KARINA: Knowledge-Augmented Representation for Ingredient-level Nutrition Analysis from Food Images

Chieh-Yu Pan and Wei-Ta Chu

***Abstract*— We leverage nutrition-related knowledge from large multimodal models (LMMs) to enhance nutrition estimation. Although directly applying an LMM to nutrition estimation is not reliable, an LMM can provide ingredient-level descriptions with rough nutrition profiles. Therefore, we propose a knowledge-augmented framework that integrates LMM-generated nutritional descriptions into a visual model through a cross-modal attention mechanism. This framework includes a multimodal feature fusion module and a mask-based data augmentation strategy for fine-grained alignment between ingredient regions and textual descriptions. Experimental results show that our approach achieves state-of-the-art performance across the Nutrition5K, NutritionVerse-Real, and NutritionVerse-Synth benchmarks.**

I. INTRODUCTION

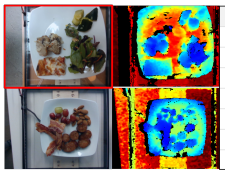
With increasing awareness toward personal health management, accurate tracking of nutritional intake has become an important aspect of daily life. However, precise nutritional information is not widely accessible. Computer vision approaches that estimate nutrition values directly from food images were therefore proposed.

Recently, Google introduced the Nutrition5K dataset [1]. This dataset contains food dish images captured from both top-view and side-view perspectives. The top-view images are with the corresponding depth information. The nutrition annotations include dish-level and ingredient-level nutrition values, including calories, mass, fat, carbohydrates, and protein, as illustrated in Figure 1. Due to its comprehensive annotations and structured format, Nutrition5K has become the most important benchmark to facilitate image-based nutrition estimation.

Many approaches focused exclusively on RGB image information to perform nutrition prediction [2][3]. Recent works demonstrated that incorporating depth information leads to better estimation accuracy [1][4][5][6][7]. Furthermore, works in [8] [9] showed that incorporating ingredient-level labels (ingredient names) as supervisory signals further enhanced performance. Despite these improvements, these methods overlooked some inherent nutritional characteristics. For instance, meats are generally high in protein, whereas fruits and starchy foods typically contain higher carbohydrates. This explicit nutritional knowledge remains underutilized.

Meanwhile, large multimodal models (LMMs) have demonstrated impressive capabilities in various vision-language tasks. How about directly prompting GPT-4o to estimate nutrition values from food images? Our preliminary results show that GPT-4o yields estimation errors of approximately 40% in terms of percentage mean absolute error (PMAE), considerably underperforming dedicated estimation models [1] [4]. On the other hand, we observe that GPT-4o can accurately identify food items and give general nutritional descriptions. For example, it recognizes that high protein content per gram is for chicken and high carbohydrate content per gram is for bananas. This observation suggests an opportunity to utilize LMM-generated nutrition knowledge.

Authors are with Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan chiehyu.pan@gmail.com, wtchu@gs.ncku.edu.tw



Food item	Mass (g)	Cal (kcal)	Fat (g)	Carbs (g)	Protein (g)
Total	216	361.8	19.9	23.6	22.2
Pizza	51	135.6	4.8	17.1	5.7
Arugula	10.5	2.6	0.1	0.4	0.3
Zucchini	48	7.2	0.2	1.3	0.5
Chicken	57.1	147.3	9.7	0	14.3
Olive oil	3.6	31.6	3.6	0	0
Garlic	0.1	0.1	0	0	0

Fig. 1. **Some samples from the Nutrition5k dataset.** Left: Top-view RGB and corresponding depth images for two food dishes. Right: Ground truth nutrition values for the dish in the red bounding box and for a subset of its ingredients, including mass (g), calories (kcal), fat (g), carbohydrates (g), and protein (g).

In this work, we introduce Knowledge-Augmented Representation for Ingredient-level Nutrition Analysis (KARINA). We take LMM-generated nutrition descriptions as external supervisory signals and propose a multimodal fusion module to enhance representation learning. The main contributions of this work include:

- Nutrition knowledge from LMMs: We prompt LMMs to generate ingredient-level descriptions and qualitative nutritional profiles. These textual outputs are then incorporated as weak semantic guidance to enhance visual representation learning.
- Fine-grained image-text alignment: We introduce a mask-based ingredient-level augmentation strategy to enable ingredient-level alignment. This approach leverages segmentation masks obtained from a pre-trained foundation model to establish fine-grained correspondences between individual ingredients and their respective nutrition descriptions.

II. METHODOLOGY

The framework of KARINA is illustrated in Figure 2.

A. Visual Feature Extraction

We first extract visual features from the RGB and depth images by two Swin Transformers [10], respectively. The Swin Transformers were pre-trained on ImageNet and are

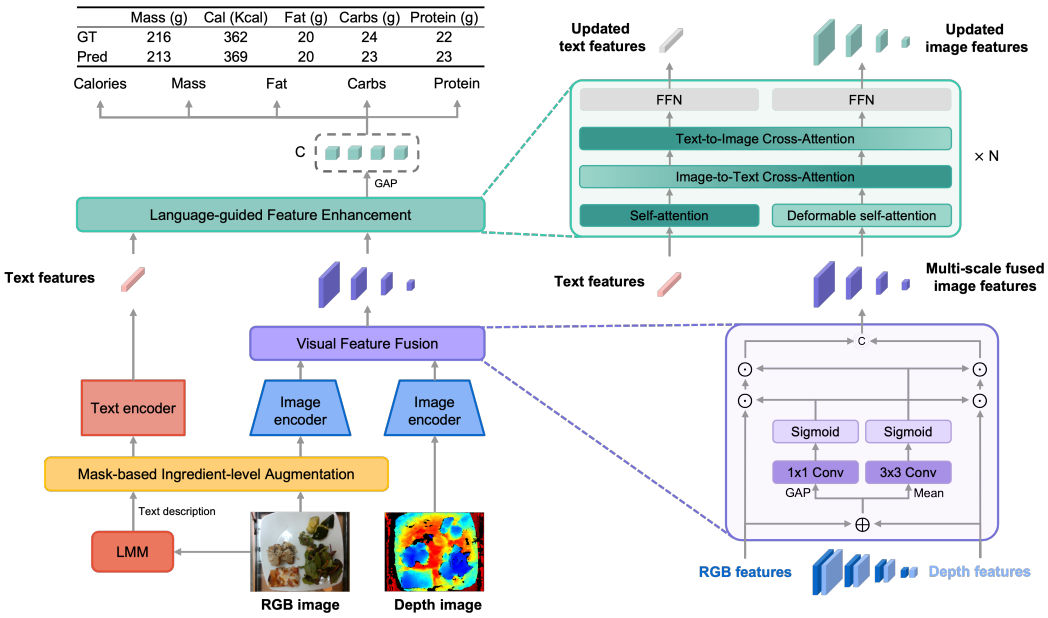


Fig. 2. The framework of KARINA, including visual feature fusion, language-guided feature enhancement, and mask-based ingredient-level augmentation.

fine-tuned during training. Each Swin Transformer processes the input images through four stages. The image is first partitioned into non-overlapping patches of size 4×4, and the initial embedding dimension is set to C_1. This results in an initial feature map of shape $(\frac{H}{4}, \frac{W}{4}, C_1)$. After each stage, the spatial resolution is downsampled by a factor of 2 in both height and width, and the channel dimension doubles progressively, resulting in channel sizes of $\{C_1, 2C_1, 4C_1, 8C_1\}$. We take the output feature maps from all four stages as multi-scale visual representations $\mathcal{V} = \{\mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \mathbf{V}^{(3)}, \mathbf{V}^{(4)}\}$.

Next, we construct a visual feature fusion (VFF) module as in [4] to integrate RGB and depth information at multiple scales. The VFF module adopts a CBAM-like attention mechanism [11]. Let $\mathbf{V}_{\text{rgb}}^{(i)} \in \mathbb{R}^{H_i \times W_i \times C}$ and $\mathbf{V}_{\text{dep}}^{(i)} \in \mathbb{R}^{H_i \times W_i \times C}$ denote the RGB and depth feature maps from the ith stage of the vision encoder, respectively. We first project both features into a shared embedding space and then fuse them by element-wise addition: $\mathbf{V}_{\text{fused}}^{(i)} = \mathbf{V}_{\text{rgb}}^{(i)} \oplus \mathbf{V}_{\text{dep}}^{(i)}$.

To compute the channel attention weights, we perform global average pooling (GAP) over spatial dimensions, followed by a 1×1 convolution with GroupNorm (GN) [12], ReLU, and a sigmoid activation.

$$\mathbf{z}_c^{(i)} = \text{GAP}(\mathbf{V}_{\text{fused}}^{(i)}), \quad \mathbf{M}_c^{(i)} = \sigma(\text{Conv}_{1\times1}(\mathbf{z}_c^{(i)})), \tag{1}$$

where $\text{Conv}_{1\times1}$ denotes a 1×1 convolution.

For spatial attention, we first compute the mean values across channels, and then apply a 3×3 convolution with GN, ReLU, and a sigmoid activation:

$$\mathbf{z}_s^{(i)} = \text{Mean}_c(\mathbf{V}_{\text{fused}}^{(i)}), \quad \mathbf{M}_s^{(i)} = \sigma(\text{Conv}_{3\times3}(\mathbf{z}_s^{(i)})), \tag{2}$$

where $\text{Mean}_c(\cdot)$ denotes taking average across channels.

The RGB and depth features are then refined independently using the shared attention weights:

$$\mathbf{V}'^{(i)}_{\text{rgb}} = \mathbf{V}^{(i)}_{\text{rgb}} \odot \mathbf{M}_c^{(i)} \odot \mathbf{M}_s^{(i)}, \quad \mathbf{V}'^{(i)}_{\text{depth}} = \mathbf{V}^{(i)}_{\text{depth}} \odot \mathbf{M}_c^{(i)} \odot \mathbf{M}_s^{(i)}, \tag{3}$$

where $\odot$ denotes element-wise multiplication.

Finally, the refined RGB and depth features are concatenated along the channel dimension and projected to produce the fused output: $\tilde{\mathbf{V}}^{(i)} = \left[\mathbf{V}'^{(i)}_{\text{rgb}}, \mathbf{V}'^{(i)}_{\text{depth}}\right]$, where $[\cdot,\cdot]$ denotes channel-wise concatenation.

This fusion process is applied independently at each stage $i \in \{1, 2, 3, 4\}$, resulting in a multi-scale fused feature set $\tilde{\mathcal{V}} = \{\tilde{\mathbf{V}}^{(i)}\}_{i=1}^{4}$.

B. Text Feature Extraction

We employ GPT-4o[1] to generate ingredient-level descriptions for food images. The following prompt is designed to obtain names, descriptions (including quantity and visual appearance), and nutrition profiles of visible ingredients:

> *Generate names of all visible ingredients in the food image, along with a short description (including quantity and visual appearance) and a nutrition profile for each ingredient. The nutrition profile should indicate the levels of calories, fat, carbohydrates, and protein using qualitative categories (high, medium, low). The response should follow this structure: ...*

The generated descriptions are structured as a JSON object containing up to five prominent ingredient entities. Each entity includes a name, a brief description, and a qualitative nutritional profile. An example is illustrated in Figure 3.

We represent each food entity by a structured string:

$$E_i = \texttt{Desc}_i \;||\; \left(||_k \; N_k^i \right), \tag{4}$$

[1]Note that we mainly use GPT-4o as the studied LMM in this paper. Other closed and open-sourced LMMs can also be used in the same manner.

{*Name: pizza; Desc: 1 slice of pizza; Nutrition: calorie–high, fat–high, carb–high, protein–medium*}.

{*Name: zucchini; Desc: small portion of green zucchini; Nutrition: calorie–low, fat–low, carb–medium, protein–low*}.

Fig. 3. A sample food image and its corresponding ingredient-level descriptions generated by GPT-4o.

where || denotes string concatenation, $\texttt{Desc}_i$ denotes a short description of the ingredient i, and N_k^i represents the nutrition level of calories, fat, carbs, and protein for the ingredient i. For example, the description for a pizza entity is expressed as:

E_{pizza} = *"a slice of pizza high calorie, high fat, high carb, medium protein."*

To avoid misalignment between different nutritions, we introduce special tokens, `[SON]` (start of nutrition) and `[EON]` (end of nutrition), at the beginning and end of each nutrition profile within the text description. The special tokens serve as a delimiter, guiding the model to treat each nutritional phrase (e.g., "high fat") separately. As a result, the final description for a pizza entity is expressed as:

E_{pizza} = *"a slice of pizza* `[SON]` *high calorie* `[EON]`, `[SON]` *high fat* `[EON]`, `[SON]` *high carb* `[EON]`, `[SON]` *medium protein* `[EON]`."

We then concatenate the descriptions of up to five entities to construct a complete description for the input food image. This description is passed into a pre-trained BERT model to produce a sequence of token embeddings, denoted as $\mathbf{T}$.

C. Language-Guided Feature Enhancement

We introduce a language-guided feature enhancement (LGFE) module to facilitate bidirectional fusion between vision and language, as illustrated in Figure 2. The LGFE module comprises four components: (1) self-attention over textual features, (2) deformable self-attention over visual features, (3) bidirectional cross-attention, and (4) feed-forward networks (FFNs).

First, the text feature $\mathbf{T}$ undergoes self-attention to capture intra-ingredient dependencies. The multi-scale visual features $\{\tilde{\mathbf{V}}^{(i)}\}_{i=1}^4$ are concatenated and flattened into a unified token sequence $\mathbf{V}$ and processed by the deformable self-attention [13] process to enhance visual information.

Then, we apply bidirectional cross-attention. In the image-to-text attention step, visual tokens $\mathbf{V}$ act as queries while text tokens $\mathbf{T}$ provide keys and values: $\hat{\mathbf{V}} = \text{Attention}(\mathbf{V}, \mathbf{T}, \mathbf{T})$. In the text-to-image attention, the roles are swapped: $\hat{\mathbf{T}} = \text{Attention}(\mathbf{T}, \mathbf{V}, \mathbf{V})$. The updated representations $\hat{\mathbf{V}}$ and $\hat{\mathbf{T}}$ are passed through two independent FFNs, respectively. The whole process is repeated N times to progressively refine the representations.

Finally, we take the enhanced multi-scale visual representation $\{\hat{\mathbf{V}}^{(i)}\}_{i=1}^4$ to do nutrition estimation.

D. Mask-based Ingredient-level Augmentation

The LGFE module does not explicitly enforce alignment between individual ingredients and their corresponding image regions, which hinders the model's ability to learn fine-grained associations. To address this limitation, we introduce a Mask-based Ingredient-level Augmentation (MIA) strategy that selectively generates ingredient-centric training samples based on segmentation masks. The augmentation allows the model to learn both dish-level semantics and ingredient-level semantics.

To generate ingredient-level masks, we employ Grounded SAM [14] that takes the ingredient names as input and produces segmentation masks conditioned on the prompts.

For each RGB image $\mathbf{I}$, we obtain: (1) a set of ingredient masks $\mathcal{M} = \{\mathbf{M}_i\}_{i=1}^n$, where $\mathbf{M}_i \in \{0,1\}^{H\times W}$ is a binary mask for the ith ingredient; (2) corresponding textual descriptions $\mathcal{T} = \{\mathbf{t}_i\}_{i=1}^n$, where $\mathbf{t}_i$ is a string describing the ith ingredient; and (3) nutrition annotations $\mathcal{A} = \{\mathbf{a}_i\}_{i=1}^n$, where $\mathbf{a}_i \in \mathbb{R}^5$ stands for the ground truth values of calories, mass, fat, carbohydrates, and protein.

During training, the MIA strategy is applied with a probability p. If the augmentation is not triggered, the model receives the full image $\mathbf{I}$ along with the concatenated textual description $\mathbf{T}$. This case is the same as Sec. II-B.

If augmentation is activated, a triplet $(\mathbf{M}_i, \mathbf{t}_i, \mathbf{a}_i)$ is randomly sampled from the augmented set. Figure 4 illustrates a sample case. A masked image is generated by applying alpha blending between the original RGB image $\mathbf{I}$ and the ingredient mask $\mathbf{M}_i$. The masked image is denoted as $\mathbf{I}_i$ and is constructed by:

$$\mathbf{I}_i = \mathbf{M}_i \odot \mathbf{I} + \alpha(1 - \mathbf{M}_i) \odot \mathbf{I}, \tag{5}$$

where $\alpha \in [0,1]$ is a blending coefficient and $\odot$ denotes element-wise multiplication. The first term preserves the region corresponding to the ingredient of interest, while the second term suppresses the surrounding region. The resulting masked image $\mathbf{I}_i$, along with the corresponding description $\mathbf{t}_i$ and label $\mathbf{a}_i$, serves as an augmented training sample.

In our implementation, we set $\alpha = 0.25$ to softly retain peripheral visual cues. The design of α serves two purposes. First, it improves robustness to imperfect masks by maintaining information of nearby regions. Second, it helps the model more effectively characterize an ingredient by considering the surrounding context. Notice that we do not apply the same masking to the depth image, as modifying the depth map may distort spatial priors related to size and volume.

E. Multi-task Nutrition Estimation

The objective is to estimate four nutrition values, i.e., calories, fat, carbohydrates, and protein, and the mass of the dish, which is formulated as a multi-task regression problem. Let $\{\hat{\mathbf{V}}^{(1)}, \hat{\mathbf{V}}^{(2)}, \hat{\mathbf{V}}^{(3)}, \hat{\mathbf{V}}^{(4)}\}$ denote the multi-scale visual

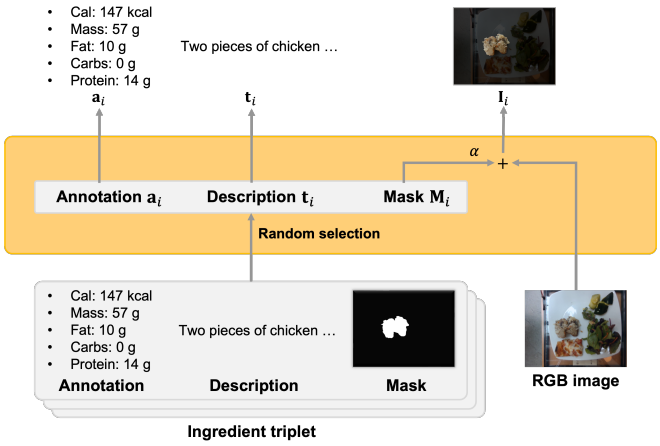


Fig. 4. **Illustration of mask-based ingredient-level augmentation.**

features produced by the LGFE module. For each scale $\hat{\mathbf{V}}^{(i)}$, we apply global average pooling (GAP) to obtain a feature vector: $\mathbf{v}^{(i)} = \mathrm{GAP}(\hat{\mathbf{V}}^{(i)}) \in \mathbb{R}^d$, where $d = 256$. The pooled features from all scales are concatenated to form a global visual representation: $\mathbf{v}_{\text{global}} = \left[\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \mathbf{v}^{(3)}, \mathbf{v}^{(4)}\right]$, where $[,]$ denotes concatenation over the feature dimension.

This global representation is fed into a two-layer shared MLP with ReLU activation to produce a common latent embedding: $\mathbf{h} = \mathrm{MLP}_{\text{shared}}(\mathbf{v}_{\text{global}})$. The shared embedding $\mathbf{h}$ is then forwarded to five independent regression heads, each corresponding to a specific nutrition value. Each head is implemented as a two-layer MLP: $\hat{y}_k = \mathrm{MLP}_k(\mathbf{h})$, where $k \in \{\text{calories}, \text{mass}, \text{fat}, \text{carbohydrates}, \text{protein}\}$.

The dynamic range and scale of different nutrition values vary significantly. Calorie and mass values are typically much larger in magnitude than fat, carbohydrate, and protein. To avoid optimization bias, we leverage the Percentage of Mean Absolute Error (PMAE) as the loss function. By comparing the predicted nutritional values and the ground truths, the Mean Absolute Error (MAE) is calculated. The PMAE of the kth nutrition is then obtained by:

$$\mathcal{L}_k = \mathrm{PMAE}_k = \frac{\mathrm{MAE}_k}{\frac{1}{M}\sum_{i=1}^{M} y_k^{(i)}}, \tag{6}$$

where $\frac{1}{M}\sum_{i=1}^{M} y_k^{(i)}$ is the mean ground truth value of the kth nutrition attribute across M samples.

The total loss is computed as the sum over all nutrients:

$$\mathcal{L}_{\text{total}} = \Sigma_k \mathcal{L}_k, \tag{7}$$

where $k \in \{\text{calorie}, \text{mass}, \text{fat}, \text{carbs}, \text{protein}\}$

III. EVALUATION

A. Implementation Details

We employ the Swin-B model [10] to extract features from RGB and depth images. For both training and evaluation, text descriptions for each image are generated by GPT-4o once and stored in advance. To encode text descriptions, we use the BERT-base model [15] as the text encoder. Text inputs are tokenized using BERT's byte-pair encoding (BPE) scheme [16].

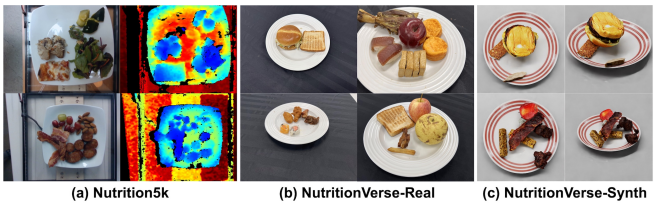


Fig. 5. Sample images from (a) Nutrition5k, (b) NutritionVerse-Real, and (c) NutritionVerse-Synth.

To learn network parameters, we adopt the AdamW optimizer. The learning rate is set to 1×10^{-5} for the encoders and to 1×10^{-4} for all other components. Weight decay is fixed at 1×10^{-4}. The learning rate scheduler is an exponential decay strategy with a gamma value of 0.99. During training, we apply a series of data augmentations, including the proposed MIA, and conventional augmentations like random horizontal and vertical flipping, sharpness and contrast adjustments, and random resizing. Training is conducted on two NVIDIA RTX 4090 GPUs with a batch size of 10 per GPU. The model is trained for 200 epochs on the Nutrition5k [1] dataset, 150 epochs on NutritionVerse-Real [17], and 30 epochs on NutritionVerse-Synth [17].

B. Datasets

The Nutrition5k dataset [1] consists of 3.5K RGB images and their corresponding depth images. Each image and each ingredient within it are annotated with five nutrition values: calories, mass, fat, carbohydrates, and protein. We follow the training and testing splits defined in [1] for fair comparison. The NutritionVerse-Real dataset [17] contains 889 RGB images representing 251 unique dishes, each annotated with the same five nutrition values and ingredient-level nutrition labels. The NutritionVerse-Synth [17] dataset further scales the dataset to 84,984 RGB-D images across 7,082 unique dishes. The images are captured from multiple viewpoints and include both global and ingredient-level annotations for all five nutritional values.

Figure 5 shows several samples of these three datasets.

C. Quantitative Results

Table I shows performance comparison in terms of PMAE. In the RGB-D setting, KARINA sets a new state of the art with a mean PMAE of 13.3%. Notably, KARINA outperforms ingredient-based baselines IMIR-Net [9] and IG RGB-D Net [8] that rely solely on raw ingredient names and coarse-grained fusion strategies. Moreover, KARINA surpasses RGB-D fusion frameworks such as CMX [18] and CDINet [20].

In the RGB-only setting, KARINA also achieves state-of-the-art performance, with a mean PMAE of 14.4%. KARINA also surpasses recent LMM-based approaches FoodLMM [27] and RoDE [28], which directly fine-tune LMMs for nutrition-related tasks. These results highlight the effectiveness and efficiency of our strategy.

TABLE I

PERFORMANCE COMPARISON IN TERMS OF PMAE ON THE NUTRITION5K DATASET.

Input	Methods	Cal ↓	Mass ↓	Fat ↓	Carbs ↓	Protein ↓	Mean ↓
RGB-D	CMX [18]	21.8	20.7	34.8	37.0	33.2	29.5
	HINet [19]	24.5	25.2	43.4	39.9	38.8	34.3
	CDINet [20]	21.1	20.4	37.1	37.1	32.8	29.7
	DEFNet [21]	32.7	34.2	48.9	40.3	43.8	39.9
	TriTransNet [22]	22.1	20.1	37.5	34.8	38.0	30.5
	Deliver [23]	29.5	25.9	48.3	47.7	46.1	39.5
	Google-Nutrition-depth [1]	18.8	18.9	18.1	23.8	20.9	20.1
	Domain Adaptation-Nutrition [24]	16.8	–	–	–	–	–
	RGB-D Net [5]	15.0	10.8	23.5	22.4	21.0	18.5
	ADFE [7]	14.4	11.7	22.5	21.0	19.4	17.8
	OVS-Nutrition [25]	13.9	8.8	21.4	22.1	19.6	17.2
	IMIR-Net [9]	14.5	10.4	21.8	20.4	20.0	17.4
	IG RGB-D Net [8]	13.7	9.8	19.2	19.3	17.6	15.9
	NuNet [6]	12.8	8.7	19.6	18.6	18.4	15.6
	KARINA	**11.4**	**8.6**	**17.1**	**15.1**	**14.1**	**13.3**
RGB	Google-Nutrition-monocular [1]	26.1	18.8	34.2	31.9	29.5	29.1
	Coarse-to-Fine Nutrition [3]	24.1	19.4	36.0	32.1	33.5	29.0
	Swin-Nutrition [2]	16.2	13.7	24.9	21.8	25.4	20.4
	Portion-Nutrition [26]	15.8	–	–	–	–	–
	DPF-Nutrition [4]	14.7	10.6	22.6	20.7	20.2	17.8
	FoodLMM [27]	53.6	39.3	67.1	49.5	52.9	52.5
	RoDE [28]	52.4	38.4	67.1	47.8	53.9	51.9
	KARINA	**12.1**	**10.2**	**18.6**	**16.2**	**14.7**	**14.4**

TABLE II

PERFORMANCE COMPARISON BETWEEN KARINA AND GPT-4O IN TERMS OF PMAE ON THE NUTRITION5K DATASET.

Input	Methods	Cal ↓	Mass ↓	Fat ↓	Carbs ↓	Protein ↓	Mean ↓
RGB	GPT-4o	30.4	31.5	44.4	40.9	41.9	37.8
	KARINA	**12.1**	**10.2**	**18.6**	**16.2**	**14.7**	**14.4**

TABLE III

PERFORMANCE COMPARISON IN TERMS OF MAE ON THE NUTRITIONVERSE DATASET.

Dataset	Input	Methods	Cal ↓	Mass ↓	Fat ↓	Carbs ↓	Protein ↓	Mean ↓
Real	RGB	NV-Real baseline [17]	296.9	115.9	16.2	29.8	62.6	104.3
	RGB	KARINA	**107.4**	**65.6**	**7.0**	**11.1**	**12.6**	**40.7**
Synth	RGB-D	NV-Synth baseline [17]	202.0	78.8	30.1	33.3	23.5	73.5
	RGB-D	KARINA	**36.2**	**18.4**	**1.7**	**3.0**	**3.6**	**12.6**

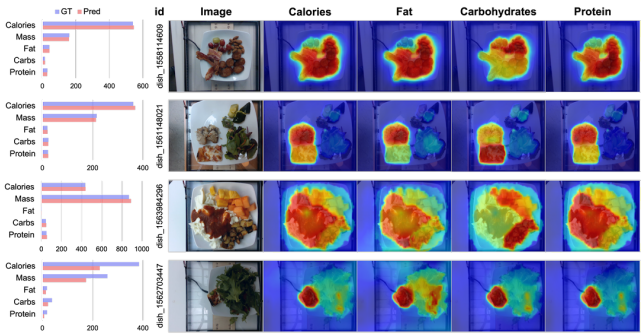


Fig. 6. Visual samples of nutrition estimation and heatmaps.

We further try to prompt GPT-4o to directly do nutrition estimation on the Nutrition5k test set. As shown in Table II, KARINA outperforms GPT-4o across all nutrition attributes.

Table III shows the comparisons on the NutritionVerse-Real and NutritionVerse-Synth datasets, using the Mean Absolute Error (MAE) as the evaluation metric. On the NutritionVerse-Real dataset, KARINA achieves a mean MAE of 40.7, substantially outperforming the NV-Real baseline. Across all nutrition attributes, KARINA consistently delivers lower errors. On the NutritionVerse-Synth dataset, KARINA achieves a mean MAE of 12.6, while the NV-Synth baseline reports 73.5. These results show that our method generalizes well to several nutrition estimation benchmarks.

D. Qualitative Results

We show several estimation examples on the Nutrition5k dataset in Figure 6. For each dish, we present bar charts comparing predicted nutrition values with ground truth, and use Grad-CAM [29] to highlight the model's attention for calories, fat, carbohydrates, and protein, respectively. Grad-CAM is applied to the final layer of the language-guided feature enhancement block in KARINA.

In the first row, the model correctly attends to bacon, sausage, and almonds in the calories, fat, and protein maps, as these items contribute significantly to those nutrients. For carbohydrates, attention is directed to grapes and almonds, which serve as the primary carbohydrate sources. In contrast, the final row shows a failure case where KARINA underestimates the nutrition values due to a key ingredient being partially occluded. Closer inspection reveals that a piece of beef is partially occluded by surrounding ingredients, suggesting that such visual occlusions remain a limitation in the dataset and present a challenge to accurate prediction.

Notice that vegetable regions contribute less to these nutrients and are less highlighted across all heatmaps.

E. Ablation Studies

Table IV presents the impact of different components on the Nutrition5k dataset. Using only the visual encoder (VE) with RGB input and MLP prediction heads, the mean PMAE is 16.7. Adding the visual feature fusion (VFF) module with RGB-D inputs improves performance to 16.0, confirming the benefit of multi-modal fusion. Introducing the LGFE module further reduces the mean PMAE to 14.1. As shown in Sec. II-B, the text descriptions like "high calorie" and "medium protein" give important hints to effectively enrich visual features. Finally, incorporating mask-based ingredient-level augmentation (MIA) lowers the error to 13.3, showing

TABLE IV

PERFORMANCE VARIATIONS OF DIFFERENT COMPONENTS, IN TERMS OF PMAE ON THE NUTRITION5K DATASET.

Inputs	VE	VFF	LGFE	MIA	Cal ↓	Mass ↓	Fat ↓	Carbs ↓	Protein ↓	Mean ↓
RGB	✓				13.8	10.7	22.2	18.5	18.1	16.7
RGB-D	✓	✓			12.7	9.0	21.5	18.8	18.0	16.0
RGB-D	✓	✓	✓		11.6	9.0	18.7	16.3	14.7	14.1
RGB-D	✓	✓	✓	✓	**11.4**	**8.6**	**17.1**	**15.1**	**14.1**	**13.3**

the value of fine-grained alignment.

IV. CONCLUSION

We have presented a nutrition estimation method that leverages nutrition knowledge from an LMM to guide cross-modal fusion. We integrate RGB with depth information and enhance visual representations via language-guided enhancement. To further improve ingredient-level alignment, we introduce a mask-based ingredient-level augmentation during training. Experimental results on the three different datasets demonstrate that our approach learns more effective visual representations and achieves state-of-the-art performance across all metrics. In the future, we plan to extend our framework to leverage multiple views and associated textual descriptions of the same dish.

ACKNOWLEDGMENT

This work was funded in part the National Science and Technology Council, Taiwan, under grants 115-2425-H-006-005, 114-2622-E-006-028, 114-2221-E-006-047-MY3, 114-2425-H-006-004, 114-2637-8-218-002, 114-2634-F-006-002, and 112-2221-E-006-136-MY3.

REFERENCES

[1] Q. Thames, A. Karpur, W. Norris, F. Xia, L. Panait, T. Weyand, and J. Sim, "Nutrition5k: Towards Automatic Nutritional Understanding of Generic Food," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[2] W. Shao, S. Hou, W. Jia, and Y. Zheng, "Rapid Non-Destructive Analysis of Food Nutrient Content Using Swin-Nutrition," *Foods*, vol. 11, no. 21, p. 3429, 2022.

[3] B. Wang, T. Bu, Z. Hu, L. Yang, Y. Zhao, and X. Li, "Coarse-to-Fine Nutrition Prediction," *IEEE Transactions on Multimedia*, vol. 26, pp. 3651–3662, 2024.

[4] Y. Han, Q. Cheng, W. Wu, and Z. Huang, "DPF-Nutrition: Food Nutrition Estimation via Depth Prediction and Fusion," *Foods*, vol. 12, no. 23, p. 4293, 2023.

[5] W. Shao, W. Min, S. Hou, M. Luo, T. Li, Y. Zheng, and S. Jiang, "Vision-based food nutrition estimation via rgb-d fusion network," *Food Chemistry*, vol. 424, p. 136309, 2023.

[6] Z. Kwan, W. Zhang, Z. Wang, A. B. Ng, and S. See, "Nutrition Estimation for Dietary Management: A Transformer Approach with Depth Sensing," *IEEE Transactions on Multimedia*, pp. 1–13, 2025.

[7] D. Zhang, B. Ma, and X.-J. Wu, "Adaptive Feature Fusion and Enhancement Network for Food Nutrition Estimation," *IEEE Transactions on AgriFood Electronics*, pp. 1–11, 2025.

[8] Z. Feng, H. Xiong, W. Min, S. Hou, H. Duan, Z. Liu, and S. Jiang, "Ingredient-Guided RGB-D Fusion Network for Nutritional Assessment," *IEEE Transactions on AgriFood Electronics*, vol. 3, no. 1, pp. 156–166, 2025.

[9] F. Nian, Y. Hu, Y. Gu, Z. Wu, S. Yang, and J. Shu, "Ingredient-guided multi-modal interaction and refinement network for RGB-D food nutrition assessment," *Digital Signal Processing*, vol. 153, p. 104664, 2024.

[10] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[11] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in *Proceedings of the European Conference on Computer Vision*, 2018.

[12] Y. Wu and K. He, "Group Normalization," in *Proceedings of the European Conference on Computer Vision*, 2018.

[13] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection," in *Proceedings of the International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=gZ9hCDWe6ke

[14] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks," *arXiv preprint arXiv:2401.14159*, 2024.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[16] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2016.

[17] C.-e. A. Tai, M. Keller, S. Nair, Y. Chen, Y. Wu, O. Markham, K. Parmar, P. Xi, H. Keller, S. Kirkpatrick, and A. Wong, "NutritionVerse: Empirical Study of Various Dietary Intake Estimation Approaches," in *Proceedings of the International Workshop on Multimedia Assisted Dietary Management*, 2023.

[18] J. Zhang, H. Liu, K. Yang, X. Hu, R. Liu, and R. Stiefelhagen, "CMX: Cross-Modal Fusion for RGB-X Semantic Segmentation With Transformers," *IEEE Transactions on intelligent transportation systems*, vol. 24, no. 12, pp. 14 679–14 694, 2023.

[19] H. Bi, R. Wu, Z. Liu, H. Zhu, C. Zhang, and T.-Z. Xiang, "Cross-modal hierarchical interaction network for RGB-D salient object detection," *Pattern Recognition*, vol. 136, p. 109194, 2023.

[20] C. Zhang, R. Cong, Q. Lin, L. Ma, F. Li, Y. Zhao, and S. Kwong, "Cross-modality Discrepant Interaction Network for RGB-D Salient Object Detection," in *Proceedings of the ACM International Conference on Multimedia*, 2021.

[21] W. Zhou, Y. Pan, J. Lei, L. Ye, and L. Yu, "DEFNet: Dual-Branch Enhanced Feature Fusion Network for RGB-T Crowd Counting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 540–24 549, 2022.

[22] Z. Liu, Y. Wang, Z. Tu, Y. Xiao, and B. Tang, "TriTransNet: RGB-D Salient Object Detection with a Triplet Transformer Embedding Network," in *Proceedings of the ACM International Conference on Multimedia*, 2021.

[23] J. Zhang, R. Liu, H. Shi, K. Yang, S. Reiß, K. Peng, H. Fu, K. Wang, and R. Stiefelhagen, "Delivering Arbitrary-Modal Semantic Segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[24] G. Vinod, Z. Shao, and F. Zhu, "Image Based Food Energy Estimation With Depth Domain Adaptation," in *Proceedings of the International Conference on Multimedia Information Processing and Retrieval*, 2022.

[25] S. Parinayok, Y. Yamakata, and K. Aizawa, "Open-Vocabulary Segmentation Approach for Transformer-Based Food Nutrient Estimation," in *Proceedings of the ACM International Conference on Multimedia*, 2023.

[26] Z. Shao, G. Vinod, J. He, and F. Zhu, "An End-to-End Food Portion Estimation Framework Based on Shape Reconstruction from Monocular Image," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2023.

[27] Y. Yin, H. Qi, B. Zhu, J. Chen, Y.-G. Jiang, and C.-W. Ngo, "FoodLMM: A Versatile Food Assistant using Large Multi-modal Model," *arXiv preprint arXiv:2312.14991*, 2023.

[28] P. Jiao, X. Wu, B. Zhu, J. Chen, C.-W. Ngo, and Y. Jiang, "RoDE: Linear Rectified Mixture of Diverse Experts for Food Large Multi-Modal Models," *arXiv preprint arXiv:2407.12730*, 2024.

[29] 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/CVF International Conference on Computer Vision*, 2017.