RESEARCH-ARTICLE

Neural Additive Adapters for Interpretable Nutrition Prediction

VITALII EMELIANOV, University of Udine, Udine, UD, Italy

NIKI MARTINEL, University of Udine, Udine, UD, Italy

Published: 27 October 2025

Citation in BibTeX format

MM '25: The 33rd ACM International Conference on Multimedia
October 27 - 31, 2025
Dublin, Ireland

Conference Sponsors:
SIGMM

https://doi.org/10.1145/3746027.3755865
ISBN: 9798400720352

Neural Additive Adapters for Interpretable Nutrition Prediction

Vitalii Emelianov
vitalii.emelianov@uniud.it
Department of Mathematics, Computer Science and Physics, University of Udine
Udine, Italy

Niki Martinel
niki.martinel@uniud.it
Department of Mathematics, Computer Science and Physics, University of Udine
Udine, Italy

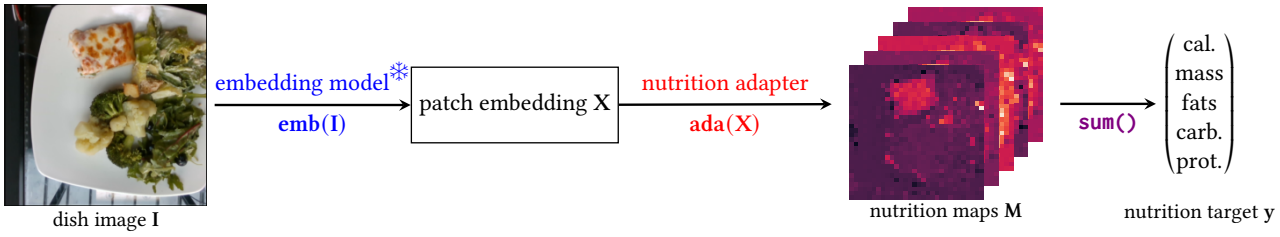


Figure 1: Illustration of the framework. The embedding model emb is frozen. The nutrition adapter ada is learnable.

Abstract

We study how large vision models (LVMs) can predict food nutrition through lightweight and interpretable adapters—the machine learning modules the predictions of which could be understood by humans. We introduce novel nutrition adapters that use features extracted by pre-trained LVMs and output the so-called nutrition maps. Nutrition maps indicate the concentration of nutrition values per each image location. We use such an interpretable representation to obtain the nutrition targets as a sum of all nutrition concentrations on the maps. To understand our approach's generalization capability, we systematically analyze the behavior of our novel interpretable adapters leveraging different LVMs with different food image-nutrition datasets. Our lightweight approach delivers better or on-par performance than the state-of-the-art models on the Nutrition5k and the Nutritionverse-Real benchmarks. The code is provided at https://github.com/vitaly-emelianov/nutrition-adapters.

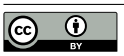
CCS Concepts

• **Computing methodologies** → **Computer vision**.

Keywords

nutrition prediction; foundation models; computer vision

ACM Reference Format:
Vitalii Emelianov and Niki Martinel. 2025. Neural Additive Adapters for Interpretable Nutrition Prediction. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3746027.3755865



MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755865

1 Introduction

Food nutrition logging is a fundamental step for controlling caloric and nutrition intake. Precisely estimating nutrition factors would significantly impact our society by guiding patients towards using dietary suggestions and promoting healthy habits. People avoid such a cumbersome activity since it requires accurate and time-consuming manual tasks.

To ease such a task, our community has recently introduced machine learning-based tools automating nutrition estimation from food images. Existing deep learning-based approaches [11, 16, 25, 28, 32] have achieved remarkable performance on current benchmarks, but they require intensive training on large-scale datasets. The outputs of these models are also usually hard to interpret: Why is the nutrition prediction made this way? The lack of interpretability leads to a lack of trust in such black-box approaches [22].

Large vision models (LVMs)—machine learning models trained to tackle common vision problems with massive amounts of image data from the Internet—achieve state-of-the-art performance on different computer vision tasks such as classification [20, 21] or semantic segmentation [18] without requiring fine-tuning. However, they cannot be directly applied to nutrition prediction—which is a regression problem based on food images—without an adaptation.

This motivates us to introduce *a novel approach for nutrition prediction via lightweight and interpretable LVM adapters*. We take inspiration from the generalized or neural additive model approaches [1, 12] proposing LVMs adapters to generate nutrition maps (Figure 1). Nutrition maps show the concentration of nutrition targets per image locations (*i.e.*, patches), which makes them interpretable representations for nutrition prediction. Nutrition targets are estimated as sums of nutrition maps over all patches.

The contribution of our work can be summarized as follows:

- We introduce lightweight adapters that allow us to estimate the concentration of nutrition targets per image locations and to regress to nutrition targets with a single interpretable module.

- We extensively study how different general-purpose LVMs, such as CLIP [21], MAE [13], and DINOv2 [20], perform with our adapters on the Nutrition5k [28] and the Nutritionverse-Real [8] image-nutrition datasets.
- We demonstrate that our adapters showcase interpretability features while yielding to novel state-of-the-art results on the Nutritionverse-Real dataset and on-par performance on the Nutrition5k dataset.

2 Related Work

2.1 Nutrition Prediction Using Food Images

While there are many food datasets available [3, 4, 8, 9, 28], only a fraction of them provide the nutrition information along with the food images. The most common benchmark dataset for nutrition prediction is the Nutrition5k dataset [28]. The dataset consists of 5006 images of different dishes composed of different food categories taken in a canteen environment. The images are accompanied with the nutrition information, such as the mass of the dish, its energy, and the masses of macro-nutrients such as fats, carbohydrates, and proteins. In addition to the Nutrition5k dataset [28], smaller datasets for nutrition prediction in more realistic environments are available [3, 8].

In addition to the Nutrition5k dataset, in [28], the authors propose a model for nutrition estimation. The model is based on an Inception-V2 backbone model [26] edited to include a few fully-connected layers with ReLU non-linearity regressing the nutrition estimates. The authors train the model with both RGB and RGB-D images. Following [28], multiple approaches improving upon their baseline model have been proposed [11, 16, 25, 27]. Such approaches to nutrition prediction use different modalities of food data. Some models (*e.g.*, [16]) are trained solely using the RGB images, while others use additional modalities, such as RGB-D images (*e.g.*, [11, 25]) or segmentation masks (*e.g.*, [32]). Overall, proposed state-of-the-art models either do not provide any explanation of the prediction or only a post-hoc explanation is provided. For example, [11, 25] measure the attention to different image parts using the Grad-CAM method [24]. However, with such a method, it is not clear how the image patches contribute to the prediction whereas in our approach the contribution is additive by design.

In our work, we propose a nutrition prediction model outputting the nutrition maps that indicate the concentration of nutrition target per image location. Similar to [11, 16, 28, 32], we use pre-trained models to first extract features from food images. In contrast to the current approaches to nutrition prediction, we keep the weights of the vision backbones frozen, and we use the embeddings of image patches to predict the nutrition maps using lightweight learnable adapters. This makes our model interpretable since a nutrition prediction can be explained using the corresponding nutrition map.

2.2 Interpretable Machine Learning

Interpretable machine learning methods exploit methods—such as decision trees or linear models—that are self-explanatory by design [22]. A general approach to interpretable machine learning can be summarized using the notion of generalized additive models [12]. The generalized additive model represents the prediction model as a sum of non-linear functions where each function depends only on a single feature.

In [1], the authors propose to represent the feature-dependent functions as feed-forward neural networks of a fixed architecture (*e.g.*, a multi-layer perceptron). This neural additive model approach [1] enables evaluation and visualization of the contribution of each feature to the target. In [31], the authors show the limitations of additive models from [1, 12] by demonstrating a bound on the performance of the additive models when the features are highly correlated. To mitigate this limitation of [1], [31] proposes to group features into bundles using a trainable module.

Another common approach to interpretable machine learning in image classification is based on showing the comparison of the image during the inference with the so-called "prototypes" images [6, 19, 23, 29] that are typical representatives of the classes. The explanations indicate which patches of the classified image are close to the prototype image, justifying the predicted class labels.

While there is a substantial work on interpretable image classification models, such approaches for image regression problems, to our knowledge, are not common [14, 15] and are applied to one-dimensional regression. In the nutrition prediction problem, we need to provide several nutrition target estimates. In this paper, we propose a different approach to prototype-based models. We base our adapter on an additive model similar in spirit to neural additive models [1] that were typically used for tabular data. In our adapters, the image patches are considered as individual features.

3 Problem Setup

An image $\mathbf{I} \in \mathcal{I} \subseteq \mathbb{R}^{H \times W \times C}$ is represented as a grid of N, $P \times P$ non-overlapping patches where the patch size $P \geq 1$. We assume to have access to an embedding model $\mathbf{emb} : \mathcal{I} \to \mathcal{X}^N$ that maps image patches to D-dimensional vectors $\mathcal{X} \subseteq \mathbb{R}^D$. Hence, an image $\mathbf{I}$ is mapped to a matrix $\mathbf{X} = (\mathbf{x}_1 \dots \mathbf{x}_N)^\intercal$ where $\mathbf{x}_p \in \mathcal{X}$ denotes the patch representation with index p. Such image representations are commonly used in vision transformers [5, 7, 13, 20].

The nutrition targets are generally a subset of the following quantities: the mass of the dish, the energy, and the masses of fats, carbohydrates, and proteins in the dish. They are represented as $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^K$. We denote by $\mathbf{ada} : \mathcal{X}^N \to \mathcal{Y}$ the adapter model that maps the patch representation of an image to nutrition targets. Therefore, we consider the class of nutrition prediction models $\mathbf{h}$ which can be represented using compositions $\mathbf{h} = \mathbf{ada} \circ \mathbf{emb}$. Our framework is illustrated in Figure 1. The notation is summarized in Table 1.

Table 1: Notation.

$H \times W \times C$	image size (height, width, number of channels)
$P \times P$	patch size
N	number of patches of an image $\mathbf{I}$
D	embedding dimension of patch representation $\mathbf{X}$
K	nutrition target dimension
$\mathbf{emb} : \mathcal{I} \to \mathcal{X}^N$	embedding model
$\mathbf{ada} : \mathcal{X}^N \to \mathcal{Y}$	adapter model
$\mathbf{h} = \mathbf{ada} \circ \mathbf{emb}$	nutrition prediction model

Following [11, 27, 28], we want to minimize the mean absolute error (MAE) of a model $\mathbf{h}$ on a dataset $\mathcal{D} = \{\mathbf{I}_i, \mathbf{y}_i\}_{i=1}^n$:

$$\text{MAE}(\mathbf{h}, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{h}(\mathbf{I}_i) - \mathbf{y}_i\|_1 . \tag{1}$$

4 Interpretable Additive Nutrition Adapters

We propose interpretable adapters for nutrition prediction that assign nutrition values to image patches. We represent the adapter **ada** as an additive function over the image patch representations:

$$\mathbf{ada}(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \boldsymbol{\beta} + \sum_{p=1}^{N} \mathbf{ada}_p(\mathbf{x}_p). \tag{2}$$

This model is a special case of the neural additive model [1] with identity link functions where image patches are considered as different image features. For each patch $\mathbf{x}_p$, the function $\mathbf{ada}_p : \mathcal{X} \to \mathcal{Y}$ assigns the nutrition target values $\mathbf{y}$. The quantity $\boldsymbol{\beta} \in \mathbb{R}^K$ is a bias term. Hence, the matrix $\mathbf{M} = (\mathbf{ada}_1(\mathbf{x}_1) \ldots \mathbf{ada}_N(\mathbf{x}_N))$ can be interpreted as a nutrition map, and it can serve as a visual explanation of the nutrition target prediction. We will illustrate such maps in Section 6.3.

As a more general case, we also consider:

$$\mathbf{ada}(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \boldsymbol{\beta} + \sum_{p=1}^{N} \mathbf{ada}_p(\mathbf{x}_1, \ldots, \mathbf{x}_N), \tag{3}$$

where the prediction of nutrition map for the patch p is a function of all other patches, *i.e.*, $\mathbf{ada}_p : \mathcal{X}^N \to \mathcal{Y}$.

4.1 Special Cases of Nutrition Adapters

We consider four families of functions $\mathbf{ada}_p$. The first three families represent functions in the form (2). The last has a form (3). We introduce different adapters ranging from simple to quite expressive modules. The objective is to find a trade-off between the model complexity and the prediction error.

Linear Adapter. We learn linear projections $\mathbf{W}_p \in \mathbb{R}^{D \times K}$ from the patch embedding space $\mathcal{X}$ to the nutrition target space $\mathcal{Y}$:

$$\mathbf{ada}_p^{\text{lin}}(\mathbf{x}_p) = \mathbf{x}_p^\intercal \mathbf{W}_p. \tag{4}$$

The number of learnable parameters per patch is $D \cdot K$.

Multi-Layer Perceptron Adapter (MLP). We learn a two-layer perceptron with a single hidden layer and a non-linearity σ:

$$\mathbf{ada}_p^{\text{mlp}}(\mathbf{x}_p) = \sigma\left(\sigma\left(\mathbf{x}_p^\intercal \mathbf{W}_p^{\text{in}}\right) \mathbf{W}_p^{\text{hid}}\right) \mathbf{W}_p^{\text{out}}. \tag{5}$$

The input dimension is equal to the dimension of the patch embedding D. The output dimension is equal to K. The hidden dimension equals to $\lceil \alpha D \rceil$ where $\alpha > 0$ is the so-called MLP-factor, and $\lceil \cdot \rceil$ is the ceiling function. The number of learnable parameters per patch is $\lceil \alpha D \rceil D + \lceil \alpha D \rceil^2 + \lceil \alpha D \rceil K$.

Bilinear Adapter. We represent the function $\mathbf{ada}_p$ as an element-wise product of two functions, that is:

$$\mathbf{ada}_p^{\text{bilin}}(\mathbf{x}_p) = \mathbf{ada}_p^{(1)}(\mathbf{x}_p) \odot \mathbf{ada}_p^{(2)}(\mathbf{x}_p), \tag{6}$$

where $\mathbf{ada}_p^{(1)}, \mathbf{ada}_p^{(2)}$ are the MLP adapters (5) and $\odot$ denotes the element-wise product between two vectors. The bilinear model involves training a pair of models which are then fused using an element-wise product. The first model can be interpreted as learning a "food mask" assigning higher values to "food" patches (*i.e.*, patches that contain food) and lower values to the "non-food" patches (*i.e.*, patches that correspond to plate or table). The number of parameters is equal to the number of parameters of the two MLP adapters.

Finally, we consider an attention-based adapter model that has the form (3).

Attention Adapter. We first transform the patch embeddings using the scaled dot-product self-attention mechanism [30] and layer normalization [2]. Then, we apply the MLP adapter (5) to the transformed patch embeddings:

$$\tilde{\mathbf{x}}_1 \ldots \tilde{\mathbf{x}}_N = \text{layernorm}\left[\text{softmax}\left(\frac{(\mathbf{X}\mathbf{W}^q)(\mathbf{X}\mathbf{W}^k)^\intercal}{\sqrt{D}}\right)\mathbf{X}W^v\right],$$
$$\mathbf{ada}_p^{\text{att}}(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \mathbf{ada}_p^{\text{mlp}}(\tilde{\mathbf{x}}_p), \tag{7}$$

where $\mathbf{X} = (\mathbf{x}_1 \ldots \mathbf{x}_N)^\intercal$ is the combined feature representation of all patches, $\mathbf{W}^k$, $\mathbf{W}^q$, $\mathbf{W}^v \in \mathbb{R}^{D \times D}$ are the key, query, and value matrices. This replicates the standard attention block commonly used in transformer models. The number of parameters of the attention adapter is equal to the number of parameters of the MLP adapter plus the number of parameters in the self-attention equal to $3D^2$.

4.2 Discussion on the Adapter Models

The additive adapter model (2) resembles the neural additive model from [1] with identity link function. This model is natural to our problem since it allows to observe how each image patch contributes to the nutrition targets.

The models considered in (4), (5) and (6) assume that all functions $\mathbf{ada}_p$ are independently parameterized. This might be beneficial to capture the complexity of nutrition maps but can also lead to over-fitting. Hence, in this paper, we consider the case when a joint model $\mathbf{f}$ is trained, that is $\mathbf{ada}_p = \mathbf{f}$ for all $p \in \{1, \ldots, N\}$.

5 Experimental Setting

We verify the applicability of our adapter models for the nutrition prediction using different embedding models (*i.e.*, vision backbones) and on different nutrition datasets.

5.1 Vision Backbones

We consider three vision foundation backbone models. All the considered backbones are the vision transformer models [7], but they were trained using different approaches which leads to different capabilities for computer vision tasks as shown in [10]. We study the performance of each approach for the nutrition prediction.

CLIP. Contrastive language-image pre-training approach [21] trains both an image encoder as well as the text encoder. Both image and text encoders are transformer models. It uses pairs of image and its textual description and is trained using a contrastive loss which enforces that image-text encoding that correspond to the same pair remain closer than image-text pairs that are not a true pair. The authors use a large collected dataset of image-text queries

of size 400 million obtained from various public resources on the Internet.

MAE. Masked auto-encoder [13] is a vision transformer encoder-decoder model. It is trained in a self-supervised manner where random patches of images are masked and the task for the decoder is to reconstruct from the embeddings of masked and unmasked patches the original image in the pixel space by minimizing the mean squared error.

DINO and DINOv2. Self-distillation with no labels [5] approach, trains a vision transformer model in a self-supervised manner. An image is split into a few global views (*i.e.*, crops of a high resolution), and several local views (*i.e.*, crops of a lower resolution). The framework consists in training a teacher and a student model of the same architecture. Given a fixed teacher model, the cross-entropy loss between the representation of a student and a teacher models is minimized. Given a fixed student model, the teacher model is updated using the moving average of a student and a teacher model. The DINOv2 model [20] proposes an improved version of the DINO model which uses Sinkhorn-Knopp centering, patch masking in the teacher model training, and the Koleo regularizer.

5.2 Datasets

We evaluate the generalization capability of the proposed approach on two publicly available food image datasets.

Nutrition5k. [28] consists of 5006 images with their nutrition information. The images were taken in a canteen from the top-down views. The dishes are composed of 555 different food categories. The nutrition information is comprised of the mass of the dish, its energy, and masses of macro-nutrients such as fats, carbohydrates, and proteins. In addition to RGB images, the images from the depth camera (RGB-D) are also provided.

Nutritionverse-Real. [8] is a dataset of 251 dishes with 889 images taken from different angles with nutrition information. The number of unique food categories is equal to 45. This dataset has the same nutrition targets as the Nutrition5k dataset, but it is more realistic: food images are taken by users with their smartphone cameras.

5.3 Metrics

We use the three following metrics to evaluate the generalization capabilities of a model $\mathbf{h}$ on the validation dataset $\mathcal{D} = \{\mathbf{I}_i, \mathbf{y}_i\}_{i=1}^n$. First, we measure the component-wise mean absolute error (MAE) where $k \in \{1, \dots, K\}$:

$$\text{MAE}_k(\mathbf{h}, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} |\mathbf{h}(\mathbf{I}_i)[k] - \mathbf{y}_i[k]|,$$

Above, $\mathbf{y}_i[k]$ denotes the component k of a vector $\mathbf{y}_i[k]$.

Second, following [11, 28], we report the percentage of mean absolute error (PMAE) that is the mean absolute error rescaled by the mean nutrition target:

$$\text{PMAE}_k(\mathbf{h}, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\mathbf{h}(\mathbf{I}_i)[k] - \mathbf{y}_i[k]}{\bar{\mathbf{y}}[k]} \right| \cdot 100\%,$$

where $\bar{\mathbf{y}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{y}_i$ is the average nutrition target value in $\mathcal{D}$.

Finally, we report on the root mean square error (RMSE) measuring the quadratic divergence of the prediction from the ground truth:

$$\text{RMSE}_k(\mathbf{h}, \mathcal{D}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\mathbf{h}(\mathbf{I}_i)[k] - \mathbf{y}_i[k])^2}.$$

This is a standard metric in regression problems.

We also consider the average metrics over all nutrition targets for a better visualization, *i.e.*, we report $\mu(\mathbf{h}, \mathcal{D}) = \frac{1}{K} \sum_{k=1}^{K} \mu_k(\mathbf{h}, \mathcal{D})$ for all metrics $\mu \in \{\text{MAE}, \text{PMAE}, \text{RMSE}\}$.

5.4 Implementation Details

We use a center crop of an image of resolution 420×420 for all vision backbones except the CLIP [21]. For the CLIP backbone, we use the center crop of resolution 224×224. We used the patch size of 14 across all vision backbones. No image augmentation is performed.

We use the Adam optimizer [17] to train each adapter model while the backbone models are kept frozen. We use a learning rate 0.0001 and a batch size of 16 to train for 300 epochs. The randomness in all the experiments is controlled by fixing a random seed $\in \{0, 1, 2\}$. The train and test splits for the Nutrition5k dataset [28] and the Nutritionverse-Real dataset [8] were provided by their creators.

6 Experimental Results

6.1 Ablation Study

We study how different components of the framework such as vision backbones and nutrition adapters impact the performance.

In Figure 2, we illustrate the ablation results on the Nutrition5k dataset [28]. First, we observe that the features extracted using the DINOv2 backbone [20] allow reaching the lowest error across all metrics and among all adapters, while the performance with the CLIP backbone is lower and is similar to the performance with the MAE backbone. Second, we observe that a simple linear adapter does not allow reaching a small prediction error and that more complex adapters such as MLP adapter and bilinear adapter perform better in general.

In Figure 3, we illustrate the ablation results for the Nutritionverse-Real dataset [8]. Overall, similarly to our results for the Nutrition5k dataset, the DINOv2 backbone allows reaching the lowest error among all backbones for all adapters. In addition, we observe that the attention adapter reaches the lowest error across all adapters. This is in contrast with our results for the Nutrition5k dataset.

The better performance of the DINOv2 backbone across both datasets and nutrition adapters may be explained by the fact that the training to reconstruct high-resolution crops from low-resolution crops allows to better capture the food features such as texture. In addition, as shown in [20, Section 7.4], the extracted features allow solving dense prediction tasks such as semantic segmentation and depth prediction with minimal tuning. Such information may be important for food volume estimation and is therefore encoded in the features extracted by the DINOv2 model. The CLIP model uses text-image pairs for training and we hypothesize that the training dataset could not cover different food varieties and, in particular, dishes composed of multiple food categories. The masked

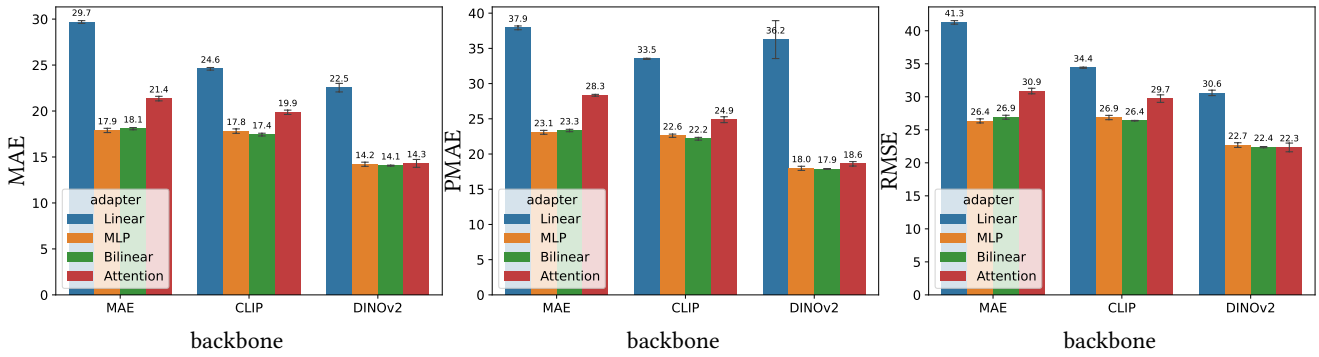


Figure 2: Ablation analysis of the different backbones and adapters on the Nutrition5k dataset [28]. The evaluation is shown for MAE, PMAE, and RMSE metrics.

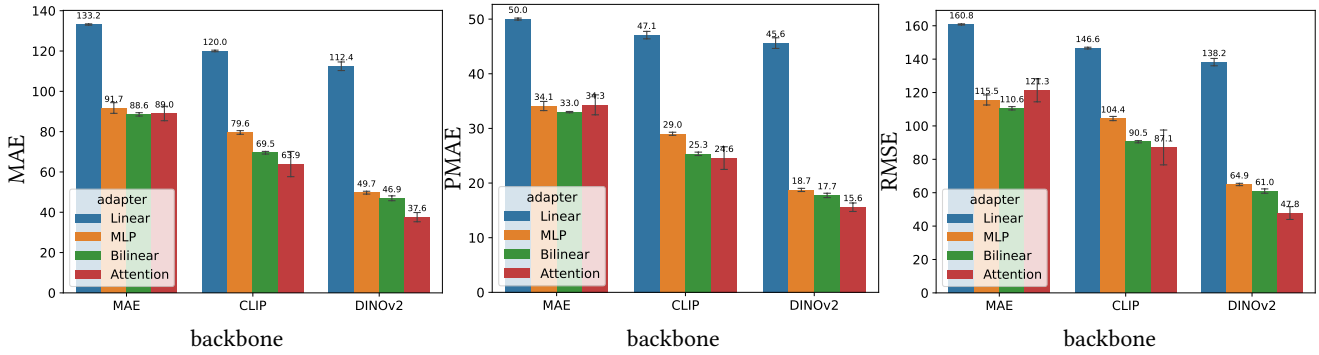


Figure 3: Ablation analysis of the different backbones and adapters on the Nutritionverse-Real dataset [8]. The evaluation is shown for MAE, PMAE, and RMSE metrics.

autoencoder model, similar, to the CLIP approach, might not be applicable to reconstructing well food items when using masking due to a high diversity of food image data.

6.2 Comparison With the State-of-the-Art

We compare the adapter-based approach with state-of-the-art models. We fix the DINOv2 [20] as the backbone model since it performed best across all metrics in our ablation analysis in Section 6.1.

In Figure 4, we provide a comparison for the Nutrition5k dataset on the MAE and PMAE metrics with the approach by [28] who use the ground-truth depth information (RGB-D) taken using the depth camera as an additional feature during both training and validation. In addition, we report the results of [11] who uses the depth information during the model training. The authors jointly train a nutrition prediction module as well as the RGB-D from RGB prediction module. Both RGB and RGB-D encodings are fused to predict the nutrition targets. The RGB-D information is used for the training but not for the validation of the model. We note that, in our approach, we do not use the RGB-D information during training nor during the validation yet we obtain a slightly better performance for proteins and carbohydrates prediction on the MAE metric when the MLP or the bilinear adapter is used.

In Figure 5, we compare the performance of our framework for the Nutritionverse-Real dataset [8], with the model of [27] who take an Inception-V2 model pre-trained on the Imagenet and the Nutrition5k dataset, and fine-tune it on the Nutritionverse-Real dataset. We observe that our framework allows reaching a lower error compared to [27] with the attention adapter used across for all macro-nutrients. Our model, in contrast to the approach by [27], does not require fine-tuning the backbone weights and it only trains a small adapter model.

6.3 Intepretability of Predictions

We visualize the nutrition maps $\mathbf{M} = (\mathbf{ada}_1, \ldots, \mathbf{ada}_N)$. For each dataset, we take the best combination of the backbone and model from Section 6.1 and we fix the random seed to 0. We plot the nutrition maps of five random images from the validation split. We note that we used both the CLS and image patch tokens for the nutrition target prediction. For visualization, we only use the image patch tokens rearranged in a square grid.

In Figure 6, we illustrate the nutrition maps for the Nutrition5k dataset. We observe that the nutrition maps highlight in some cases the parts of the image that correspond to a higher concentration of the nutrition target. For example, in the last row, we can see that the calorie map highlights the slice of pizza as having higher calories,

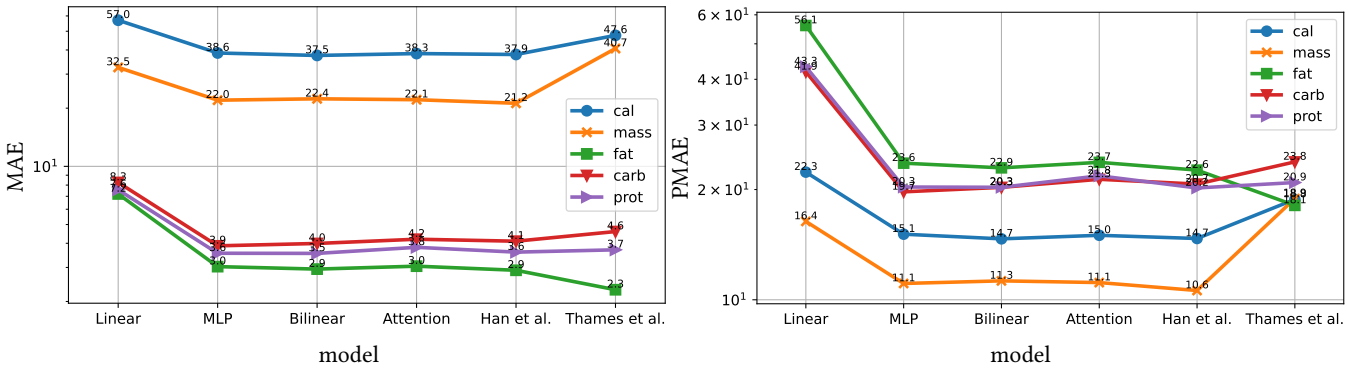


Figure 4: Comparison with state-of-the-art on the MAE and the PMAE metric for the Nutrition5k dataset [28]. The image patch representations are obtained using the DINOv2 backbone [20]. The comparison is performed with models from [28] and [11].

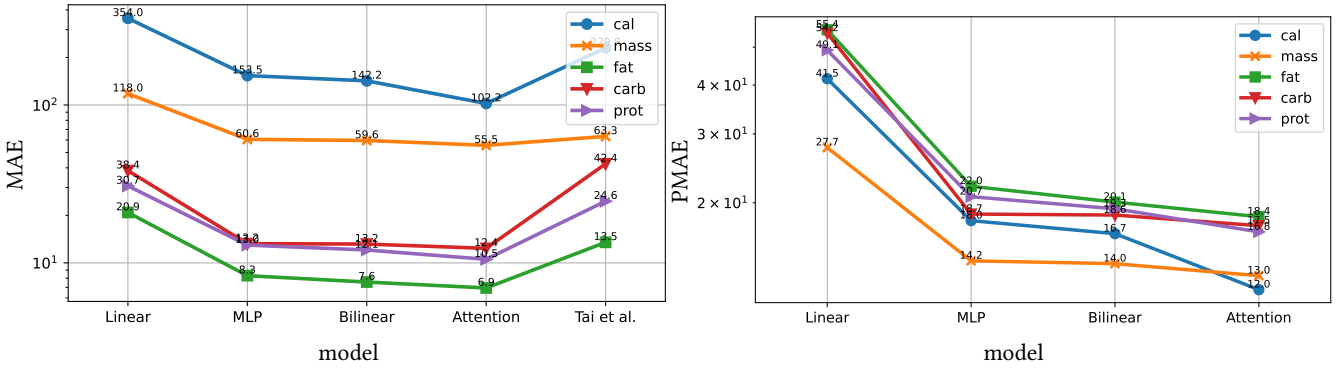


Figure 5: Comparison with state-of-the-art on the MAE and PMAE metric for the Nutritionverse-Real [8] dataset. The image patch representations are obtained using the DINOv2 backbone [20]. The comparison on the MAE metric (left panel) is performed with the model from [27]. The authors of [27] do not report the PMAE metric (right panel).

and, on the mass map, the broccoli pieces are highlighted as having more weight. On the contrary, in the second row, we observe that the map for fats is not meaningful since the background is assigned more fat values than the food item.

In Figure 7, we illustrate the nutrition maps for the Nutritionverse-Real dataset. We observe that the last row provides meaningful nutrition map for the calories. For the second row, the nutrition maps do not reflect the reality for calories, fats, and carbohydrates: the food patches are assigned lower nutrition values than the non-food patches. It is an important direction for future work to understand this behavior and to enforce the model to pay less attention to the non-food patches by using food masks.

6.4 Computational Cost

To illustrate the computational cost of nutrition adapters and backbones, we report the number of parameters, the inference time, and the number of floating point operations per second (FLOPS). The mean and standard deviation of the inference time are reported for 10 passes for an image of a resolution $H \times W = 224 \times 224$ with a patch size $P \times P = 14 \times 14$ and an embedding dimension $D = 1024$.

Table 2: Computational cost of adapters.

adapter	num. param.	inference time	FLOPS
Linear	5.12K	0.06 ± 0.03 ms	9.22M
MLP	6.31M	0.93 ± 0.01 ms	11.34G
Bilinear	12.6M	1.83 ± 0.01 ms	22.67G
Attention	13.65M	2.79 ± 0.09 ms	27.87G

Table 3: Computational cost of backbones.

backbone	num. param.	inference time	FLOPS
MAE	630.76M	40.01 ± 0.26 ms	323.79G
CLIP	303.18M	21.38 ± 0.23 ms	155.60G
DINOv2	304.37M	20.97 ± 0.31 ms	164.68G

In Table 2, we observe that the MLP adapter is twice as quick and a smaller model, than both the bilinear and the attention adapters. The attention adapter is the largest and the slowest one. In Table 3, we observe that both the CLIP and the DINOv2 model have a similar

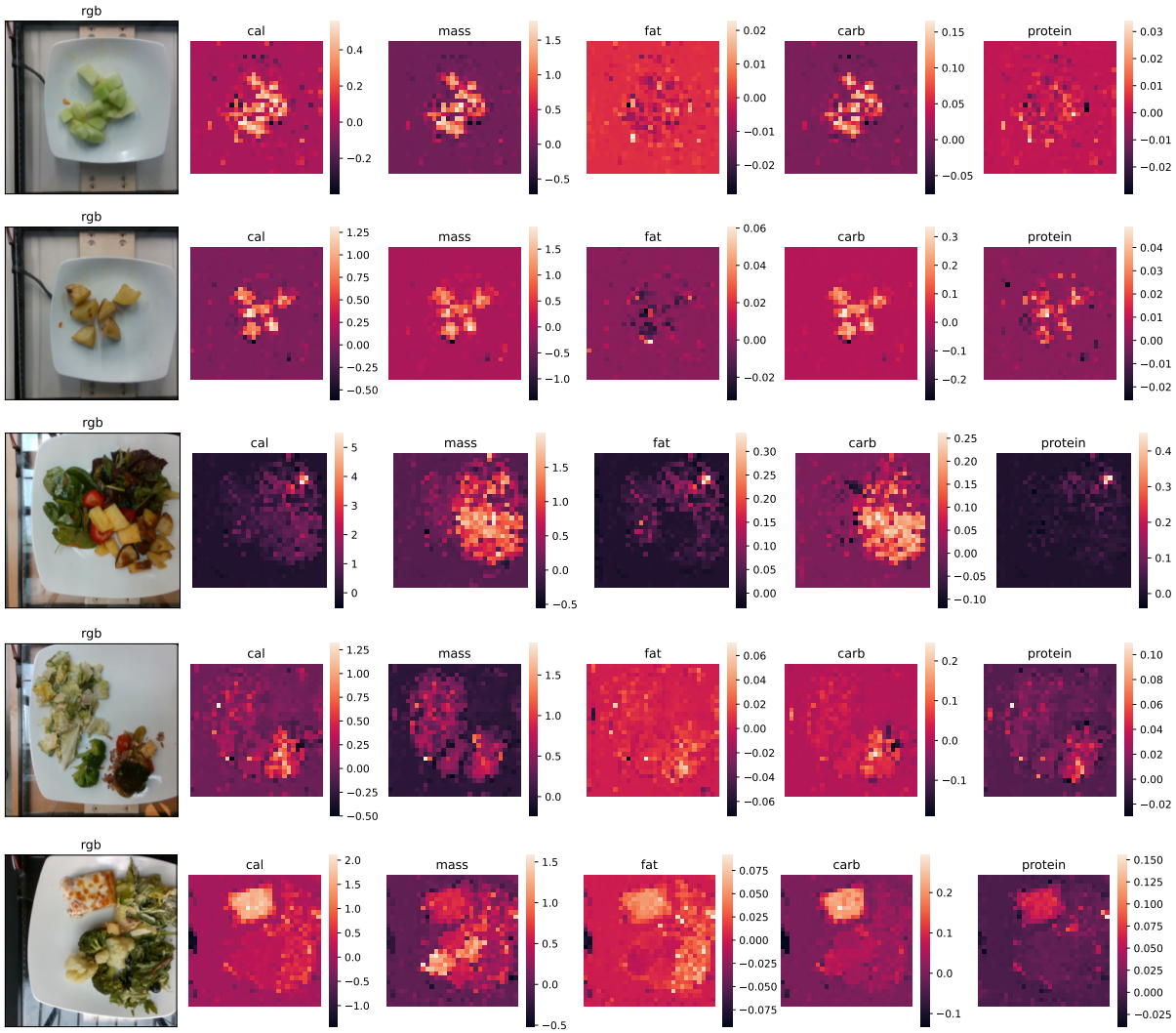


Figure 6: Illustration of nutrition maps for the Nutrition5k dataset [28]. The model uses the DINOv2 backbone and the bilinear adapter.

number of parameters, inference time, and FLOPS while the MAE is a bigger and slower model.

7 Conclusion and Discussion

We proposed an interpretable approach to nutrition prediction using large pre-trained vision models and lightweight interpretable adapters that output nutrition maps. We believe that such nutrition maps can serve as an explanation for nutrition prediction. The end-users may take them as a decision factor whether to trust the model prediction or not.

We note that the datasets [8, 28] that we used for training the nutrition adapters did not include the ground-truth nutrition maps. Such nutrition maps in our approach were induced by the models via minimizing the mean absolute error of the nutrition target prediction. Therefore, we can only empirically verify the obtained nutrition maps, and they cannot serve as a ground truth. It is an important direction for future works to collect datasets that contain such nutrition maps.

Acknowledgment

This research was partially funded by the University of Udine in the framework of the Strategic Plan 2022–25 – Interdepartmental Research Project CibiAmo.

References

[1] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. 2021. Neural Additive Models: Interpretable Machine Learning with Neural Nets. In *Advances in Neural Information Processing Systems*, Vol. 34. 4699–4711.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. arXiv:1607.06450 [stat.ML]

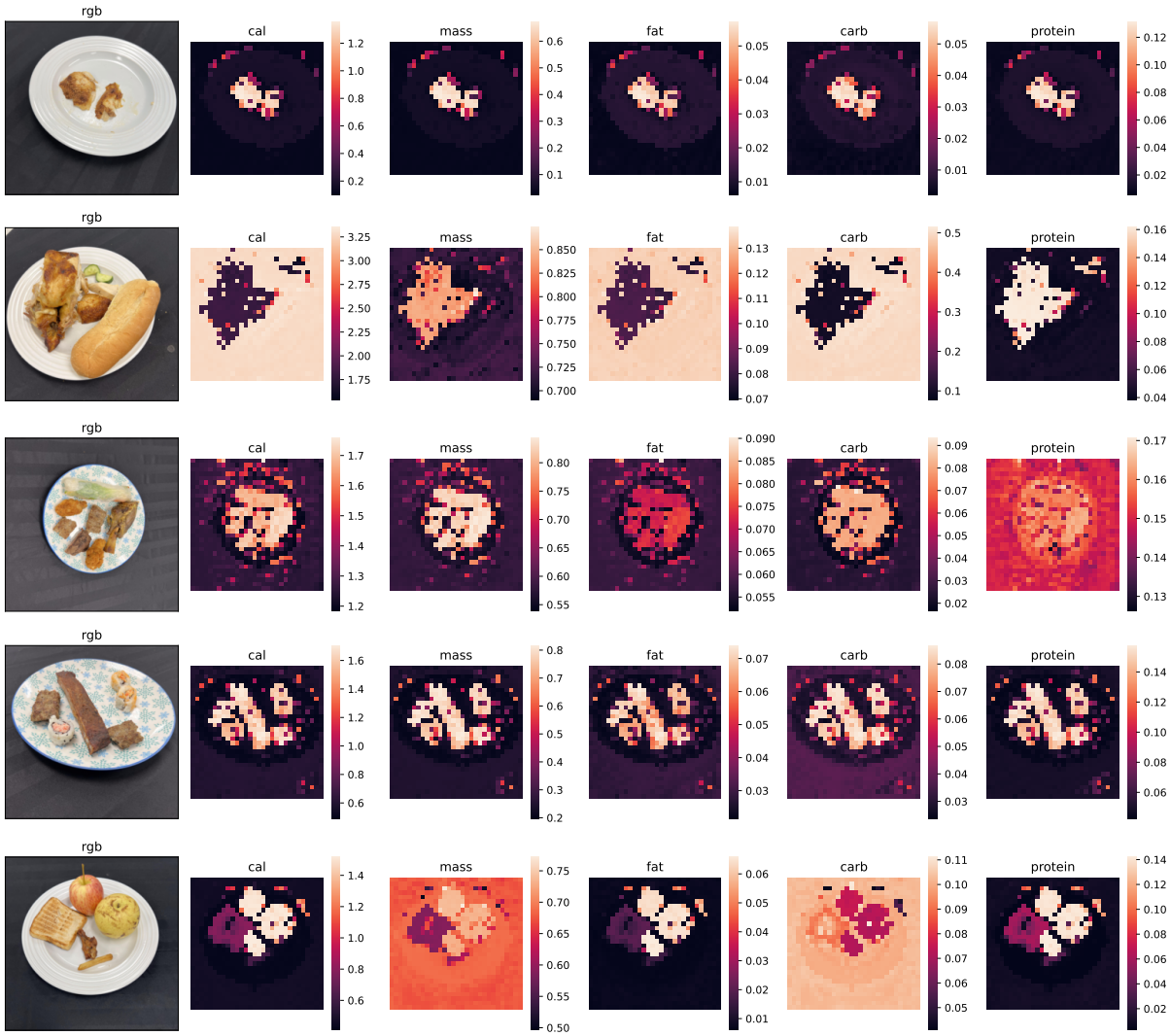


Figure 7: Illustration of nutrition maps for the Nutritionverse-Real dataset [8]. The model uses the DINOv2 backbone and the attention adapter.

[3] Oscar Beijbom, Neel Joshi, Dan Morris, Scott Saponas, and Siddharth Khullar. 2015. Menu-Match: Restaurant-Specific Food Logging from Images. *Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015* (02 2015), 844–851. doi:10.1109/WACV.2015.117

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101-Mining Discriminative Components with Random Forests. In *ECCV*.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Herve Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. doi:10.1109/iccv48922.2021.00951

[6] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. 2019. *This looks like that: deep learning for interpretable image recognition.* Red Hook, NY, USA.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929 [cs.CV]

[8] Chi en Amy Tai, Saeejith Nair, Olivia Markham, Matthew Keller, Yifan Wu, Yuhao Chen, and Alexander Wong. 2023. NutritionVerse-Real: An Open Access Manually Collected 2D Food Scene Dataset for Dietary Intake Estimation. arXiv:2401.08598 [cs.CV]

[9] Giovanni Maria Farinella, Dario Allegra, Marco Moltisanti, Filippo Stanco, and Sebastiano Battiato. 2016. Retrieval and classification of food images. *Computers in Biology and Medicine* 77 (2016), 23–39. doi:10.1016/j.compbiomed.2016.07.006

[10] Micah Goldblum, Hossein Souri, Renkun Ni, Manli Shu, Viraj Prabhu, Gowthami Somepalli, Prithvijit Chattopadhyay, Mark Ibrahim, Adrien Bardes, Judy Hoffman, Rama Chellappa, Andrew Gordon Wilson, and Tom Goldstein. 2023. Battle of the backbones: a large-scale comparison of pretrained models across computer vision tasks. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 1277, 29 pages.

[11] Yuzhe Han, Qimin Cheng, Wenjin Wu, and Ziyang Huang. 2023. DPF-Nutrition: Food Nutrition Estimation via Depth Prediction and Fusion. *Foods* 12, 23 (Nov. 2023). doi:10.3390/foods12234293

[12] Trevor Hastie and Robert Tibshirani. 1986. Generalized Additive Models. *Statist. Sci.* 1, 3 (1986), 297 – 310. doi:10.1214/ss/1177013604

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollar, and Ross Girshick. 2022. Masked Autoencoders Are Scalable Vision Learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. doi:10.1109/cvpr52688.2022.01553

[14] Linde S. Hesse, Nicola K. Dinsdale, and Ana I.L. Namburete. 2024. Prototype Learning for Explainable Brain Age Prediction. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 7888–7898. doi:10.1109/wacv57701.2024.00772

[15] Linde S. Hesse and Ana I. L. Namburete. 2022. INSightR-Net: Interpretable Neural Network for Regression using Similarity-based Comparisons to Prototypical Examples. arXiv:2208.00457 [cs.CV] https://arxiv.org/abs/2208.00457

[16] Matthew Keller, Chi en Amy Tai, Yuhao Chen, Pengcheng Xi, and Alexander Wong. 2024. NutritionVerse-Direct: Exploring Deep Neural Networks for Multitask Nutrition Prediction from Food Images. arXiv:2405.07814 [cs.CV]

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. *arXiv:2304.02643* (2023).

[19] Chiyu Ma, Jon Donnelly, Wenjun Liu, Soroush Vosoughi, Cynthia Rudin, and Chaofan Chen. 2024. Interpretable Image Classification with Adaptive Prototype-based Vision Transformers. arXiv:2410.20722 [cs.CV]

[20] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research* (2024).

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 8748–8763.

[22] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. 2021. Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges. arXiv:2103.11251 [cs.LG]

[23] Sascha Saralajew, Ashish Rana, Thomas Villmann, and Ammar Shaker. 2024. A Robust Prototype-Based Network with Interpretable RBF Classifier Foundations. arXiv:2412.15499 [cs.LG]

[24] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2019. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision* 128, 2 (Oct. 2019), 336–359. doi:10.1007/s11263-019-01228-7

[25] Wenjing Shao, Weiqing Min, Sujuan Hou, Mengjiang Luo, Tianhao Li, Yuanjie Zheng, and Shuqiang Jiang. 2023. Vision-based food nutrition estimation via RGB-D fusion network. *Food Chemistry* 424 (2023). doi:10.1016/j.foodchem.2023.136309

[26] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2818–2826. doi:10.1109/CVPR.2016.308

[27] Chi-en Amy Tai, Matthew Keller, Saeejith Nair, Yuhao Chen, Yifan Wu, Olivia Markham, Krish Parmar, Pengcheng Xi, Heather Keller, Sharon Kirkpatrick, and Alexander Wong. 2023. NutritionVerse: Empirical Study of Various Dietary Intake Estimation Approaches. In *International Workshop on Multimedia Assisted Dietary Management*. doi:10.1145/3607828.3617799

[28] Quin Thames, Arjun Karpur, Wade Norris, Fangting Xia, Liviu Panait, Tobias Weyand, and Jack Sim. 2021. Nutrition5k: Towards Automatic Nutritional Understanding of Generic Food. *CVPR* (2021).

[29] Hugues Turbé, Mina Bjelogrlic, Gianmarco Mengaldo, and Christian Lovis. 2025. Tell me why: Visual foundation models as self-explainable classifiers. arXiv:2502.19577 [cs.CV]

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc.

[31] Weiqiu You, Helen Qu, Marco Gatti, Bhuvnesh Jain, and Eric Wong. 2023. Sum-of-Parts: Self-Attributing Neural Networks with End-to-End Learning of Feature Groups. arXiv:2310.16316 [cs.LG]

[32] Yaping Zhao, Ping Zhu, Yizhang Jiang, and Kaijian Xia. 2024. Visual nutrition analysis: leveraging segmentation and regression for food nutrient estimation. *Frontiers in Nutrition* 11 (2024).