significant characteristic of this dataset is its inclusion of "multi-food images," where two or more food items appear in a single image, alongside images containing only a single food item. To support object localization and recognition tasks, the dataset provides bounding box annotations for each food item.

UEC Food256 [22] contains 256 food categories, comprising a total of 26,184 images with approximately 28,429 annotated bounding boxes. Its distinctive feature is its extensive cultural diversity, covering cuisines from multiple countries including France, Italy, the United States, China, Thailand, Vietnam, Japan, and Indonesia.

ZSFooD [23] comprises 20,603 food images collected from 10 real restaurant scenarios. This means its images better reflect the lighting, backgrounds, and food presentation found in actual dining environments. Each food object in these scenes is annotated with bounding boxes, totaling 95,322 annotated bounding boxes across 228 categories.

In this study, we used 80% of each of the three datasets for training and the remaining 20% for validation. This approach allowed us to evaluate our model's effectiveness in food image localization and recognition tasks.

B. Implementation Details

In our experiments, all models were trained using the Stochastic Gradient Descent (SGD) optimizer. The initial learning rate was set to 0.01, and the momentum factor was 0.937. Training was conducted for a total of 300 epochs with a batch size of 16. During both training and testing, the input image size was consistently 640×640 pixels.

C. Comparison with Other Methods

To evaluate the effectiveness of YOLO-MF2, we conducted experiments on the UEC Food100, UEC Food256, and ZSFooD datasets. We then compared our model's results against other lightweight localization and recognition models, including YOLOv5 [28], YOLOv5-Ghost, YOLOv8 [31], YOLOv10 [32], YOLOv11 [52], and YOLOv12 [33]. Among them, YOLOv5-Ghost is derived by replacing all standard convolutions in the YOLOv5 architecture with GhostConv from the lightweight GhostNet [20]. Additionally, we performed experiments using various lightweight backbone networks, specifically the general lightweight backbones MobileNetV4 [42] MobileViTv2 [49], and RepViT [43], and the food image-specific lightweight backbone GSNet [12]. Finally, we compared our model with the smallest-scale models under other frameworks, including YOLOv7-tiny [29], EfficientDet [53], NanoDet [54], and PP-YOLOE [55].

For evaluation across all datasets, we used mean average precision (mAP) as the primary metric, consistent with the COCO evaluation, specifically reporting mAP_{50} and mAP_{50-95}. Specifically, mAP_{50} represent the average precision calculated using IoU thresholds of 0.5, while mAP_{50-95} is computed by averaging the precision at ten IoU thresholds ranging from 0.5 to 0.95, incremented by 0.05.

1) Results on UEC Food100 dataset: The results of Table I clearly show the strong performance of YOLO-MF2 across various parameter groups. In the first group of models, YOLO-MF2 demonstrates a remarkable improvement over YOLOv5-Ghost and YOLOv5. It achieves 0.7% and 3.6% higher mAP_{50}, and 1.5% and 4.7% higher mAP_{50-95}, respectively, despite having 0.3M and 1.1M fewer parameters. Furthermore, its FLOPs are equivalent to YOLOv5-Ghost and 1.8G less than YOLOv5. For the second group, while YOLO-MF2 has slightly more parameters (0.2M, 0.3M and 0.4M) than YOLOv10, YOLOv11, and YOLOv12, it delivers higher mAP_{50} by 0.3%, 1.6% and 1.5%, respectively, with mAP_{50-95} remaining largely comparable. In the third group, YOLO-MF2 stands out with the smallest parameter count and FLOPs. Compared to YOLOv12, YOLO-MF2 reduces parameters by 2.6M and FLOPs by 4.5G, yet still boasts 2.2% and 0.7% higher mAP at both thresholds. For the fourth and fifth model groups, YOLO-MF2 consistently achieves the lowest parameter counts and FLOPs (13.0M and 29.2G for the fourth group; 20.8M and 45.2G for the fifth group). Simultaneously, it maintains the highest mAP_{50} (81.3% and 82.2% respectively) and comparable mAP_{50-95} (60.6% and 62.0%) to other models. Specifically, in the fourth group, YOLO-MF2's mAP_{50} is 2.5%, 0.1%, 0.1%, 1.6% and 4.1% higher than YOLOv5, YOLOv8, YOLOv10, YOLOv11, and YOLOv12, respectively. In the fifth group, its mAP_{50} is 3.4%, 0.3%, 2.4%, 4.7% and 3% higher than YOLOv5, YOLOv8, YOLOv10, YOLOv11 and YOLOv12, respectively.

YOLO-MF2 also exhibits robust performance when compared to the largest parameter models within each detector family, as detailed in Table II. Specifically, YOLO-MF2 achieves mAP_{50} and mAP_{50-95} values that are both 1.8 percentage points superior to YOLOv12, despite its parameter count being merely 34.8% of YOLOv12's, and its FLOPs only 22.5% of YOLOv12's. When contrasted with YOLOv5, YOLOv5-Ghost, YOLOv10, and YOLOv11, YOLO-MF2 consistently demonstrates the lowest parameter count and FLOPs while simultaneously achieving the highest mAP_{50} and mAP_{50-95}. Relative to YOLOv8, although YOLO-MF2's mAP_{50} is only marginally higher by 0.1 percentage points and its mAP_{50-95} is 1.4 percentage points lower, it maintains a substantially reduced model complexity, with its parameter count being just 30.5% that of YOLOv8 and its FLOPs only 17.5% that of YOLOv8.

Table III presents the experimental results when different backbone networks are employed within the YOLOv5 framework. In these experiments, we replaced all standard convolutions in YOLOv5's neck with GhostConv and C3Ghost before swapping out the backbone networks. We compared two

TABLE I
COMPARATIVE PERFORMANCE OF YOLO-MF2 VERSUS YOLO VARIANTS ON THE UEC FOOD100 DATASET.

Method	Params (M)	FLOPs (G)	mAP_{50} (%)	mAP_{50-95} (%)
YOLOv5n	1.9	4.6	64.4	44.1
YOLOv5n-Ghost	1.1	2.8	67.3	47.2
YOLO-MF2(Ours)	**0.8**	**2.8**	**68.0**	**48.8**
YOLOv5s	7.3	16.8	73.3	51.4
YOLOv5s-Ghost	3.9	9.1	76.6	54.8
YOLOv8n	3.2	8.7	76.5	57.4
YOLO-MF2(Ours)	3.0	8.6	**77.4**	57.5
YOLOv10n	2.8	9.1	77.1	**57.9**
YOLOv11n	2.7	6.8	75.8	57.1
YOLOv12n	**2.6**	**6.8**	75.9	57.1
YOLOv5m	21.3	49.5	77.2	56.2
YOLOv8s	11.2	28.9	78.6	59.2
YOLOv11s	9.5	21.8	78.9	**60.2**
YOLOv12s	9.3	21.7	76.8	58.5
YOLOv10s	9.2	25.2	79.0	59.7
YOLOv5m-Ghost	8.9	19.8	77.4	56.7
YOLO-MF2(Ours)	**6.7**	**17.2**	**79.0**	59.2
YOLOv5l	46.7	109.9	78.8	57.5
YOLOv8m	25.8	79.4	81.2	**61.8**
YOLOv12m	20.2	68.2	77.2	58.7
YOLOv11m	20.1	68.6	79.7	60.8
YOLOv10m	16.6	64.6	81.2	61.9
YOLOv5l-Ghost	16.1	35.2	79.3	58.4
YOLO-MF2(Ours)	**13.0**	**29.2**	**81.3**	60.6
YOLOv5x	86.9	206.8	78.8	58.3
YOLOv8l	43.7	165.8	81.9	**62.7**
YOLOv12l	26.5	89.8	79.2	60.1
YOLOv10l	25.9	128.0	79.8	60.9
YOLOv5x-Ghost	25.7	55.7	79.0	58.1
YOLOv11l	25.4	87.7	77.5	59.2
YOLO-MF2(Ours)	**20.8**	**45.2**	**82.2**	62.0

TABLE II
COMPARATIVE PERFORMANCE OF YOLO-MF2 VERSUS MAXIMUM-SCALE YOLO VARIANTS ON THE UEC FOOD100 DATASET.

Method	Params (M)	FLOPs (G)	mAP_{50} (%)	mAP_{50-95} (%)
YOLOv5x	86.9	206.8	78.8	58.3
YOLOv8x	68.2	258.7	82.1	**63.4**
YOLOv12x	59.2	200.5	80.4	60.2
YOLOv11x	57.0	196.1	79.3	60.9
YOLOv10x	38.8	172.1	77.2	57.7
YOLOv5x-Ghost	25.7	55.7	79.0	58.1
YOLO-MF2(Ours)	**20.8**	**45.2**	**82.2**	62.0

scales of MF2Net. For the small variant of MF2Net, the parameter count is only 3M, making it the most compact model among all compared baselines. Specifically, it uses 4.9M fewer parameters than RepViT, 4.3M fewer than GSNet, 3.9M fewer than MobileNetV4, and 2.2M fewer than MobileViTv2. Despite its compact size, MF2Net-small achieves a mAP_{50} that is 11.1 points higher than MobileNetV4, 10.1 points higher than MobileViTv2, and only 1.2 points below GSNet. More importantly, its mAP_{50-95} surpasses MobileNetV4 by 9.5 points, MobileViTv2 by 19.3 points, and even GSNet by 1.3 points, demonstrating superior localization accuracy under stricter IoU thresholds. For the large version of MF2Net, it also maintains the smallest parameter count at just 6.7M, which

TABLE III
BACKBONE COMPARISON ON UEC FOOD100 DATASET: MF2NET VERSUS LIGHTWEIGHT MODELS

Backbone	Params (M)	FLOPs (G)	mAP_{50} (%)	mAP_{50-95} (%)
RepViT	7.9	20.6	78.6	58.2
GSNet	7.3	**6.1**	78.6	56.2
MobileNetV4	6.9	7.2	66.3	48.0
MF2Net-l	6.7	17.2	**79.0**	**59.2**
MobileViTv2	5.2	9.4	67.3	38.2
MF2Net-s	**3.0**	8.6	77.4	57.5

TABLE IV
COMPARATIVE PERFORMANCE OF YOLO-MF2 VERSUS OTHER LIGHTWEIGHT MODEL ON THE UEC FOOD100 DATASET.

Method	Params (M)	FLOPs (G)	mAP_{50} (%)	mAP_{50-95} (%)
NanoDet-m	0.9	**0.7**	65.6	46.4
YOLO-MF2(Ours)	**0.8**	2.8	**68.0**	**48.8**
PP-YOLOE-Lite	7.9	17.4	74.3	55.6
YOLOv7-tiny	6.3	14.0	76.1	53.7
EfficientDet-Lite	3.9	**2.5**	36.1	22.9
YOLO-MF2(Ours)	**3.0**	8.6	**77.4**	**57.5**

is 1.2M less than RepViT. Yet, it achieves mAP increases of 0.4 and 1 percentage points compared to RepViT. When compared to GSNet, it has 0.6M fewer parameters while showing mAP increases of 0.4 and 3 percentage points. Furthermore, against MobileNetV4, it reduces parameters by 0.2M and delivers significant mAP increases of 12.7 and 11.2 percentage points. Compared with MobileViTv2, MF2Net introduces an additional 1.5M parameters and 7.8G FLOPs; however, these increases yield substantial performance gains, improving mAP by 11.7 percentage points and 21.0 percentage points across the respective metrics. Overall, MF2Net strikes an effective balance between achieving high detection accuracy and maintaining a lightweight architecture on the UEC-Food100 dataset.

On the UEC Food100 dataset, the results in Table IV show that under a small-scale setting, YOLO-MF2 with 0.8M parameters and 2.8G FLOPs attains mAP_{50} of 68.0% and mAP_{50-95} of 48.8%, yielding consistent gains over NanoDet-m at 65.6% and 46.4% with only modest additional computation. Under a higher-capacity setting, YOLO-MF2 with 3.0M parameters and 8.6G FLOPs reaches mAP_{50} of 77.4% and mAP_{50-95} of 57.5%, surpassing PP-YOLOE-Lite at 74.3% and 55.6% and YOLOv7-tiny at 76.1% and 53.7% while using fewer parameters. By contrast, EfficientDet-Lite, despite a similar parameter budget and lower FLOPs, yields mAP_{50} of 36.1% and mAP_{50-95} of 22.9%, indicating limited effectiveness on this benchmark.

2) Results on UEC Food256 dataset: Table V presents YOLO-MF2's performance against other detectors on the UEC Food256 dataset across various parameter groupings. In the first model group, YOLO-MF2 demonstrates notable efficiency and accuracy gains. It uses 0.3M and 1.1M fewer parameters than YOLOv5-Ghost and YOLOv5, respectively, and reduces FLOPs by 0.1G and 2G. Concurrently, it achieves a 0.8% and

TABLE V
COMPARATIVE PERFORMANCE OF YOLO-MF2 VERSUS YOLO VARIANTS ON THE UEC FOOD256 DATASET.

Method	Params (M)	FLOPs (G)	mAP_{50} (%)	mAP_{50-95} (%)
YOLOv5n	2.1	5.3	60.5	44.4
YOLOv5n-Ghost	1.3	3.4	63.3	47.0
YOLO-MF2(Ours)	**1.0**	**3.3**	**64.1**	**48.6**
YOLOv5s	7.7	18.1	71.3	53.3
YOLOv5s-Ghost	4.4	10.4	74.0	56.0
YOLOv8n	3.4	10.0	70.1	54.1
YOLO-MF2(Ours)	3.4	9.9	**74.5**	**56.8**
YOLOv10n	2.9	9.6	72.1	55.3
YOLOv11n	2.7	**7.0**	71.6	55.1
YOLOv12n	**2.7**	**7.1**	73.3	56.7
YOLOv5m	21.9	51.5	75.2	57.2
YOLOv8s	11.2	29.2	74.4	57.5
YOLOv5m-Ghost	9.6	21.8	75.8	57.9
YOLOv11s	9.5	22.1	75.3	58.2
YOLOv12s	9.4	22.1	73.7	57.5
YOLOv10s	8.3	25.9	76.2	58.7
YOLO-MF2(Ours)	**7.3**	**19.2**	**76.6**	**58.7**
YOLOv5l	47.5	112.6	76.8	58.9
YOLOv8m	26.0	79.9	76.5	59.3
YOLOv11m	20.3	69.3	76.2	59.1
YOLOv12m	20.3	68.8	74.1	58.0
YOLOv5l-Ghost	16.9	37.9	77.0	59.2
YOLOv10m	16.8	65.6	77.8	60.4
YOLO-MF2(Ours)	**13.8**	**31.9**	**78.6**	**60.6**
YOLOv5x	87.9	210.1	77.9	59.7
YOLOv8l	43.8	166.5	77.6	60.5
YOLOv5x-Ghost	26.8	59.1	77.4	59.4
YOLOv12l	26.6	90.5	76.2	59.4
YOLOv10l	26.2	129.4	77.9	60.6
YOLOv11l	25.5	88.4	76.7	59.8
YOLO-MF2(Ours)	**21.9**	**48.5**	**79.2**	**61.1**

3.6% higher mAP_{50}, and a 1.6% and 4.2% higher mAP_{50-95}.

Table VI presents the results of YOLO-MF2 against the largest parameter models of various detectors on the UEC Food256 dataset. YOLO-MF2 achieves an mAP_50 of 79.2% and an mAP_{50-95} of 61.1% with a parameter count of just 21.9M and FLOPs of 48.5G. Compared to YOLOv12, YOLO-MF2's parameter count is only 36.8% and its FLOPs are only 24.1% of YOLOv12's. Despite this, its mAP_50 is 1.4% higher and its mAP_{50-95} is 0.2% higher. Relative to YOLOv11, YOLO-MF2's parameter count is merely 38.3% and its FLOPs are just 24.6% of YOLOv11's. Yet, it achieves mAP_50 and mAP_{50-95} scores that are 5.6% and 3.3% higher, respectively. From the results presented in Table V, it is evident that YOLO-MF2 consistently achieves the lowest parameter count and FLOPs, while simultaneously demonstrating the highest mAP scores.

Table VII presents the experimental results on the UEC Food256 dataset using various lightweight backbone networks. The model utilizing the small version of MF2Net has the lowest parameter count and moderate FLOPs, though with slightly lower mAP scores. However, compared with MobileViTv2, the small variant of MF2Net maintains the same FLOPs while reducing the parameter count by 2.2M, and achieves notable performance gains—improving mAP by 11.4 percentage points and 17.4 percentage points under the two

TABLE VI
COMPARATIVE PERFORMANCE OF YOLO-MF2 VERSUS MAXIMUM-SCALE YOLO VARIANTS ON THE UEC FOOD256 DATASET.

Method	Params (M)	FLOPs (G)	mAP_{50} (%)	mAP_{50-95} (%)
YOLOv5x	87.9	210.1	77.9	59.7
YOLOv8x	68.4	259.5	78.1	60.9
YOLOv12x	59.4	201.5	77.8	60.9
YOLOv11l	57.2	197.1	73.6	57.8
YOLOv10x	32.2	173.7	77.7	60.3
YOLOv5x-Ghost	26.8	59.1	77.4	59.4
YOLO-MF2(Ours)	**21.9**	**48.5**	**79.2**	**61.1**

TABLE VII
BACKBONE COMPARISON ON UEC FOOD256 DATASET: MF2NET VERSUS LIGHTWEIGHT MODELS.

Backbone	Params (M)	FLOPs (G)	mAP_{50} (%)	mAP_{50-95} (%)
RepViT	8.3	22.0	76.4	58.7
GSNet	7.7	**7.5**	76.4	57.4
MobileNetV4	7.3	7.6	66.4	50.4
MF2Net-l	7.3	19.2	**76.6**	**58.7**
MobileViTv2	5.6	9.9	63.1	39.4
MF2Net-s	**3.4**	9.9	74.5	56.8

IoU thresholds, respectively. In contrast, the model incorporating the large version of MF2Net shows comparable parameter counts to MobileNetV4. However, due to the inclusion of Vision Transformer (ViT), its FLOPs are 11.6G higher than MobileNetV4, yet it achieves significantly higher mAP across both thresholds (10.2% higher mAP_{50} and 8.3% higher mAP_{50-95}). Compared to GSNet, the large MF2Net model uses 0.4M fewer parameters, albeit with 11.7G higher FLOPs. Despite this, it yields 0.2% higher mAP_{50} and 1.3% higher mAP_{50-95}. Finally, when compared to RepViT, the large MF2Net model reduces parameters by 1M and FLOPs by 2.8G, while achieving a 0.2% higher mAP_{50} and comparable mAP_{50-95}.

On the UEC Food256 dataset, as summarized in Table VIII, the compact YOLO-MF2 with 1.0M parameters and 3.3G FLOPs achieves 64.1% mAP_{50} and 48.6% mAP_{50-95}, exceeding NanoDet-m at 53.6% and 37.9% with only modest computation. With a larger configuration of 3.4M parameters and 9.9G FLOPs, YOLO-MF2 reaches 74.5% mAP_{50} and 56.8% mAP_{50-95}, outperforming PP-YOLOE-Lite at 65.2% and 49.1% and YOLOv7-tiny at 72.2% and 54.6% while using fewer parameters than the latter. EfficientDet-Lite records 38.2% and 28.4% despite a similar parameter budget and lower FLOPs, underscoring limited effectiveness on this benchmark.

3) Results on ZSFooD dataset: We also compared YOLO-MF2 with different detectors on ZSFooD, and the experimental results for different parameter count groups are presented in Table IX. In the first group, YOLO-MF2 exhibited 0.2M fewer parameters and 0.2G fewer FLOPs than YOLOv5-Ghost, while achieving a 0.1 percentage point higher mAP in both metrics. Compared to YOLOv5, it demonstrated a reduction of 0.1M in parameters and 2.1G in FLOPs, with mAP improvements of 0.2 and 0.3 percentage points, respectively. In the second group, although YOLO-MF2 showed an increase

TABLE VIII
COMPARATIVE PERFORMANCE OF YOLO-MF2 VERSUS OTHER LIGHTWEIGHT MODEL ON THE UEC FOOD256 DATASET.

Method	Params (M)	FLOPs (G)	mAP_{50} (%)	mAP_{50-95} (%)
NanoDet-m	1.1	**1.2**	53.6	37.9
YOLO-MF2(Ours)	**1.0**	3.3	**64.1**	**48.6**
PP-YOLOE-Lite	8.3	19.4	65.2	49.1
YOLOv7-tiny	6.6	16.1	72.2	54.6
EfficientDet-Lite	4.3	**4.5**	38.2	28.4
YOLO-MF2(Ours)	**3.4**	9.9	**74.5**	**56.8**

TABLE IX
COMPARATIVE PERFORMANCE OF YOLO-MF2 VERSUS YOLO VARIANTS ON THE ZSFOOD DATASET.

Method	Params (M)	FLOPs (G)	mAP_{50} (%)	mAP_{50-95} (%)
YOLOv5n	2.1	5.2	97.0	93.8
YOLOv5n-Ghost	1.2	3.3	97.1	94.0
YOLO-MF2(Ours)	**1.0**	**3.1**	**97.2**	**94.1**
YOLOv5s	7.6	17.8	97.8	95.4
YOLOv5s-Ghost	4.3	10.1	97.9	95.1
YOLO-MF2(Ours)	3.3	9.6	**97.9**	**95.5**
YOLOv8n	3.3	9.9	97.6	94.4
YOLOv10n	2.9	9.5	97.4	95.2
YOLOv11n	2.7	7.0	97.7	95.5
YOLOv12n	**2.6**	**6.8**	97.7	95.5
YOLOv5m	21.8	51.1	97.9	95.7
YOLOv8s	11.2	29.1	97.8	95.5
YOLOv11s	9.5	21.8	97.6	95.8
YOLOv5m-Ghost	9.4	21.3	97.9	95.4
YOLOv12s	9.3	21.7	97.7	**96.4**
YOLOv10s	8.2	25.7	97.6	94.8
YOLO-MF2(Ours)	**7.2**	**18.8**	**97.9**	95.9
YOLOv5l	47.3	112.0	97.8	95.8
YOLOv8m	25.9	79.8	97.8	96.3
YOLOv12m	20.2	68.2	97.7	96.3
YOLOv11m	20.1	68.6	97.6	96.2
YOLOv5l-Ghost	16.8	37.3	97.9	95.5
YOLOv10m	16.7	65.4	97.7	94.5
YOLO-MF2(Ours)	**12.6**	**31.2**	**97.9**	**96.5**
YOLOv5x	87.7	209.4	97.8	95.8
YOLOv8l	43.8	166.3	97.9	96.4
YOLOv5x-Ghost	26.6	58.3	97.9	95.6
YOLOv12l	26.5	89.8	97.7	96.4
YOLOv10l	26.1	129.1	97.5	94.8
YOLOv11l	25.4	87.7	97.7	96.2
YOLO-MF2(Ours)	**21.6**	**47.8**	**98.0**	**96.5**

of 0.7M in parameters and 2.8G in FLOPs compared to YOLOv12, it achieved a 0.2 percentage point higher mAP_{50}, with mAP_{50-95} remaining consistent. While YOLOv11 and YOLOv12 exhibited the same mAP, the disparities in parameter count and FLOPs between YOLO-MF2 and YOLOv11 were reduced. Compared to YOLOv8, YOLO-MF2 maintained an identical parameter count but achieved 0.3G lower FLOPs, with mAP improvements of 0.3% and 1.1%, respectively. In the third, fourth, and fifth groups, YOLO-MF2 consistently exhibited the minimum parameter count and FLOPs, while its mAP was comparable to or even surpassed that of other detectors.

Table X presents the results of YOLO-MF2 compared to the largest models of various detectors on ZSFooD. From

TABLE X
COMPARATIVE PERFORMANCE OF YOLO-MF2 VERSUS MAXIMUM-SCALE YOLO VARIANTS ON THE ZSFOOD DATASET.

Method	Params (M)	FLOPs (G)	mAP_{50} (%)	mAP_{50-95} (%)
YOLOv5x	87.7	209.4	97.8	95.8
YOLOv8x	68.4	256.3	98.0	96.1
YOLOv5x-Ghost	26.6	58.3	97.9	95.6
YOLOv12x	59.2	200.5	97.7	96.4
YOLOv10x	32.1	173.4	97.5	94.9
YOLOv11x	57.0	196.9	97.8	96.3
YOLO-MF2(Ours)	**21.6**	**47.8**	**98.0**	**96.5**

the results shown in the table, we can observe that YOLO-MF2 possesses the smallest parameter count and FLOPs. Its parameter count is merely 21.6M, which is only 45.2% of YOLOv12, and its FLOPs are only 47.8G, constituting 23.8% of YOLOv12. Despite these significantly reduced computational resources, its mAP_{50} is 0.3% higher than YOLOv12, and its mAP_{50-95} is 0.1% higher. Compared to other detectors, YOLO-MF2 achieves the lowest parameter count and FLOPs while maintaining comparable or even superior mAP performance.

We also experimented with various lightweight backbone networks on the ZSFooD dataset, with results presented in Table XI. When employing the small version of MF2Net as the backbone, it achieved the highest mAP compared to other lightweight backbone networks. It also boasted the smallest parameter count, at just 3.3M. Its FLOPs were only marginally higher than those of the purely convolutional backbones MobileNetV4 and GSNet, and remained comparable to MobileViTv2. The large version of MF2Net delivered an even higher mAP_{50-95}, reaching 95.9%.

On the ZSFood dataset, as summarized in Table XII, the compact YOLO-MF2 with 1.0M parameters and 3.1G FLOPs attains 97.2% mAP_{50} and 94.1% mAP_{50-95}, yielding gains over NanoDet-m at the cost of moderately higher computation. With a larger configuration of 3.3M parameters and 9.6G FLOPs, YOLO-MF2 reaches 97.9% mAP_{50} and 95.5% mAP_{50-95}, while using substantially fewer parameters and FLOPs than PP-YOLOE-Lite and YOLOv7-tiny. EfficientDet-Lite records 72.9% and 63.7%, indicating limited effectiveness on this benchmark.

4) Statistical Significance and Practical Relevance: Table XIII reports repeated experiments on UEC Food100, UEC Food256, and ZSFooD using five different random seeds ($n = 5$), presented as mean±standard deviation. Compared with YOLOv5n, YOLO-MF2 improves mAP_{50} on UEC Food100 from 64.0%±0.3% to 67.6±0.4 (absolute +3.6%; +5.6% relative), and mAP_{50-95} from 43.9%±0.2% to 48.6%±0.4% (absolute +4.7%; +10.7%). On UEC Food256, mAP_{50} increases from 60.2%±0.2% to 63.6%±0.5% (+3.4%; +5.7%), and mAP_{50-95} from 44.1%±0.2% to 48.1%±0.3% (+4.0%; +9.1%). On ZSFooD, mAP_{50} and mAP_{50-95} rise from 96.6%±0.4% and 93.4%±0.3% to 96.8%±0.4% (+0.2%; +0.21%) and 93.8%±0.3% (+0.4%; +0.43%), respectively. All standard deviations are ≤ 0.5, indicating low across-seed variance and stable outcomes; moreover, the gains are more

TABLE XI
BACKBONE COMPARISON ON ZSFOOD DATASET: MF2NET VERSUS LIGHTWEIGHT MODELS.

Backbone	Params (M)	FLOPs (G)	mAP_{50} (%)	mAP_{50-95} (%)
RepViT	8.2	21.7	97.7	95.4
GSNet	7.6	**7.2**	97.7	95.3
MobileNetV4	7.2	7.5	96.1	90.9
MF2Net-l	7.2	18.8	**97.9**	**95.9**
MobileViTv2	5.5	9.6	95.1	89.2
MF2Net-s	**3.3**	9.6	97.9	95.5

TABLE XII
COMPARATIVE PERFORMANCE OF YOLO-MF2 VERSUS OTHER LIGHTWEIGHT MODEL ON THE ZSFOOD DATASET.

Method	Params (M)	FLOPs (G)	mAP_{50} (%)	mAP_{50-95} (%)
NanoDet-m	1.0	**1.1**	96.7	93.1
YOLO-MF2(Ours)	**1.0**	3.1	**97.2**	**94.1**
PP-YOLOE-Lite	8.2	18.9	97.8	95.4
YOLOv7-tiny	6.5	14.9	97.8	95.1
EfficientDet-Lite	4.2	**4.1**	72.9	63.7
YOLO-MF2(Ours)	**3.3**	9.6	**97.9**	**95.5**

pronounced under the stricter mAP_{50-95} metric, reflecting stronger robustness to high IoU thresholds and finer localization.

D. Model Efficiency and Latency Analysis

Table XIV highlights a clear efficiency–latency trade-off across lightweight detectors. All inference experiments were conducted on an NVIDIA A800 GPU. Each model was evaluated using four CPU threads and the same test samples. YOLO-MF2 uses the fewest parameters at 0.8M with 2.8G FLOPs and delivers 19.3 ms latency, yielding a strong balance of compactness and responsiveness. YOLOv5n achieves the lowest latency at 13.2 ms but requires 1.9M parameters and 4.6G FLOPs. YOLOv10n reaches 16.1 ms with a larger compute budget of 9.1G FLOPs and 2.8M parameters. PP-YOLOE-Lite incurs substantial computation at 17.4G FLOPs yet runs at 20.6 ms, while NanoDet-m, though extremely small at 0.9M parameters and 0.7G FLOPs, is slower at 24.8 ms. EfficientDet-Lite records 23.0 ms with 3.9M parameters and 2.5G FLOPs. YOLOv8n, YOLOv11n, and YOLOv12n show higher latency, with 31.6 ms, 54.4 ms, and 53.9 ms respectively. Overall, YOLO-MF2 offers a favorable efficiency–latency profile by combining the smallest model size with sub-20 ms inference.

E. Qualitative Study and Visualization

Given that UEC Food256 is an extension of the UEC Food100 dataset, we focused our visualization experiments solely on the UEC Food256 and ZSFooD datasets. To visually demonstrate the performance advantages of our method, Fig. 9 presents heatmaps generated using the Grad-CAM method [56] for both the baseline model and our proposed approach. The results clearly indicate that YOLO-MF2 effectively detects

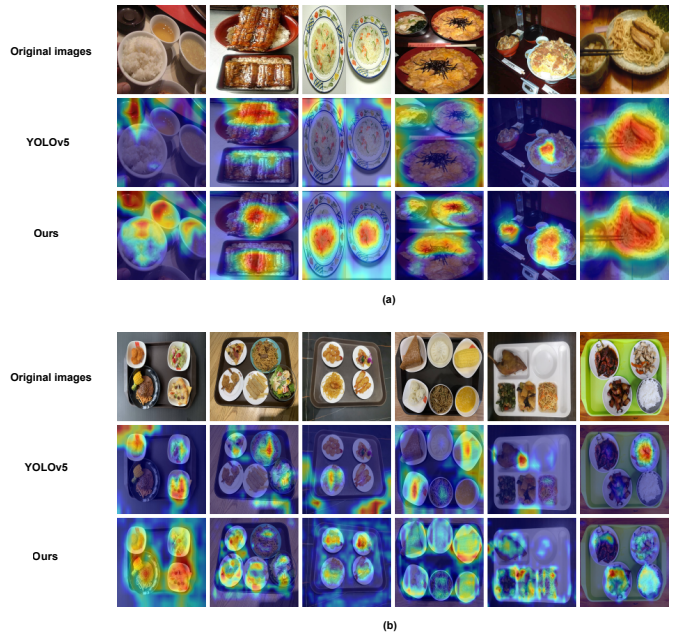

Fig. 9. Attention heatmap visualization: (a) Sample images from UEC Food256; (b) Sample images from ZSFooD. Highlighted regions indicate areas of maximum network attention concentration.

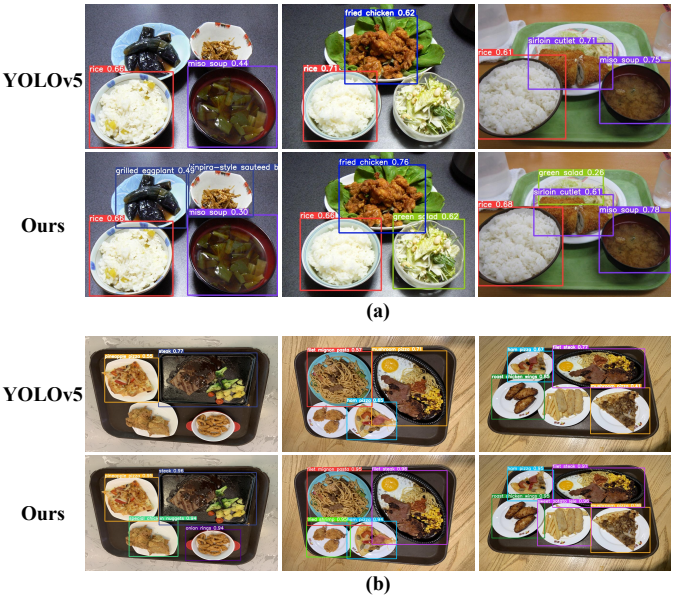

Fig. 10. Detection result visualization: (a) Examples predictions on UEC Food256; (b) Examples predictions on ZSFooD. Bounding boxes and confidence scores demonstrate model performance across diverse food categories.

all food items within an image. This superiority is attributed to our enhanced ability to extract food image features. We've designed various modules to effectively improve the model's localization and recognition performance by extracting local, multi-scale, and global features from food images. Fig. 10 showcases the visualization results of YOLO-MF2 on both the UEC Food256 and ZSFooD datasets. As observed, our model exhibits no instances of missed localization and recognitions or false positives.

TABLE XIII
COMPARISON OF YOLO-MF2 AND VARIOUS LIGHTWEIGHT MODELS IN PARAMETERS, FLOPS, AND LATENCY.

Method	UEC Food100		UEC Food256		ZSFooD	
	mAP$_{50}$	**mAP$_{50-95}$**	**mAP$_{50}$**	**mAP$_{50-95}$**	**mAP$_{50}$**	**mAP$_{50-95}$**
YOLOv5n	64.0±0.3	43.9±0.2	60.2±0.2	44.1±0.2	96.6±0.4	93.4±0.3
YOLO-MF2(Ours)	**67.6±0.4**	**48.6±0.4**	**63.66±0.5**	**48.1±0.3**	**96.8±0.4**	**93.8±0.3**

TABLE XIV
COMPARISON OF YOLO-MF2 AND VARIOUS LIGHTWEIGHT MODELS IN PARAMETERS, FLOPS, AND LATENCY.

Method	Params(M)	FLOPs(G)	Latency(ms)
YOLOv11n	2.7	6.8	54.4
YOLOv12n	2.6	6.8	53.9
YOLOv8n	3.2	8.7	31.6
NanoDet-m	0.9	**0.7**	24.8
EfficientDet-Lite	3.9	2.5	23.0
PP-YOLOE-Lite	7.9	17.4	20.6
YOLO-MF2(Ours)	**0.8**	2.8	19.3
YOLOv10n	2.8	9.1	16.1
YOLOv7-tiny	6.3	14.0	14.0
YOLOv5n	1.9	4.6	**13.2**

TABLE XV
ABLATION ANALYSIS OF YOLO-MF2 COMPONENTS ON UEC FOOD100, UEC FOOD256, AND ZSFOOD. CONV: STANDARD 3×3 CONVOLUTION BASELINE; W/O [X]: MODEL TRAINED WITHOUT MODULE [X]; [A] → [B]: MODULE A REPLACED BY MODULE B.

Dataset	Ablation	Params (M)	FLOPs (G)	mAP$_{50}$ (%)	mAP$_{50-95}$ (%)
UEC Food100	**YOLO-MF2**	**3.0**	**8.6**	**77.4**	**57.5**
	w/o GB	2.9	7.8	76.5	56.3
	MSFB→Conv	3.1	8.3	75.3	56.0
	w/o B-ViT	2.9	8.4	76.5	56.6
	w/o ICE	3.0	8.6	75.8	56.3
UEC Food256	**YOLO-MF2**	**3.4**	**9.9**	**74.5**	**56.8**
	w/o GB	3.3	9.1	73.6	56.3
	MSFB→Conv	3.5	9.6	72.9	56.2
	w/o B-ViT	3.3	9.8	72.4	55.6
	w/o ICE	3.4	9.9	73.6	56.2
ZSFooD	**YOLO-MF2**	**3.3**	**9.6**	**97.9**	**95.5**
	w/o GB	3.2	8.8	97.7	95.1
	MSFB→Conv	3.4	9.3	97.7	95.2
	w/o B-ViT	3.2	9.5	97.7	94.9
	w/o ICE	3.3	9.6	97.7	95.3

F. Ablation study

We conducted comprehensive module ablation experiments on UEC Food100, UEC Food256, and ZSFooD to evaluate component contributions. Results are summarized in Table XV.

GB Module Analysis. To validate the importance of local features for the accurate recognition of food image categories, we removed the GB module from the network and conducted experiments on three datasets. On the UEC Food100 dataset, the model without the GB module exhibited a 0.9% reduction in mAP$_{50}$ and a 1.2% reduction in mAP$_{50-95}$ compared to our proposed model. On the UEC Food256 dataset, the model without the GB module showed a 0.9% decrease in mAP$_{50}$ and a 0.5% decrease in mAP$_{50-95}$ relative to our model. On the ZSFooD dataset, the corresponding reductions were 0.2% in mAP$_{50}$ and 0.4% in mAP$_{50-95}$. These results underscore the critical role of extracting local features from food images in enhancing the accuracy of food image recognition.

MSFB Module Analysis. We replaced MSFB with standard convolutional layers. As shown in Table 10, on UEC Food100, ablation reduced mAP$_{50}$ by 2.1% and mAP$_{50-95}$ by 1.5%; on UEC Food256, mAP$_{50}$ decreased 1.6% and mAP$_{50-95}$ decreased 0.6%; on ZSFooD, both metrics declined 0.2-0.3%. These statistically significant reductions confirm that MSFB's multi-scale extraction capability is essential for handling food appearance variations.

B-ViT module analysis. Ablation of the B-ViT resulted in significant performance degradation across all datasets: on UEC Food100, mAP$_{50}$ and mAP$_{50-95}$ decreased uniformly by 0.9%; on UEC Food256, mAP$_{50}$ dropped 2.1% with mAP$_{50-95}$ declining 1.2%; while ZSFooD exhibited reductions of 0.2% and 0.6% in mAP$_{50}$ and mAP$_{50-95}$ respectively. This performance deterioration demonstrates B-ViT's critical role in capturing global food context, particularly for occluded ingredients.

ICE Module Analysis. To validate the effectiveness of the ICE module, we conducted ablation studies on all three benchmark datasets by removing ICE from YOLO-MF2. On UEC Food100, ICE removal reduced mAP$_{50}$ by 1.6% and mAP$_{50-95}$ by 1.2%. For UEC Food256, we observed 0.9% and 0.6% decreases in mAP$_{50}$ and mAP$_{50-95}$ respectively. Finally, on ZSFooD, both metrics decreased by 0.2%. These results demonstrate that ICE's context-aware feature reorganization significantly enhances localization and recognition performance in food recognition tasks.

V. DISCUSSION

This study introduces a novel lightweight framework for food localization and recognition, YOLO-Multi Feature Fusion (YOLO-MF2), designed to address the challenges of food image analysis on edge devices. By synergistically integrating local detail, multi-scale feature, and global context modeling into the YOLOv5 architecture, YOLO-MF2 demonstrates exceptional performance across multiple benchmark datasets.

In this paper, we designed the Multi-Scale Feature Bottleneck (MSFB) module, which leverages cascaded depthwise convolutions to fuse multi-scale receptive fields at a minimal parameter cost, crucial for identifying foods of varying sizes and morphologies. Furthermore, we proposed the Bidirectional Vision Transformer (B-ViT) module, which employs an axial attention mechanism to efficiently capture long-range dependencies, such as those among dispersed ingredients,

without incurring the high computational overhead of traditional ViTs. In particular, B-ViT operates without positional encoding or patch embedding, thus preserving the original spatial information of the image. To fully integrate the diverse information extracted from food images, we developed the Information Cross-Exchange (ICE) module, which facilitates information flow between different feature streams through grouping and cross-exchange reorganization, significantly improving the model's final recognition accuracy.

We conducted extensive experiments on three public datasets. It is worth noting that our model achieves a high mAP of nearly 96% on the ZSFooD dataset. We clarify that this result is not obtained under a zero-shot setting. We attribute this superior performance to the relatively low image variability and predominantly well-lit conditions in the ZSFooD dataset. Compared to state-of-the-art YOLOv12 [36], YOLO-MF2 models at various scales exhibit comparable or even smaller parameter counts and FLOPs while consistently maintaining higher accuracy. The low parameter and low computation characteristics of YOLO-MF2 make it highly suitable for deployment on edge devices, with different model scales catering to diverse task requirements. This work is expected to revolutionize traditional dietary assessment, which is based on manual and often inaccurate methods such as user-recorded nutritionist interviews. An efficient and precise food detection model that can be deployed on smartphones can enable automated dietary tracking, real-time nutritional tracking, and personalized nutrition advice through a simple photograph, providing a core technological pillar for modern health management.

This study prioritizes the development of a lightweight architecture with low computational complexity, with less emphasis on optimizing inference latency. Consequently, while YOLO-MF2 demonstrates significant advantages in parameter count and FLOPs, its inference time may not be as competitive as other state-of-the-art methods. Additionally, our error analysis reveals that YOLO-MF2 may fail to detect extremely small objects, such as tiny decorative elements. In future work, we plan to address these limitations from two perspectives: first, by further optimizing inference speed to ensure the model is not only lightweight but also faster during deployment; second, by enhancing the detection head and integrating semantic information to improve accuracy on small object detection. Beyond these two directions, we will also explore cross-domain detection experiments to more rigorously verify and enhance the generalization of model.

VI. Conclusion

We proposed YOLO-MF2, a novel lightweight architecture for food image localization and recognition. The framework integrates four key innovations: (1) A Ghost Bottleneck-enhanced backbone that preserves local features while maintaining lightweight; (2) A Multi-Scale Feature Bottleneck (MSFB) enabling hierarchical feature extraction with minimal parameters; (3) A dimension-decoupled Bidirectional Vision Transformer (B-ViT) for efficient global context modeling; and (4) An Information Cross-Exchange (ICE) module that optimizes feature fusion through cross-exchange reorganization. Extensive evaluations on UEC Food100, UEC Food256, and ZSFooD demonstrate YOLO-MF2's state-of-the-art performance among lightweight detectors, achieving mAP improvements of 0.2-3.0% while reducing parameters by 4.7-5.7M and computation by 42.0-44.6 GFLOPs compared to YOLO baselines. Future work will explore inference acceleration through hardware-aware optimizations including neural reparameterization and quantization.

References

[1] Y. Lu, T. Stathopoulou, M. F. Vasiloglou, S. Christodoulidis, Z. Stanga, and S. G. Mougiakakou, "An artificial intelligence-based system to assess nutrient intake for hospitalised patients," *IEEE Trans. Multim.*, vol. 23, pp. 1136–1147, 2021.

[2] A. Ishino, Y. Yamakata, H. Karasawa, and K. Aizawa, "Recipelog: Recipe authoring app for accurate food recording," in *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021* (H. T. Shen, Y. Zhuang, J. R. Smith, Y. Yang, P. César, F. Metze, and B. Prabhakaran, eds.), pp. 2798–2800, ACM, 2021.

[3] A. Rostami, N. Nagesh, A. Rahmani, and R. C. Jain, "World food atlas for food navigation," in *Proceedings of the 7th International Workshop on Multimedia Assisted Dietary Management, MADiMa 2022, Lisboa, Portugal, 10 October 2022* (S. G. Mougiakakou, G. M. Farinella, K. Yanai, and D. Allegra, eds.), pp. 39–47, ACM, 2022.

[4] A. Rostami, V. Pandey, N. Nag, V. Wang, and R. C. Jain, "Personal food model," in *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020* (C. W. Chen, R. Cucchiara, X. Hua, G. Qi, E. Ricci, Z. Zhang, and R. Zimmermann, eds.), pp. 4416–4424, ACM, 2020.

[5] Y. Yamakata, A. Ishino, A. Sunto, S. Amano, and K. Aizawa, "Recipe-oriented food logging for nutritional management," in *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022* (J. Magalhães, A. D. Bimbo, S. Satoh, N. Sebe, X. Alameda-Pineda, Q. Jin, V. Oria, and L. Toni, eds.), pp. 6898–6904, ACM, 2022.

[6] T. Yao, G. Wang, L. Yan, X. Kong, Q. Su, C. Zhang, and Q. Tian, "Online latent semantic hashing for cross-media retrieval," *Pattern Recognit.*, vol. 89, pp. 1–11, 2019.

[7] J. Chen, C. Ngo, and T. Chua, "Cross-modal recipe retrieval with rich food attributes," in *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017* (Q. Liu, R. Lienhart, H. Wang, S. K. Chen, S. Boll, Y. P. Chen, G. Friedland, J. Li, and S. Yan, eds.), pp. 1771–1779, ACM, 2017.

[8] J. Chen, C. Ngo, F. Feng, and T. Chua, "Deep understanding of cooking procedure for cross-modal recipe retrieval," in *2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018* (S. Boll, K. M. Lee, J. Luo, W. Zhu, H. Byun, C. W. Chen, R. Lienhart, and T. Mei, eds.), pp. 1020–1028, ACM, 2018.

[9] C. Trattner, M. Rokicki, and E. Herder, "On the relations between cooking interests, hobbies and nutritional values of online recipes: Implications for health-aware recipe recommender systems," in *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization, UMAP 2017, Bratislava, Slovakia, July 09 - 12, 2017* (M. Bieliková, E. Herder, F. Cena, and M. C. Desmarais, eds.), pp. 59–64, ACM, 2017.

[10] W. Min, L. Liu, Z. Luo, and S. Jiang, "Ingredient-guided cascaded multi-attention network for food recognition," in *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019* (L. Amsaleg, B. Huet, M. A. Larson, G. Gravier, H. Hung, C. Ngo, and W. T. Ooi, eds.), pp. 1331–1339, ACM, 2019.

[11] J. M. Rodríguez-De-Vera, I. G. Estepa, M. Bolaños, B. Nagarajan, and P. Radeva, "Lofi: Long-tailed fine-grained network for food recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 3750–3760, June 2024.

[12] G. Sheng, W. Min, T. Yao, J. Song, Y. Yang, L. Wang, and S. Jiang, "Lightweight food image recognition with global shuffle convolution," *IEEE Transactions on AgriFood Electronics*, 2024.

[13] S. Ren, K. He, R. B. 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.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[15] Z. Cai and N. Vasconcelos, "Cascade R-CNN: delving into high quality object detection," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 6154–6162, Computer Vision Foundation / IEEE Computer Society, 2018.

[16] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," in *Proceedings of the 8th International Conference on Information Technology and Quantitative Management, ITQM 2020 & 2021, Developing Global Digital Economy after COVID-19, July 9-11, 2021, Chengdu, China* (Y. Liu, Y. Shi, Y. Wang, D. Ergu, D. Berg, J. M. Tien, J. Li, and Y. Tian, eds.), vol. 199 of *Procedia Computer Science*, pp. 1066–1073, Elsevier, 2021.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), vol. 9905 of *Lecture Notes in Computer Science*, pp. 21–37, Springer, 2016.

[18] T. Hou, C. Leng, J. Wang, Z. Pei, J. Peng, I. Cheng, and A. Basu, "Mfel-yolo for small object detection in uav aerial images," *Expert Systems with Applications*, p. 128459, 2025.

[19] L. Wang, S. Wang, B. Wang, Z. Yang, and Y. Zhang, "Jujube-yolo: a precise jujube fruit recognition model in unstructured environments," *Expert Systems with Applications*, p. 128530, 2025.

[20] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1580–1589, 2020.

[21] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," in *2012 IEEE international conference on multimedia and expo*, pp. 25–30, IEEE, 2012.

[22] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III 13*, pp. 3–17, Springer, 2015.

[23] P. Zhou, W. Min, Y. Zhang, J. Song, Y. Jin, and S. Jiang, "Seeds: Semantic separable diffusion synthesizer for zero-shot food detection," in *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 8157–8166, 2023.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[25] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[26] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[27] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[28] G. Jocher *et al.*, "Yolov5 by ultralytics (version 7.0)[computer software]," 2020.

[29] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7464–7475, 2023.

[30] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[31] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolo (version 8.0. 0)[computer software]," 2023.

[32] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, *et al.*, "Yolov10: Real-time end-to-end object detection," *Advances in Neural Information Processing Systems*, vol. 37, pp. 107984–108011, 2024.

[33] Y. Tian, Q. Ye, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," *arXiv preprint arXiv:2502.12524*, 2025.

[34] E. Aguilar, B. Remeseiro, M. Bolaños, and P. Radeva, "Grab, pay, and eat: Semantic food detection for smart restaurants," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3266–3275, 2018.

[35] W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, "A survey on food computing," *Acm Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–36, 2019.

[36] S. Aslan, G. Ciocca, D. Mazzini, and R. Schettini, "Benchmarking algorithms for food localization and semantic segmentation," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 12, pp. 2827–2847, 2020.

[37] T. Ege and K. Yanai, "Simultaneous estimation of food categories and calories with multi-task cnn," in *2017 fifteenth IAPR international conference on machine vision applications (MVA)*, pp. 198–201, IEEE, 2017.

[38] L. Rachakonda, S. P. Mohanty, and E. Kougianos, "ilog: An intelligent device for automatic food intake monitoring and stress detection in the iomt," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 2, pp. 115–124, 2020.

[39] A. Ramdani, A. Virgono, and C. Setianingsih, "Food detection with image processing using convolutional neural network (cnn) method," in *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pp. 91–96, IEEE, 2020.

[40] J. Sun, K. Radecka, and Z. Zilic, "Foodtracker: A real-time food detection mobile application by deep convolutional neural networks," *arXiv preprint arXiv:1909.05994*, 2019.

[41] W. Shimoda and K. Yanai, "Webly-supervised food detection with foodness proposal," *IEICE TRANSACTIONS on Information and Systems*, vol. 102, no. 7, pp. 1230–1239, 2019.

[42] D. Qin, C. Leichner, M. Delakis, M. Fornoni, S. Luo, F. Yang, W. Wang, C. R. Banbury, C. Ye, B. Akin, V. Aggarwal, T. Zhu, D. Moro, and A. Howard, "Mobilenetv4 - universal models for the mobile ecosystem," *CoRR*, vol. abs/2404.10518, 2024.

[43] A. Wang, H. Chen, Z. Lin, H. Pu, and G. Ding, "Repvit: Revisiting mobile CNN from vit perspective," vol. abs/2307.09283, 2023.

[44] Y. Yang, W. Min, J. Song, G. Sheng, L. Wang, and S. Jiang, "Lightweight food recognition via aggregation block and feature encoding," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 10, pp. 1–25, 2024.

[45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[46] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, and Y. Yuan, "Efficientvit: Memory efficient vision transformer with cascaded group attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14420–14430, 2023.

[47] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 11531–11539, Computer Vision Foundation / IEEE, 2020.

[48] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5693–5703, 2019.

[49] S. Mehta and M. Rastegari, "Separable self-attention for mobile vision transformers," *arXiv preprint arXiv:2206.02680*, 2022.

[50] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, 2018.

[51] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3994–4003, 2016.

[52] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[53] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781–10790, 2020.

[54] RangiLyu, "Nanodet-plus: Super fast and high accuracy lightweight anchor-free object detection model.." `https://github.com/RangiLyu/nanodet`, 2021.

[55] S. Xu, X. Wang, W. Lv, Q. Chang, C. Cui, K. Deng, G. Wang, Q. Dang, S. Wei, Y. Du, *et al.*, "Pp-yoloe: An evolved version of yolo," *arXiv preprint arXiv:2203.16250*, 2022.

[56] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.