Instance-Aware Multimodal Fusion for Fine-Grained Food Recognition with Automated Dataset Collection

Fatema Kotb and Maryam Mohamed

December 29, 2025

Abstract

Fine-grained food recognition in real-world images is challenging because dishes often appear in cluttered, multi-instance scenes, while many cuisines remain underrepresented in commonly used benchmarks. In this work, we propose an instance-aware multimodal fusion pipeline that performs dish recognition at the level of individual dish instances rather than full-image classification. Given an input image, we first localize dish instances using instance segmentation, then extract ROI-aligned visual backbone features and complementary geometric cues derived from monocular depth estimation and segmentation masks. These signals are fused through a lightweight conditional modulation head to produce per-instance predictions.

We evaluate our approach in two settings: (i) a 25-class subset of Food-101 to compare against pretrained CLIP baselines, and (ii) a custom underrepresented-cuisine dataset, focusing on an Egyptian subset, to evaluate specialization beyond web-scale categories. In addition, we implement an automated data collection and annotation workflow that enables scalable dataset construction for underrepresented cuisines with minimal manual labeling.

1 Introduction

Food recognition is a key component in applications such as dietary tracking, restaurant intelligence, and recipe retrieval. While recent vision–language foundation models (e.g., CLIP) have made large gains in zero-shot recognition through image–text alignment, their performance can degrade in realistic food scenarios for two main reasons. First, food images frequently contain multiple dish instances, partial occlusions, and strong background clutter, making full-image classification an unreliable proxy for per-dish recognition. Second, many cuisines and regional dishes are underrepresented in large-scale pretraining corpora and standard benchmarks, which limits the ability of general-purpose models to reliably distinguish fine-grained dish categories in these domains.

This work focuses on *instance-level dish recognition* and *specialization to underrepresented cuisines*. Instead of treating the input as a single global category prediction, we explicitly detect dish instances and classify each one using features extracted from the instance region. We further enrich these appearance features with geometric signals that are naturally available in food imagery scenes: segmentation masks provide structured shape and spatial support for pooling, while monocular depth maps provide complementary cues about volume, height profiles, and foreground separation. We integrate these modalities through a lightweight conditional modulation fusion head that allows instance geometry to guide which appearance channels are emphasized for classification.

In addition to the model design, we address the practical bottleneck of data availability. Because underrepresented cuisines lack large curated benchmarks, we implement an automated data collection and annotation pipeline that scales dataset creation by combining web data acquisition

with automated instance-level labeling. This enables rapid construction of training and evaluation sets beyond standard datasets, and provides a reproducible workflow for expanding to additional cuisines and dish types.

2 Literature Review

This section introduces the key models, methods, and design patterns that our system builds upon. We first explain each component at an intuitive level and then describe why it is relevant to food recognition, especially for underrepresented cuisines.

2.1 Food Recognition Datasets and the Long-Tail Problem

A central challenge in food recognition is class imbalance and long-tailed distribution: a small number of popular dishes have many images online, while many regional dishes have very few. Standard datasets such as Food-101 provide a strong benchmark for common categories, but they do not represent the full diversity of global cuisines. As a result, models trained and evaluated only on such datasets may generalize poorly to underrepresented cuisines. This motivates the creation of custom datasets and data collection pipelines tailored to the target culinary domain.

2.2 Vision–Language Pretraining and CLIP

CLIP (Contrastive Language–Image Pretraining) is a vision–language model trained to align images and text using a contrastive objective. At inference time, CLIP can perform zero-shot classification by comparing an image embedding to text embeddings of class prompts (e.g., "a photo of falafel"). This is powerful because it enables classification without explicit supervised training on the target dataset. However, CLIP performance depends heavily on how well the class concept is represented in its pretraining data. For niche dishes and local cuisines, the representation may be weak or ambiguous, leading to confusion among visually similar classes.

Prompting and prompt ensembling. A common pattern for improving CLIP is prompt engineering and prompt ensembling: writing multiple templates per class (e.g., "a photo of ...", "a close-up of ...", "a dish of ...") and averaging text embeddings. This can reduce sensitivity to prompt wording and improve stability.

2.3 Fine-Tuning and Linear Probes

A practical approach to specialize CLIP is to freeze the backbone and train a lightweight classifier on top of CLIP image embeddings (linear probing). This is data-efficient and reduces training cost compared to full fine-tuning. In our work, we fine-tune a CLIP linear head as a specialist for Egyptian dishes.

2.4 Specialist–Generalist Routing (Hybrid Models)

When a single model must cover both broad categories and niche categories, a common design pattern is a hybrid or mixture-of-experts style system:

- a generalist model handles the broad taxonomy (e.g., Food-101),
- a specialist model handles a narrower domain (e.g., Egyptian dishes),
- a router selects which output to trust, often based on confidence or calibration.

This pattern is attractive for underrepresented cuisines because it allows focused learning without sacrificing general coverage. Our Hybrid CLIP baseline follows this approach using a confidence threshold. The fine-tuned specialist is used strictly as a routing component within the hybrid baseline: it complements the general Food-101 CLIP head by improving discrimination among Egyptian dish classes, rather than serving as the primary mechanism by which the overall system addresses underrepresented classes.

2.5 Instance Segmentation and Mask R-CNN

Mask R-CNN extends Faster R-CNN by predicting a pixel-wise mask for each detected instance. Instead of classifying the entire image, instance segmentation enables object-centric recognition: we can isolate each dish in multi-dish scenes, reduce background leakage, and classify each dish independently. This is especially important when multiple plates appear in one photo or when the background is correlated with a class (dataset bias).

2.6 Region of Interest (ROI) Feature Extraction and ROIAlign

ROIAlign is a differentiable pooling operation that extracts fixed-size feature tensors from a backbone feature map given a bounding box. It is widely used in detection and segmentation pipelines to obtain per-instance representations. In our system, ROIAlign allows us to reuse a single global backbone forward pass and then compute per-instance features efficiently. ROI-based features are also a key design pattern for instance-aware classification: they focus learning on the dish region rather than the whole frame.

2.7 Monocular Depth Estimation and ZoeDepth

Monocular depth estimation predicts a dense depth map from a single RGB image. While absolute depth may be uncalibrated, relative depth and depth structure provide valuable cues:

- separating foreground dish from background surfaces,
- capturing dish volume or height profile (e.g., stacked vs. flat dishes),
- detecting occlusions and layered presentations.

ZoeDepth combines relative depth cues with a learned metric-depth prior, enabling strong zero-shot transfer across datasets. A *learned metric-depth prior* refers to a data-driven model component trained on datasets where depths are available in real-world units (metric depth). During training, the model learns statistical regularities that map image appearance to plausible absolute depth scales and distributions (e.g., typical camera heights, object sizes, and scene layouts). At inference time, this prior helps anchor predictions toward physically meaningful depth magnitudes, reducing scale ambiguity that is inherent to monocular depth estimation. In our pipeline, we summarize depth under each instance mask (statistics and ROI depth crops) to provide geometric signals to the classifier.

2.8 Multimodal Feature Fusion

Combining multiple sources of information is typically done via:

- Early fusion: concatenate inputs/features before heavy processing,

- Late fusion: combine independent model outputs (e.g., ensembling),
- Conditional fusion: use one modality to modulate another (attention/FiLM).

Food recognition benefits from conditional fusion because instance masks and depth act as structured, low-dimensional cues that can guide which visual patterns are most relevant.

2.9 FiLM and Conditional Modulation

Feature-wise Linear Modulation (FiLM) is a conditioning mechanism that transforms an intermediate representation using learned scale and shift parameters:

$$X' = \gamma(c) \odot X + \beta(c), \tag{1}$$

where c is a conditioning vector derived from an auxiliary modality. FiLM is simple, lightweight, and effective for injecting context.

In our fusion head, the auxiliary modality is the instance geometry captured by the segmentation mask and the depth crop. The design goal is to allow geometric cues to *control* which appearance features are emphasized for classification, rather than treating all visual patterns equally. Concretely, for each detected dish instance we:

1. Encode the mask–depth ROI into a compact conditioning vector h that summarizes shape and relative surface structure.
2. Map h to per-channel modulation parameters (γ, β) that match the channel dimensionality of the ROI backbone features.
3. Apply FiLM to the ROI features so that channels consistent with the instance geometry are amplified (via γ) and channels inconsistent with it are suppressed or shifted (via β).

This conditional modulation is particularly useful in fine-grained food recognition because many dish classes share similar textures and colors, while differing in geometric properties such as height profiles, mound-like vs. flat presentations, or the presence of distinct layered structures. FiLM allows these geometric cues to act as a soft, learnable gate on appearance features. The resulting representation can therefore focus on discriminative patterns *within* the dish region that are most relevant given its geometry, improving robustness to background clutter and style variations.

2.10 Automated Dataset Collection and Weak/Automatic Annotation

Underrepresented cuisines require scalable dataset creation. A common pattern is automated collection from web sources followed by weak supervision or automatic annotation. Modern foundation models enable semi-automated pipelines that:

- collect candidate images from queries and filters,
- detect relevant objects using open-vocabulary detection (e.g., DINO-style detectors),
- segment precise masks using promptable segmentation (e.g., SAM),
- optionally add human verification at the final stage.

We adopt this approach to reduce manual labeling overhead and to build a multi-country, multi-dish dataset for evaluation and training.

3 System Design

This section describes our end-to-end pipeline and the model components implemented in our codebase.

3.1 High-Level Pipeline

Given an input RGB image:

1. **Instance segmentation:** Mask R-CNN predicts candidate dish instances as masks and bounding boxes.
2. **Depth estimation:** ZoeDepth predicts a dense depth map for the full image.
3. **Backbone features:** EfficientNet extracts feature maps from the full image.
4. **Instance-aware fusion block:** for each dish instance, we compute ROI-aligned backbone features, resized mask and depth crops, and depth statistics under the instance mask.
5. **Prediction head:** a fusion classifier head produces logits over the target classes.

3.2 Instance Segmentation Block (Mask R-CNN)

We use TorchVision's pretrained Mask R-CNN (COCO) and wrap it in a block that includes additional post-processing:

- Guaranteed tiny-mask removal based on a minimum mask area ratio relative to image size.
- Duplicate and union-mask removal using IoU and containment checks to prefer separate instances (A, B) over merged detections (A+B).

3.3 Depth Estimation Block (ZoeDepth)

We integrate ZoeDepth NK using torch.hub, downloading the public checkpoint and producing a dense depth map. We note that output depth is typically relative/affine without calibration, but remains useful as a geometric cue.

3.4 Backbone Feature Extraction (EfficientNet)

We use an EfficientNet visual backbone to produce feature maps that are later consumed by ROIAlign to yield per-instance features. EfficientNet converts the input image into a hierarchy of feature maps with decreasing spatial resolution and increasing channel dimensionality. We use the last feature map as the ROIAlign input because it provides strong semantic features while remaining spatially structured.

ROIAlign and spatial scale. ROIAlign extracts a fixed-size tensor (here 7×7) from the backbone feature map given a bounding box defined in the input-image coordinate system. Because the feature map is downsampled relative to the input image, ROIAlign needs a *spatial scale* factor that maps image coordinates to feature-map coordinates. If the backbone produces a feature map of size (H_f, W_f) from an input image of size (H, W), then the spatial scale is approximately H_f/H (and similarly W_f/W). This factor ensures that the ROI bounding box is projected to the correct region on the feature map. ROIAlign then samples the corresponding feature region using bilinear interpolation and aggregates it into a $C \times 7 \times 7$ tensor without quantizing coordinates, preserving alignment between the instance region and the extracted features.

3.5 Instance-Aware Fusion Block

For each predicted instance with bounding box b and mask m, we construct the following tensors:

- ROI features $F_{\text{roi}} \in \mathbb{R}^{C\times 7\times 7}$ via ROIAlign on the backbone's last feature map.
- Mask ROI $M_{\text{roi}} \in \mathbb{R}^{1\times 7\times 7}$ by resizing the binary mask inside b.
- Depth ROI $D_{\text{roi}} \in \mathbb{R}^{1\times 7\times 7}$ by resizing depth within b and zeroing values outside the mask.
- Depth statistics $s \in \mathbb{R}^8$ computed over full-resolution depth values under the mask (mean, std, MAD, and percentiles).

How resizing is done and why. The instance mask m and the depth map are originally defined at the input-image resolution, while F_{roi} is extracted as a fixed 7×7 grid. To make mask and depth cues spatially compatible with ROI features, we crop the full-resolution mask and depth to the instance bounding box b and resize each crop to 7×7 using bilinear interpolation for depth and bilinear interpolation followed by thresholding (or nearest-neighbor interpolation) for the binary mask. This produces M_{roi} and D_{roi} with the same spatial dimensions as the ROI feature tensor. Aligning all modalities to a common 7×7 grid enables elementwise operations (e.g., masked pooling) and allows the mask–depth encoder to reason about the instance geometry at the same spatial granularity as the backbone features. The 7×7 size is a standard ROI resolution that balances spatial detail and computational efficiency, while remaining compatible with common detection architectures and stable for batching across variable-sized instances.

These outputs form the FusionOutput used for training and inference.

3.6 Fusion Classification Head

Our InstanceAwareFusionHead takes F_{roi}, M_{roi}, D_{roi}, and s and predicts logits over K classes. The head consists of:

1. **Mask+Depth encoder:** concatenate M_{roi} and D_{roi}, encode with a small CNN into a 64D vector h.
2. **FiLM modulation:** map h to per-channel (γ, β), then modulate ROI features:

$$X = F_{\text{roi}} \odot (1 + \gamma) + \beta. \tag{2}$$

3. **Masked average pooling:** pool X under M_{roi} to obtain a C-dimensional instance feature.
4. **MLP classifier:** concatenate pooled feature with s (depth stats), then pass through an MLP to produce logits.

Mask+Depth encoder design rationale. The concatenated mask and depth ROI form a small spatial tensor that encodes instance shape (mask) and coarse surface layout (depth). A compact CNN is a natural choice for this input because it can capture local spatial patterns such as boundaries, thickness cues, and simple depth gradients that correlate with dish presentation (e.g., layered vs. flat structures). Unlike an MLP over flattened pixels, a CNN preserves neighborhood structure and yields a representation that is translation-tolerant within the ROI grid. The resulting embedding is intentionally low-dimensional to act as a conditioning signal rather than a competing classifier: it should summarize geometry in a way that is easy to map to FiLM parameters without overfitting. We use a 64D vector as a practical balance between expressiveness and stability; it is

large enough to encode diverse geometric configurations across dishes while remaining small enough to keep the conditioning pathway lightweight and regularized.

Masked average pooling (how and why). After FiLM modulation, we obtain modulated ROI features $X \in \mathbb{R}^{C\times 7\times 7}$. To convert this spatial tensor into a single per-instance vector while explicitly restricting aggregation to the dish region, we compute a mask-weighted mean over spatial locations:

$$f_c = \frac{\sum_{u,v} M_{\text{roi}}(u,v)\, X_c(u,v)}{\sum_{u,v} M_{\text{roi}}(u,v) + \epsilon}, \quad c = 1, \ldots, C, \tag{3}$$

where (u, v) index the 7×7 grid and ϵ prevents division by zero for degenerate masks. This pooling operation removes background influence by excluding pixels outside the predicted dish mask and yields a robust instance descriptor even when bounding boxes include clutter. It also provides a consistent, fixed-dimensional feature for the classifier regardless of the instance size, while retaining the benefits of the instance segmentation signal during aggregation.

3.7 Baselines

3.7.1 Pretrained CLIP (Food-101)

For Benchmark 1, we compare our fusion pipeline against pretrained CLIP using Food-101 prompts for the 25-class subset.

3.7.2 Hybrid CLIP Baseline

For Benchmark 2, we compare against Hybrid CLIP, which runs:

- A general Food-101 CLIP head, and
- A specialist fine-tuned CLIP linear classifier trained on Egyptian dish embeddings.

A confidence-threshold rule returns the Egyptian prediction if its confidence exceeds τ; otherwise it returns the general prediction.

3.8 Automated Dataset Collection and Annotation

To address data scarcity for underrepresented cuisines, we built an automated pipeline for dataset collection and labeling as part of our implementation. The workflow includes automated web data acquisition, filtering, and an annotation stage that combines open-vocabulary detection (DINO-style) with promptable segmentation (SAM-style) to generate high-quality instance masks for food images. This automation reduces manual annotation effort and enables scaling to multiple countries and cuisines.

4 Experimental Setup

4.1 Benchmarks

Benchmark 1: Food-101 (25-class subset). We evaluate on a 25-class subset, comparing:

1. pretrained CLIP baseline, and
2. our instance-aware fusion pipeline.

Benchmark 2: Underrepresented cuisines dataset (Egyptian 5-dish subset). We evaluate on our custom dataset of underrepresented cuisines, focusing on a 5-dish Egyptian subset, comparing:

1. Hybrid CLIP baseline, and
2. our instance-aware fusion pipeline.

4.2 Training Procedure

The fusion head is trained on cached per-instance tensors produced by the segmentation, depth, and backbone blocks. For each training image, we run the full pipeline once to obtain $(F_{\text{roi}}, M_{\text{roi}}, D_{\text{roi}}, s)$ for every detected dish instance, then serialize these instance-level tensors to disk as sharded files and index them with JSONL metadata. This decouples expensive feature computation from the classifier training loop and allows efficient minibatch sampling across instances.

During training, we load batches of instances, each labeled with the ground-truth dish class, and optimize the fusion head parameters end-to-end (while keeping Mask R-CNN, ZoeDepth, and EfficientNet frozen). We use cross-entropy loss over the K target classes and the AdamW optimizer for stable convergence under limited data. The learning rate schedule and weight decay are set to regularize the lightweight head and reduce overfitting to the small fine-grained benchmarks. We split the data into training and validation sets at the instance level (ensuring that images are not shared across splits) and select the final checkpoint based on the highest validation top-1 accuracy. This selection criterion directly matches the primary evaluation metric and provides a consistent basis for comparing our approach against CLIP baselines.

4.3 Evaluation Metrics

We report top-1 accuracy for each benchmark. Top-1 accuracy measures the fraction of instances for which the model's highest-confidence predicted class matches the ground-truth label.

5 Results

5.1 Benchmark 1: Food-101 (25-class subset)

Table 1 compares pretrained CLIP against our pipeline.

Table 1: Benchmark 1 results on Food-101 (25-class subset).

Method	Top-1 Acc. (%)
Pretrained CLIP	[fill]
Instance-aware fusion (ours)	[fill]

5.2 Benchmark 2: Underrepresented cuisines (Egyptian 5-dish subset)

Table 2 compares Hybrid CLIP against our pipeline.

Table 2: Benchmark 2 results on Egyptian 5-dish subset.

Method	Top-1 Acc. (%)
Hybrid CLIP baseline	[fill]
Instance-aware fusion (ours)	[fill]

5.3 Discussion

We expect the fusion pipeline to outperform CLIP-only baselines in scenarios where:

- multiple dishes appear in the same image (instance awareness),
- background/plate bias is strong (mask-based pooling),
- geometric cues help distinguish dish shape/volume (depth statistics),
- training data is limited but structured features provide a useful inductive bias.

An *inductive bias* is a built-in preference of a learning method toward certain kinds of solutions or patterns, which helps it generalize from limited data. In our setting, explicitly providing structured geometric signals (mask shape and depth summaries) biases the model toward using physically meaningful instance cues, reducing reliance on spurious background correlations and improving sample efficiency.

Hybrid CLIP provides a strong baseline for specialization, but it does not explicitly incorporate geometric cues and relies on confidence thresholding to route predictions.

6 Conclusion

We presented an instance-aware multimodal fusion pipeline for food recognition that combines segmentation, depth estimation, and ROI-aligned visual features. It is considered *multimodal* because the model integrates multiple complementary sources of information: RGB appearance features from the visual backbone, geometric depth cues from monocular depth estimation, and structured instance shape cues from segmentation masks. We evaluated on (i) a Food-101 subset and (ii) a custom underrepresented-cuisine dataset (Egyptian subset), comparing against CLIP baselines. In addition, we described an automated data collection and annotation workflow that supports scaling to multiple cuisines and countries. Future work includes (i) expanding the dataset to cover a larger set of Egyptian dishes and additional underrepresented cuisines with consistent annotation protocols, (ii) improving robustness in multi-dish scenes by incorporating stronger instance filtering and association across images (e.g., tracking or deduplication for near-identical instances), (iii) ablations and calibration studies to better understand when depth and mask cues contribute most, and (iv) enabling end-to-end fine-tuning of the fusion head together with selected backbone layers to better align ROI appearance features with geometric conditioning under limited supervision.

References

1. Bossard, L., Guillaumin, M., Van Gool, L. Food-101 – Mining Discriminative Components with Random Forests. In ECCV, 2014.

2. Radford, A. et al. Learning Transferable Visual Models From Natural Language Supervision. In ICML, 2021.

3. He, K., Gkioxari, G., Dollár, P., Girshick, R. Mask R-CNN. In ICCV, 2017.

4. Tan, M., Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In ICML, 2019.

5. Bhat, S.F. et al. ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth. 2023.

6. Kirillov, A. et al. Segment Anything. 2023.

7. Carion, N. et al. End-to-End Object Detection with Transformers. In ECCV, 2020.