A New CNN-based Single-Ingredient Classification Model and Its Application in Food Image Segmentation

Ziyi Zhu and Ying Dai [*]

Posted Date: 14 August 2023

doi: 10.20944/preprints202308.1030.v1

Keywords: CNN architecture; Single Ingredient Classification model; Food Ingredients segmentation; evaluation metrics; Hierarchical Multi-level learning

Article

A New CNN-Based Single-Ingredient Classification Model and Its Application in Food Image Segmentation

Ziyi, Zhu [1] and Ying Dai [2,*]

[1] Iwate Prefectural University, Takizawa, Iwate, Japan; g236t002@s.iwate-pu.ac.jp
[2] Iwate Prefectural University, Takizawa, Iwate, Japan
* Correspondence: dai@iwate-pu.ac.jp

Abstract: It is important for food recognition to separate each ingredient within a food image at the pixel level. In this paper, we propose a new approach to segment ingredients by utilizing a CNN-based Single-Ingredient Classification Model. In detail, we firstly introduce a standardized biological-based hierarchical ingredient structure and construct a single-ingredient image dataset based on this structure. Then, we build a single-ingredient classification model based on a novel convolutional neural network (CNN) architecture that utilizes an attention mechanism. Afterwards, we propose a new framework for segmentation using the above single-ingredient classification model as the backbone. In this framework, two methods are involved in segmenting ingredients in the food images. We introduce five evaluation metrics (IoU, Dice, Purity, Entirety, Loss of GTs) to assess the performance of ingredient segmentation in terms of ingredient classification. Extensive experiments demonstrate the effectiveness of the proposed method, achieving an maximal mIoU of 0.65, mDice of 0.77, mPurity of 0.83, mEntirety of 0.80, and mLoGTs of 0.06 on the FoodSeg103 dataset. The results confirm that our CNN-based architecture achieves higher segmentation performance compared to ResNet18 and EfficientNet-B0 when used as the backbone for ingredient segmentation. We believe that our ingredient segmentation approach lays the foundation for subsequent ingredient recognition.

Keywords: CNN architecture; Single Ingredient Classification model; Food Ingredients segmentation; evaluation metrics; Hierarchical Multi-level learning

1. Introduction

With the rapid development of deep learning techniques in recent years, food computing [1] has emerged as an interesting field owing to its wide range of applications in the health, culture, and other domains. It is important to analyze and understand food images from different perspectives such as nutrition estimation, food choices, food diaries, and healthy eating recommendations.

Among various tasks in food computing, food recognition has attracted considerable research interest in recent years. Existing studies include deep-based recognition, which leverages different deep-food recognition models [2][3][4][5]. However, food recognition is a challenging task owing to high intra-class variance and high inter-class similarity. Martinel et al. [6] proposed a novel network for food recognition which exploited vertical food traits. Additionally, some studies utilized additional context information, such as ingredients and location, to improve recognition performance. For example, based on ingredients and restaurant information, Zhou et al. [7] exploited the rich relationships among categories using bipartite graph labels for food recognition. Min et al. [8] incorporated the ingredient information of food items to localize multiple ingredient regions, and fused these regional features into the final representation for recognition. Qiu et al. [9] adopted a progressive training strategy to mine the ingredient regions of a food image to achieve an accurate food recognition. However, food categories are almost unlimited and their names vary across different areas. This results in significant technical difficulties for food recognition. Accordingly, researchers have begun to focus on food ingredient recognition because ingredient categories are limited and usually defined according to standard food taxonomy.

doi:10.20944/preprints202308.1030.v1

Food ingredient recognition involves the automatic identification of multiple ingredients in a food image. Recognition of food ingredients is helpful for dietary assessment, food tracking, and nutritional analysis. However, food ingredient recognition remains a challenging task because of the high variety in the appearance of ingredients produced using different cooking methods. As depicted in Figure 2, the visual appearance of the same ingredient can vary significantly such as egg, while different ingredients may exhibit similarities such as spinach and Bok choy. This poses a challenge in accurately classifying ingredients.

Food ingredient recognition typically incorporates multi-label classifications. For example, some researchers introduced a CNN model to learn ingredients from dish images using a multilabel learning method [10][11]. To enhance performance, numerous approaches based on multi-task and region-based deep learning have been proposed to improve accuracy [12][13]. Some researchers have utilized visual attention techniques to develop a multi-attention network that can identify and localize ingredient regions on various scales, leading to improved performance. Furthermore, Chen et al. [14] deployed a multi-relational graph convolutional network that considers the relations between different ingredients, including ingredient hierarchy, co-occurrence, and cooking and cutting methods.

These previous works, however, still face challenges including:

1) Existing models are based on multi-label ingredient recognition, because they are typically supervised with multiple ingredient labels within the dish image. However, this method is not optimal because it does not directly and accurately learn visual ingredient representations, and is often influenced by the mutual interference of adjacent ingredients in the images during training and testing.

2) Lack of standardized ingredient datasets which cover a wide range of ingredient categories.

To exclude the inference of adjacent ingredients in ingredient recognition, some studies [21][22] explored methods for food ingredient segmentation. These studies trained segmentation networks on pixel-level annotated ingredient datasets such as FoodSeg103 [21]. However, pixel-level annotation for each image is time-consuming.

A database called AI4Food-NutritionDB was developed in [15]. This database categorizes foods based on a nutritional 4-level pyramid structure and analyzes food recognition tasks using a nutrition taxonomy. However, this database does not account biological inherent hierarchical structure among the ingredients.

To address these challenges, we aim to construct a single-ingredient images dataset with a standardized biological-based hierarchical structure that covers a wide range of ingredient categories. A single-ingredient image indicates that only one type of ingredient was present in the image. We collect various single-ingredient images for each ingredient category, which is cooked by various ways with different visual appearances.

Furthermore, we propose a single-ingredient classification model by training a novel Attention-based CNN network on the above dataset. We also incorporate the hierarchical relationships between ingredients into a multilevel single-ingredient classification network by employing a multi-task learning approach.

We propose a new food ingredient segmentation framework that separates visible ingredients in food images to address the mutual interference of multiple ingredients in the food images. This framework uses the single-ingredient classification model as a backbone to extract feature maps from the food images and generate the corresponding ingredient masks to gain the ingredient segments.

Our main contributions can be concluded as:

1. A single-ingredient image dataset was constructed based on food taxonomy standards [16][17] to train the single-ingredient classification model.
2. A novel CNN-based architecture utilizing attention mechanism for single-ingredient classification was proposed.
3. A new multi-ingredient segmentation framework utilizing the above model as an extractor of feature maps was proposed. Furthermore, two methods were introduced for processing the above feature maps to generate ingredient masks for the ingredient segmentation.

The organization of this paper is as follows. In Section II, we provide a review of relevant works. In Section III, we introduce two new datasets for the proposed methods. In Section IV, we introduce a novel CNN-based architecture with an attention mechanism for the single-ingredient classification model. In Section V, we present a new multi-ingredient segmentation framework that utilizes the aforementioned model. Section VI covers the introduction of five metrics for evaluating ingredient segmentation. In Section VII, we analyze the performance of the single-ingredient classification model and the proposed multiple-ingredient segmentation framework. Finally, in Section VIII, we present our conclusions.

2. Related work

In this section, we briefly review several related studies including food ingredient segmentation, multi-task learning, and K-Means Clustering.

2.1. Food Ingredient Segmentation

Before discussing food ingredient segmentation, we briefly introduce food segmentation. The food segmentation [23] aims to segment each food category and its pixel-wise location in a food image. Aguilar et al. [18] combined food/non-food binary masks and food localization bounding boxes to achieve food segmentation. Sharma et al. [19] introduced a network called GourmetNet, which adopts Waterfall Atrous Spatial Pooling (WASPv2) module, and employs dual attention (channel and spatial) mechanisms for multi-scale waterfall features to improve the food segmentation performance. Okamoto et al. [20] introduced a region-based segmentation model for multiple-dish segmentation. Liang et al. [24] proposed a ChineseFoodSeg approach which using color and texture features of superpixels to separate different dishes from single plate.

Food ingredient segmentation has recently emerged as a promising task for identifying each ingredient category and its specific location within a food image at the pixel level. However, ingredient segmentation poses notable challenges due to the inherent high variability of ingredients. For instance, eggs exhibit significant intra-class variance depending on the cooking method employed, such as boiling or steaming. On the other hand, certain categories, like spinach and kale, present a high inter-class similarity as they are both green leaves and are often prepared in recipes of similar shapes and sizes. One additional challenge in ingredient segmentation is the similarity between certain ingredients and the background. Wu et al. [21] proposed a food image segmentation framework, which consists two modules: Recipe Learning Module (ReLeM) and Image Segmentation module. Specially, ReLeM incorporates recipe information and integrates recipe embedding with the visual representation of a food image to enhance the visual representation of ingredient. Wang et at. [25] proposed a Swin Transformer-based pyramid network to combine multi-scale features from global and local region of food image for food image segmentation. Xia et al. [22] proposed a network consisting of two subnetworks to refine the boundary of the ingredient segmentation. Specifically, this study incorporated both Hyperspectral Imaging (HSI) and RGB images as inputs for feature extraction. The latest study, Segment Anything [26] introduced an efficient transformer-based model that unifies various segmentation tasks into a general framework to implement class-agnostic instance segmentation.

However, to the best of our knowledge, all the aforementioned studies rely on training proposed segmentation models on training datasets with pixel-level annotations. Acquiring such datasets, especially for food ingredient segmentation, requires a significant amount of manual labeling, which is time-consuming and prone to errors. In contrast to these methods, we propose a weakly supervised segmentation approach that requires only single-ingredient images and their corresponding labels, thereby reducing the need for pixel-level annotation.

2.2.Food-related Public Datasets

Along with research on food ingredient recognition, there are some large-scale datasets such as VIREO Food-172 [27] and ISIA Food 500 [28]. VIREO Food-172 is one of the first datasets to consider these ingredients. It contains 110,241 images from 172 food categories. The images were manually annotated based on 353 ingredients.

Several public datasets are available for food segmentation. Food-201 [29] contains 12,093 images with 201 food categories and 29,000 dishes. UECFoodPix COMPLETE [30] is another widely used food image dataset that contains 10,000 food images across 102 food categories and 14,011 masks. However, these datasets only provide dish-level and not ingredient-level segmentation masks. FoodSeg-103, proposed in [21], consists of 7,118 images and more than 40,000 masks, covering 103 food ingredient categories. FoodSeg-103 is the first large-scale food image dataset with ingredient-level segmentation masks. HSIFoodIngr-64, proposed in [22], contains 21 dish and 64 ingredient categories. This study provides annotated HSI images that contain more informative properties of ingredients.

In this study, we use the FoodSeg103 dataset to evaluate the performance of the proposed method for ingredient segmentation, and use the UECFoodPix COMPLETE dataset to evaluate the performance of food segmentation. As the dataset from study [22] has not been made publicly available yet, we do not compare our results with it.

2.3. Multi-task Learning

Multi-task learning is widely used approach in computer vision and plays a critical role in image classification [31][32], object detection [33], and semantic segmentation [34][35]. A general pipeline involves learning shared representations from multiple tasks to leverage multiple related properties to improve the performance of each task. Additionally, it provides task-specific representations for each task. Hierarchical multi-task learning is a sub-field strategy for multi-task learning, where tasks are organized in a hierarchical structure based on their relationships and dependencies. Li et al. [36] proposed a multi-task network cascade network that consists of three stages for each task and, unlike common methods, whereas the latter stage relies on the previous stage. This study proposed a sequential feature-sharing method among tasks.

Another study [37] argued that among all the tasks, some tasks are simpler than others, referred to as low-level tasks, and high-level tasks in contrast. They introduced a hierarchical network by setting the supervision of low-level tasks in the bottom layers and high-level tasks in the top layer of the model.

Our study was inspired by the bottom-up supervisor-setting method of study [37]. We introduced a multi-level learning strategy for single-ingredient classification.

2.4. K-Means Clustering

Clustering is a simple and effective method of image segmentation. Specifically, the k-means clustering is most widely used in many works [38][39]. The main concept of image segmentation using the k-means method is to partition a collection of pixels in an image into k clusters, based on their similarity. Zheng et al. [39] introduced an adaptive k-means algorithm for LAB color space to improve the performance of image segmentation. Caron et al. [40] proposed a deep neural network called Deep Embedded Clustering (DEC), which incorporates both an autoencoder and clustering module. Van et al. [41] introduced a method for clustering learned pixel embeddings into groups to address unsupervised segmentation.

3. Dataset

In this section, we will first introduce the Hierarchical Ingredient Structure based on food taxonomy standards [16] [17]. Secondly, we will introduce the Single-Ingredient Image Dataset for training a single-ingredient classification model. Lastly, we will introduce the Multiple-Ingredient Image Dataset for evaluating ingredient segmentation.

3.1. Hierarchical Ingredient Structure

In a previous study [42], we proposed a three-level structure for ingredient categories based on the Japanese food taxonomy [16] and [17], as shown in Figure 1. This structure is biological, and now we expand it by adding Level 4 ingredient categories based on the same taxonomy as in [17]. As a result, we have a four-level hierarchical structure for the ingredient categories.

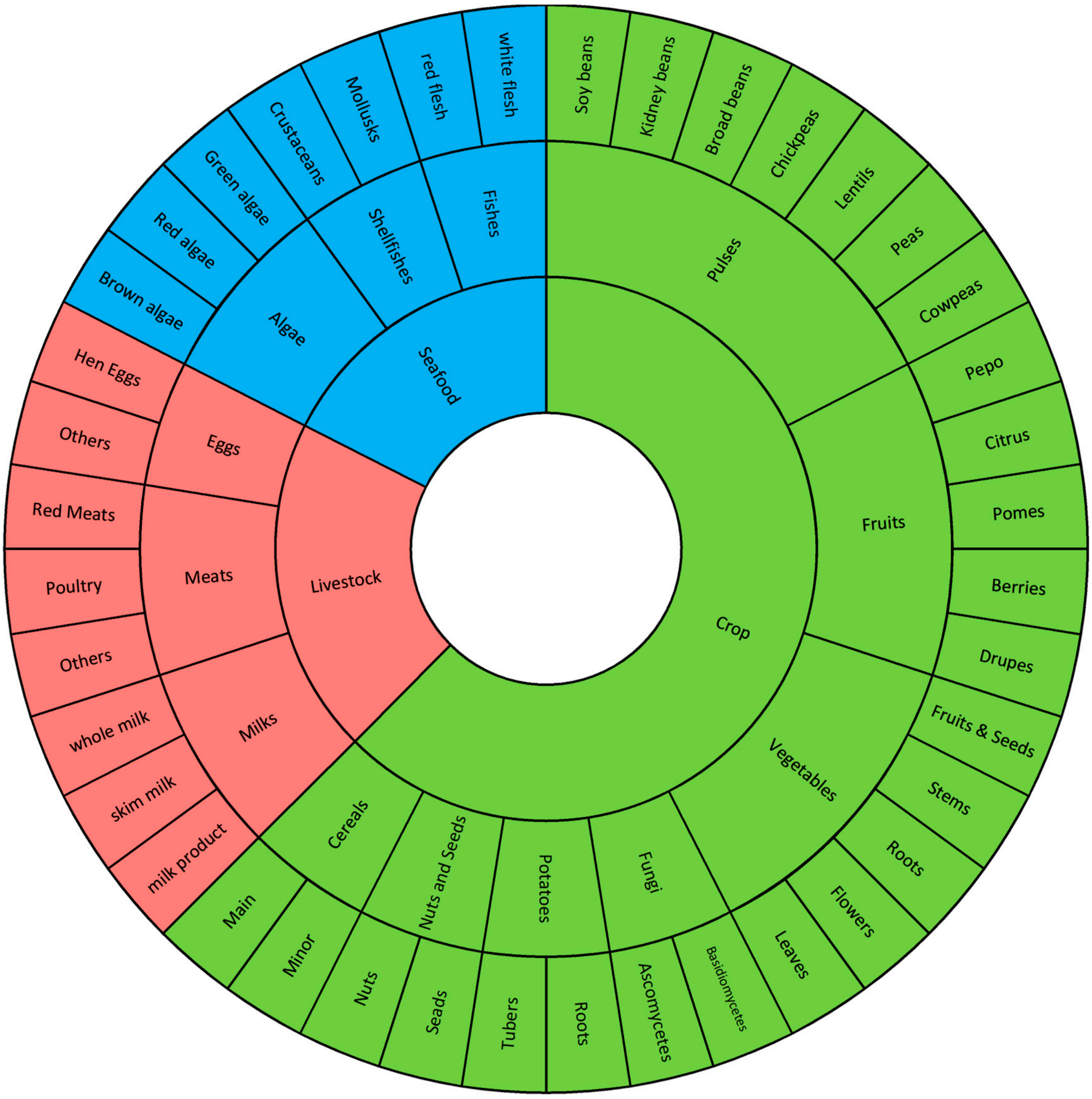

Figure 1. Architecture of First 3 levels of hierarchical ingredient structure.

In this four-level hierarchical structure, level 1 ingredient categories are defined based on the standard described in [16], including Crop, Livestock, and Seafood. Level 2 to level 4 ingredient categories is defined based on another standard described in [17]. Each ingredient category at a lower level belongs to only one type of ingredient at a higher level. For example, as shown in Figure 1, "Fruits", "Vegetables" and "Meats" are level 2 ingredient categories, "Fruits" and "Vegetables" belong to "Crop" (level 1 ingredient category), while "Meats" belongs to "Livestock". As a results, level 2 includes 13 ingredient categories, level 3 includes 32 ingredient categories, and level 4 includes 110 commonly used ingredient categories, which providing a comprehensive coverage of food taxonomy.

3.2. Single-Ingredient Image Dataset (SI110)

In this work, we construct a novel single-ingredient image dataset. We follow several criteria for data collection. First, we exclude invisible ingredients such as salt and sugar because our goal is to recognize visually observable ingredients in food images. Second, we ensure that each single-ingredient image contains only one type of ingredient as defined in the level 4 category list. This is achieved by capturing single-ingredient images or extracting single-ingredient regions from the food images manually. Third, based on the different cooking conditions, we collect as many visual variants as possible for each ingredient category. For example, we gather different visual appearances of eggs, potatoes, and pumpkins under various cooking conditions (Figure 2). Finally, we ensure that 5-10 images are selected for each type of visual appearance of each ingredient category to prevent training bias towards specific visual appearance.

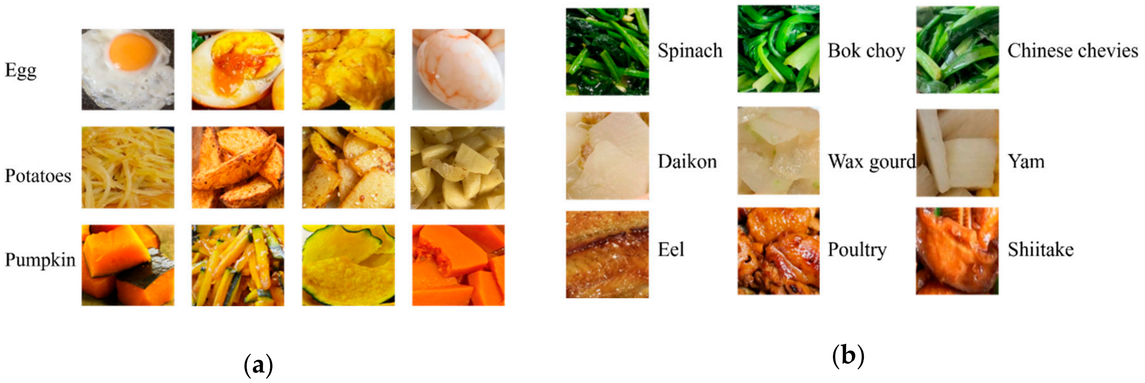

Figure 2. Intra-class variance and Inter-class similarity(b). (a): Samples of three ingredients with high intra-class variance. (b): Samples of three sets exhibiting high inter-class similarity.

In the scope of 110 food ingredient categories, we collect food images from Google Pictures using English, Chinese, and Japanese keywords including these ingredients. Subsequently, we perform several processing on the collected images, including: 1) Cropping out the regions having the individual ingredient. These cropped regions were then used as single-ingredient samples. 2) ensuring that each food ingredient has 5 to 10 samples for each visual variation. Currently, SI110 contains 10,750 single-ingredient images, including 110 level 4 ingredient categories, covering the entire range of food taxonomy. The images are then assigned to corresponding categories at the three upper levels. The distribution of ingredient categories for each level is shown in Figures 3-6, respectively. The detailed names corresponding to the notations of the horizontal coordinates in Figure 6 are shown in the Appendix A.

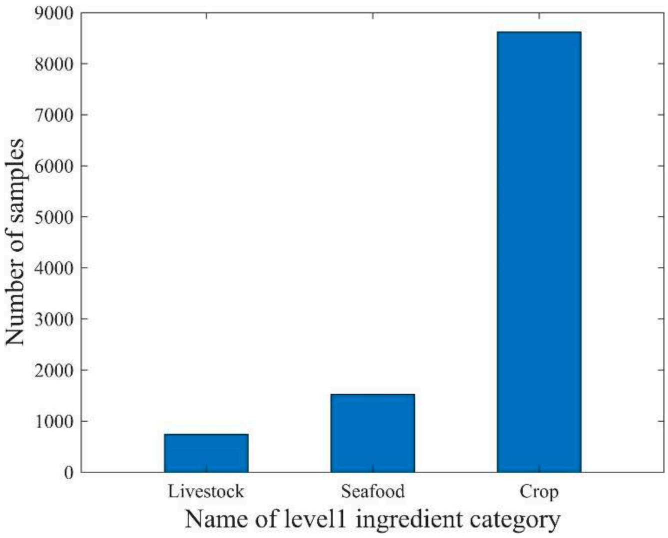

Figure 3. The distribution of sample counts for ingredient categories in level 1 of the SI110 dataset.

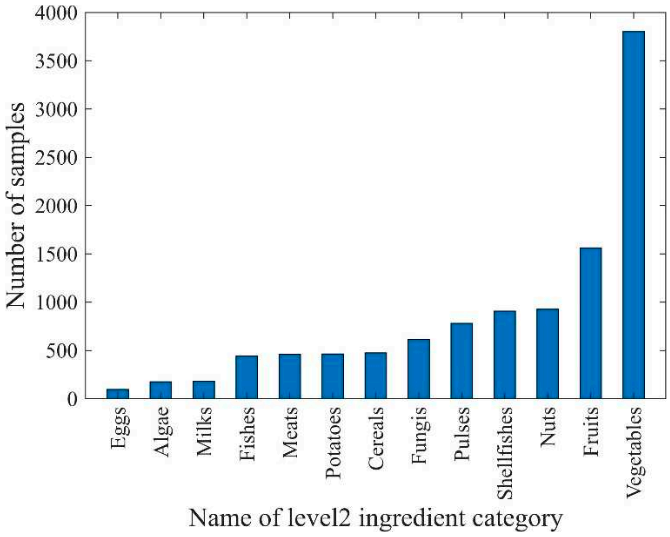

Figure 4. The distribution of sample counts for ingredient categories in level 2 of the SI110 dataset.

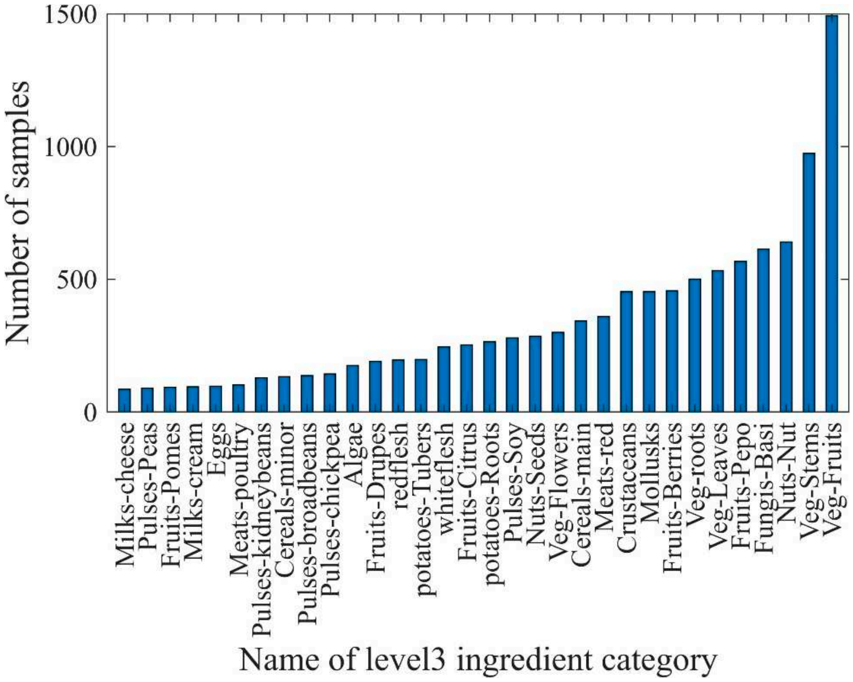

Figure 5. The distribution of sample counts for ingredient categories in level 3 of the SI110 dataset.

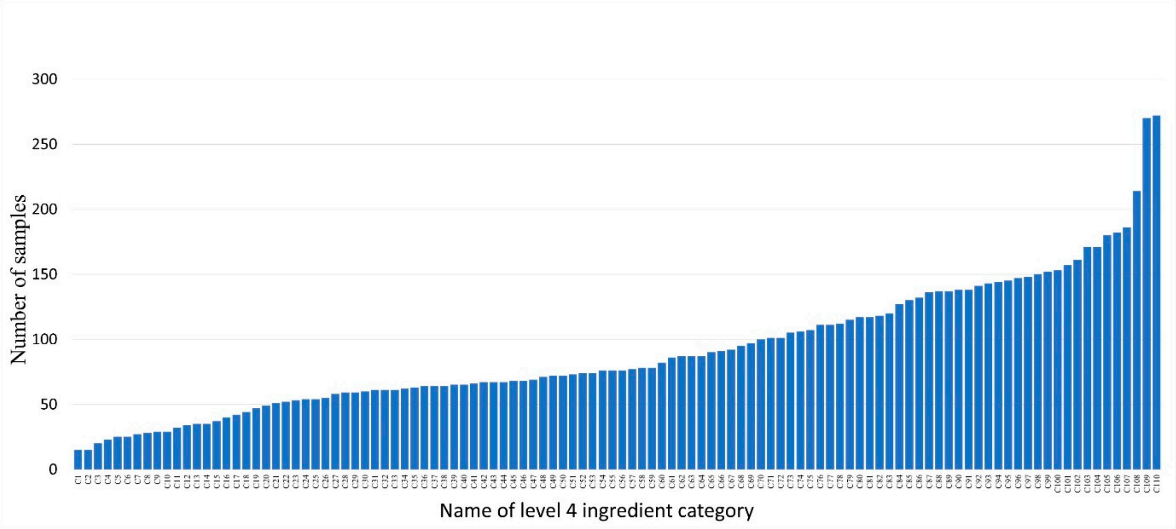

Figure 6. The distribution of sample counts for ingredient categories in level 4 of the SI110 dataset.

The SI110 dataset is randomly divided into 80% for training and 20% for testing the single-ingredient classification model.

Data samples at each level follows a long-tailed distribution. For example, at level 4, ingredient categories that offer many different cooking ways, such as shrimp and wheat products, contain more than 250 samples. However, ingredients that are not prepared in many ways, such as green caviar and raspberry contain only about 20 samples.

3.3. Multiple Ingredient Food Image Dataset (MIF110)

In addition to constructing a single-ingredient image dataset for training the single-ingredient classification model, we further construct a multiple-ingredient food image dataset (MIF110) to evaluate the performance of multiple-ingredient segmentation in food images. Unlike the single-ingredient image dataset, this dataset includes food images containing multiple ingredients. Similar to the SI10 dataset, we collected cooking images from Google Pictures within the scope of 110 food ingredients. Subsequently, we extracted the cooking regions from each image to create samples for the MIF110 dataset. Currently, MIF110 contains 2066 food images, with an average of 2.72 ingredients per image. We demonstrate the distribution of food images with different numbers of ingredients in Figure 7. The number of images containing two ingredients is the highest.

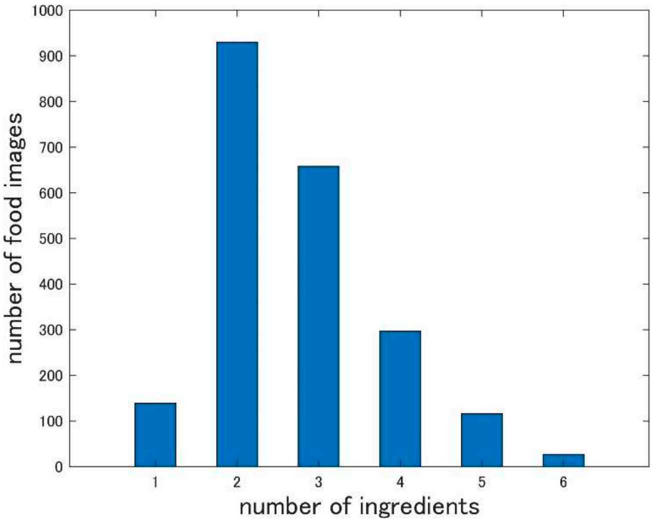

Figure 7. Distribution of food images with different ingredient numbers in MIF110.

4. Single ingredient classification model

In this section, we propose a new CNN-based architecture for a single-ingredient classification model trained on the SI110 dataset.

4.1. Proposed CNN-based architecture

In this subsection, we present AttNet, a novel CNN-based architecture that incorporates an attention mechanism. The complete structure of an AttNet is shown in Figure 8. This network consists of eight CNN blocks with identical structures. Each CNN block has four layers. The first layer is the convolutional layer. Subsequently, a batch normalization layer is added. Similar to the architecture of the EfficientNet [43], we adopt a 3D attention mechanism [44] for each CNN Block. Thus, we then follow a sigmoid layer to compute the activation value of the feature map from the batch normalization layer. Its principle is to map the input feature values to a probability range between 0 and 1. Finally, we add an element-wise multiplication layer to multiply the activation value with the feature map output from the batch normalization layer. Finally, we add a global average pooling layer after the last CNN block and added a classification layer at the end. We represent the whole network parameter in Table 1. Different channel numbers are set for the convolutional layers in each CNN block. As the network deepened, the number of channels in the convolutional layer increased. In this study, we set different channel numbers for the convolution layer in the eight CNN blocks as follows: {64, 128, 128, 256, 256, 512, 512, C#}, where C# is the number of ingredient categories.

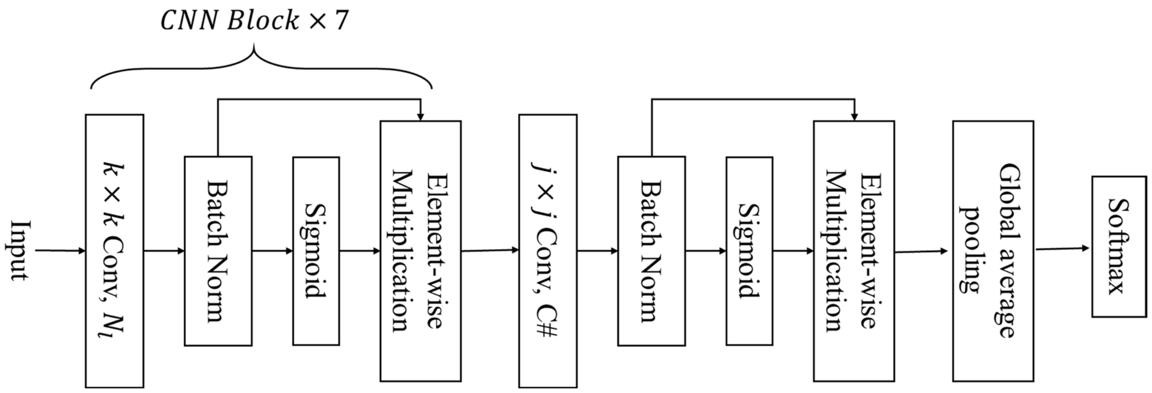

Figure 8. Architecture of CNN-based AttNet network, where k and j mean the kernel size of convolutional layer, $\boldsymbol{N_l}$ represents the channel number of the Convolutional layer in the CNN block l, and C# indicates the number of categories.

In this work, we explore the use of two different sizes for convolutional layer in the CNN blocks. Firstly, we use the kernel size of 1. We refer to this network as the AttNet(1). We chose to only use 1 × 1 convolutional layer because it significantly reduces computational cost and memory size. Secondly, we use the kernel size of 3 which is commonly utilized in CNN networks. We refer to this network as the AttNet(3). We further propose two variants of AttNet, by setting one type of kernel size at first seven CNN blocks, and setting another type of kernel size at last CNN blocks, which are :1) AttNet(1+3), where the first seven CNN blocks use 1 × 1 convolutional layers and the last CNN block uses a 3 × 3 convolutional layer; 2) AttNet(3+1), where the first seven CNN blocks use 3 × 3 convolutional layers and the last CNN block uses a 1 × 1 convolutional layer.

Furthermore, we modify both the pre-trained ResNet18[43] and EfficientNet-B0 by removing all layers after their last CNN block, adding a 1 × 1 convolutional layer with C# channels, followed by a global average pooling layer, and finally adding a classification layer. Finally, all models, including AttNets, modified ResNet18 [45], and EfficientNet-B0, are trained on the SI110 train dataset.

doi:10.20944/preprints202308.1030.v1

4.2. Training Models

In this subsection, we introduce a method for training a single-ingredient classification model using the SI110 dataset. First, we present the baseline work, which is a single-level learning method that trains only the classification model for level 4 ingredient categories. This method aims to predict the level 4 ingredient categories. The models trained using this method are referred to as the SLMs (Single Level Models). By contrast, we propose a multi-level learning strategy for single-ingredient classification that simultaneously leverages four levels of ingredient information from the hierarchical structure. A diagram of the proposed method is shown in Figure 9. Furthermore, we employ a bottom-up feature sharing mechanism by setting a CNN Block for each level after 7th CNN Block. These CNN Blocks are sequentially stacked from level 4 to level 1. Additionally, for each level's CNN Block, we include a global average pooling layer (gap) and followed by a Softmax layer to perform classification for each level. We demonstrate the detail model structure of this multi-level feature sharing mechanism in Figure 10.

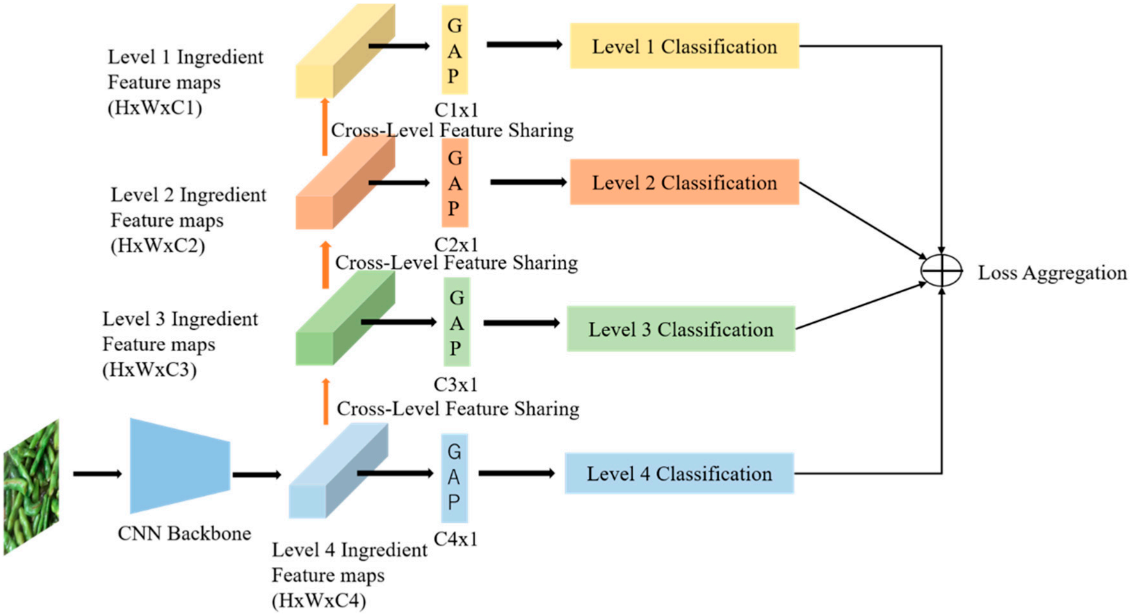

Figure 9. The diagram of multi-level learning of multi-level single-ingredient classification that utilizes a bottom-up feature sharing mechanism to facilitate multi-level learning. Where {C1, C2, C3, C4} indicates the number of categories at each level.

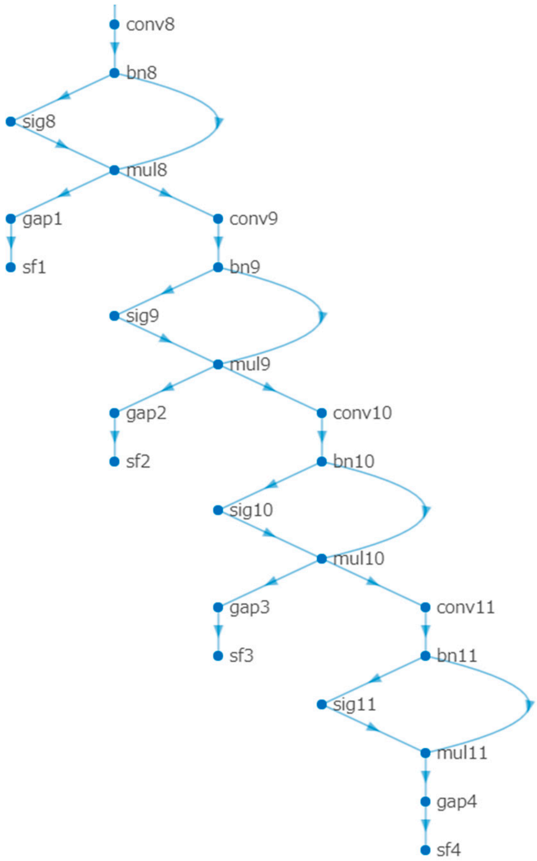

Figure 10. The diagram of multi-level feature sharing mechanism with bottom-up feature sharing mechanism. CNN Blocks are sequentially stacked from level 4 to level 1.

Table 1. Architecture of proposed AttNet network.

Stage	Operator	Resolution	Channels
1	Input	224×224	3
2	CB, k x k conv, stride 2	112×112	64
3	CB, k x k conv, stride 2	56×56	128
4	CB, k x k conv, stride 2	28×28	128
5	CB, k x k conv, stride 1	28×28	256
6	CB, k x k conv, stride 1	28×28	256
7	CB, k x k conv, stride 1	28×28	512
8	CB, k x k conv, stride 1	28×28	512
9	CB, j x j conv, stride 1	28×28	C#
10	avgpool, softmax	1×1	C#

The models trained using this multi-level learning method are referred to as MLMs (Multiple Level Models). During the training process, we compute the standard cross-entropy loss for each level's ingredient classification, and optimize the weighted sum of the four losses with different weights. To ensure balanced training, we assign a higher weight to bottom levels than upper levels. This is because the number of bottom ingredient categories is greater than the number of upper ingredient categories, and we aim to account for this discrepancy in the training process. We define the total loss function for multi-level ingredient classification as follow.

$$L_{total} = \sum_{i}^{4} \lambda_i L_i(sf(z^i, y^i) \quad (1)$$

Where L_i means the cross-entropy loss function for level i; sf means the Softmax function, z^i represents the output of global average pooling layer. y^i represents the ground-truth class label for level i. In this work, we adopt a fixed set of weights {1.0, 0.5, 0.3, and 0.1} in a descending order from level 4 to level 1. The purpose of assigning decreasing weights to classification tasks at different levels is to emphasize the importance of level 4 ingredients during the training process. We decrease the weights of tasks at upper levels. Based on the hierarchical structure, we gradually decrease the importance of upper-level tasks by considering the cross-level distance between different levels and the fourth level.

5. Ingredient segmentation framework

Based on the above single-ingredient classification model, we propose a new framework for ingredient segmentation, as shown in Figure 11. The framework involves extracting feature maps of multiple-ingredient food images using a pre-trained single-ingredient classification model. The input image is denoted by X. Feature maps are extracted from the last convolutional layer of the model and are denoted by f(X). where $f(X) \in \mathbb{R}^{H \times W \times C4}$ and C4 is the number of level 4 ingredient categories. Subsequently, the feature maps f(X) is processed to generate the ingredient masks. In the following section, we introduce two methods for feature-map processing.

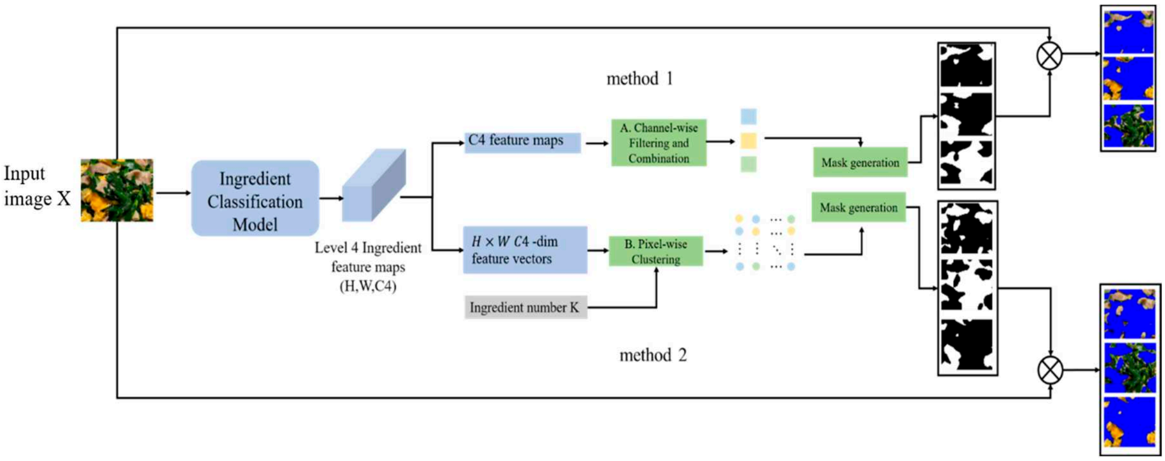
Input image X
Ingredient Classification Model
Level 4 Ingredient feature maps (H,W,C4)
method 1
C4 feature maps
A. Channel-wise Filtering and Combination
Mask generation
$H \times W$ $C4$ -dim feature vectors
B. Pixel-wise Clustering
Ingredient number K
Mask generation
method 2

Figure 11. The diagram multiple ingredient segmentation framework.

In Method 1, we first transform the 3D feature map f(X) into a 2D feature map with C4 channels. The first step is to filter the C4 feature maps. We calculate the global average value of each feature map, then normalize it using the sigmoid function to compute a score, and then selected the feature maps with scores greater than the threshold of 0.5. The second step is to combine the feature maps. We calculate the correlation coefficient for each pair of the filtered feature maps. When the correlation coefficient exceeded the threshold of 0, a pair of feature maps is merged into one feature map. The intuition behind this step is that the positively correlated feature maps tend to encode redundant information. By merging these results, we obtain a more complete activation result for a particular component. Finally, we binarize all processed feature maps to create masks for ingredient segmentation.

In Method 2, we transform the 3D feature map into H × W pixel-wise feature vectors with C4-dimensionals. We then apply k-means clustering to these vectors to obtain K clusters, where K represents the number of ingredients in the dish image. Each cluster, which is composed of pixels, generates a mask for the ingredient segmentation.

Finally, we resize the masks to the same size as the input image and then applied element-wise multiplication to each mask and input image to obtain the segments of the ingredients in the food image.

6. Segmentation evaluation metrics

In this section, we introduce five metrics to evaluate the performance of ingredient segmentation. IoU is a measurement of the overlap between the predicted segmentation mask and the ground truth mask. It is calculated by dividing the intersection of the predicted and ground truth regions by their union. Additionally, the Dice coefficient is another metric which quantifies the similarity between the predicted and ground truth masks. It is calculated by taking twice the intersection area of the predicted and ground truth masks, and dividing it by the sum of the areas of the predicted mask and ground truth mask. In this study, we employ these metrics to evaluate the performance of food ingredient segmentation.

In addition to IoU and Dice metrics, the purity and entirety of segmentation are crucial aspects for recognizing ingredients in the next step. Furthermore, the region loss of the ingredient of the ground truth (LoGT) should also be considered for the evaluation because the loss of the ingredient makes it impossible to be recognized. The Definitions of Purity and Entirety are equivalent to those proposed in [46]. Specifically, Purity measures the ratio of the ingredient of the ground truth (GT) contained in a segment, and Entirety measures the ratio of the segment contained in an ingredient of the GT.

All mentioned metrics are calculated using Equations 2-6, and it should be noted that IoU is a comprehensive metric of Purity and Entirety.

$$\text{intersection over union (IoU)} = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

$$\text{Dice} = \frac{2 \times |\mathrm{A} \cap \mathrm{B}|}{|\mathrm{A}| + |\mathrm{B}|} \tag{3}$$

$$\text{purity} = \frac{|\mathrm{A} \cap \mathrm{B}|}{|\mathrm{B}|} \tag{4}$$

$$\text{entirety} = \frac{|\mathrm{A} \cap \mathrm{B}|}{|\mathrm{A}|} \tag{5}$$

$$\text{LoGTs} = \frac{\left|\bigcup_{\mathrm{i}}^{\mathrm{I}} \mathrm{A_i}\right| - \left|\bigcup_{\mathrm{i}}^{\mathrm{I}} \mathrm{A_i} \cap \bigcup_{\mathrm{j}}^{\mathrm{J}} \mathrm{B_j}\right|}{\left|\bigcup_{\mathrm{i}}^{\mathrm{I}} \mathrm{A_i}\right|} \tag{6}$$

Where A and B denote the masks of the GT and the segment, respectively. I denotes the number of GTs in the sample. J is the number of segments in the sample.

Moreover, we calculate the mean IoU (mIoU), mean Dice(mDice) and mean purity(mPurity) by averaging the maximum purity of all segments, and the mean entirety(mEntirety) by averaging the maximum entirety of all GTs. Mean LoGTs (mLoGTs) is calculated by computing the average region loss of the foreground for each image.

7. Experiments and analysis

To evaluate the generalization of the proposed method, we conduct assessments on both the FoodSeg103 dataset, which is a publicly available dataset specifically designed for ingredient segmentation, and the UEC-FoodPix Complete dataset, which is the most recognized dataset for food segmentation. There are other food databases, such as UNIMIB2016. Because our primary objective in this work is ingredient segmentation, we prioritize the evaluation on UEC-FoodPix Complete dataset. We will extend the evaluation of the generalization on other public datasets in the next work.

In this section, we evaluate the 1) performance of the single-ingredient classification model in single-ingredient identification. 2) Performance of the ingredient segmentation framework for multi-ingredient segmentation. 3) Performance of the Ingredient Segmentation framework for food segmentation.

7.1. Implementation Setups

The experiments were implemented on a computer with the following specifications: an Intel(R) Core i7-10870H CPU @ 2.20GHz and an NVIDIA GeForce RTX 3060 Laptop GPU, with 16 GB memory. The operating system used was Windows 11, and the codes were written in Matlab (2022a).During the training process, we trained all single-level single-ingredient classification models and all multi-level single-ingredient classification models on the SI110 training dataset. The single-level models were trained for 30 epochs, while the multi-level models were trained for 50 epochs. We used a mini-batch size of 32 and an initial learning rate of 3e-2. To facilitate learning, we implemented a piece learning rate schedule, where the learning rate was multiplied by 0.2 when decreased. Furthermore, we utilized the Adam optimizer with a squared gradient decay factor of 0.9.

To assess the effectiveness of our proposed method, we conducted evaluations in two stages. Firstly, we evaluated the performance of single-ingredient classification on the SI110 test dataset. Next, we evaluated the performance of ingredient segmentation on the FoodSeg103 datasets. FoodSeg103 contain 2135 test images with multiple ingredients in a food image with pixel-level annotations.

7.2. Evaluation on Single Ingredient Classification Model

In this section, we present a thorough evaluation of our proposed AttNets and two pretrained models: ResNet18 and EfficientNet-B0, for single-ingredient identification. To explore the effectiveness of different kernel sizes in the CNN blocks of AttNets, we designate four types of AttNets, all with the same architecture, but varying kernel sizes.

1. AttNet (1): uses convolutional layer with kernel size=1 in each CNN block
2. AttNet(1+3): uses a convolutional layer with kernel size=1 in each CNN block, except for the convolutional layer with kernel size=3 in the last CNN block.
3. AttNet(3+1): uses a convolutional layer with kernel size=3 in each CNN block, except for the convolutional layer with kernel size=1 in the last CNN block.
4. AttNet(3): uses convolutional layer with kernel size=3 in each CNN block

We employ four metrics to evaluate the performance of single ingredient classification: accuracy, precision, recall, and F1-score. The experimental results are presented in Table 2.

Table 2. Performance of SLMs and MLMs on SI110 dataset for single ingredient classification.

Types	Model	Accuracy	mPrecision	mRecall	mF1
SLM	AttNet(1)	0.2712	0.1933	0.1955	0.1693
	AttNet(1+3)	0.2716	0.1840	0.1946	0.1619
	AttNet(3+1)	0.2665	0.1846	0.1947	0.1623
	AttNet(3)	0.2553	0.1786	0.1805	0.1575
	EfficientNet-B0	0.8437	0.8161	0.8063	0.8017
	ResNet18	**0.8684**	**0.8498**	**0.8571**	**0.8466**
MLM	AttNet(1)	0.2326	0.1576	0.1544	0.1333
	AttNet(1+3)	0.2116	0.1787	0.1412	0.1285
	AttNet(3+1)	0.387	0.41	0.3254	0.324
	AttNet(3)	0.3205	0.3307	0.2686	0.2574
	EfficientNet-B0	0.8307	0.8290	0.8253	0.8168
	ResNet18	0.6940	0.6923	0.6603	0.6495

In terms of the performance of SLM, our observations reveal that the modified ResNet18 achieves the highest performance among the SLMs. It attains an accuracy of 0.8684, precision of 0.8498, recall of 0.8571, and an F1 score of 0.8466. Among the AttNet models, AttNet(1) demonstrates the top performance, with an accuracy of 0.2712, precision of 0.1933, recall of 0.1955, and an F1 score of 0.1693.

Regarding the performance of MLM, we found that the modified EfficientNet-B0 exhibits the best performance, with an accuracy of 0.8684, precision of 0.8307, recall of 0.8253, and an F1 score of 0.8168. Among the AttNet models, AttNet(3+1) achieves the highest performance, with an accuracy of 0.387, precision of 0.41, recall of 0.3254, and an F1 score of 0.324.

In comparing SLMs and MLMs, our experimental results clearly demonstrate that applying multilevel learning significantly improves the performance of the AttNet(3), AttNet(3+1), and EfficientNet-B0 models. However, it decreases the performance of the AttNet(1), AttNet(1+3), and ResNet18 models.

In the following subsection, we utilize all of the aforementioned models as the backbone of our proposed ingredient segmentation framework for multi-ingredient segmentation and assess their performance. This is because the performance of single ingredient classification does not clearly correlate with the performance of multi-ingredient segmentation.

7.3. Evaluation on Ingredient Segmentation

As our objective is to identify multiple ingredients through multiple ingredient segments in food images, we evaluate the effectiveness of ingredient segmentation using metrics such as mIoU, mDice, mPurity, mEntirety, and mLoGTs, which are relevant to ingredient classification.

We evaluate the performance of ingredient segmentation on the FoodSeg103 dataset [21], which consists of images containing multiple ingredients along with pixel-level ingredient labels. However, the majority of food images in this dataset also include non-food background areas. Since our proposed method aims to segment only the ingredients in food images while excluding the background, the presence of the background can potentially affect the ingredient segmentation results. To address this issue, we replace the background areas of these images with a blue background. In this process, each pixel in the background area is assigned an RGB value of (0, 0, 255), as this blue color is rarely encountered in food ingredients.

In this experiment, we employ mIoU, mDice, mPurity, mEntirety, and mLoGTs as evaluation metrics to assess the segmentation performance on this dataset. We also compare our mIoU with the one reported in [21]. However, it should be noted that the FoodSeg103 dataset comprises 103 ingredients, which is not entirely consistent with our defined 110 ingredient categories. As a result, direct comparison of the accuracy results presented in [21] may not be feasible.

7.3.1. Analysis of Method 1

We discuss the segmentation performance of various backbones using Method 1, and the results are listed in Table 3. Based on the Purity results, SLM-ResNet18 achieves the highest score of 0.7679, followed by MLM-ResNet18 with a score of 0.694. However, these models exhibit low performance in terms of the Entirety score. Since our segmentation results will be used for the subsequent ingredient classification task, we consider these models not to be suitable for the segmentation used as backbones.

Table 3. Performance of multiple ingredient segmentation using SLMs and MLMs on FoodSeg103 with Method 1.

Types	Model	mPurity	mEntirety	mLoGTs	mIoU	mDice
SLM	AttNet(1)	0.6712	0.7555	0.1751	0.6051	0.7376
	AttNet(1+3)	0.6845	0.7394	0.1874	0.6028	0.7354
	AttNet(3+1)	0.6838	0.7675	0.1621	0.6056	0.7371
	AttNet(3)	0.6831	0.7791	0.1525	0.6072	0.7384
	EfficientNet-B0	0.7679	0.3116	0.6592	0.3666	0.5141
	ResNet18	0.5828	0.7879	0.1925	0.531	0.6789
MLM	AttNet(1)	0.6144	0.8463	0.0959	0.59	0.7265
	AttNet(1+3)	**0.6036**	**0.8579**	**0.0917**	**0.5759**	**0.7157**
	AttNet(3+1)	0.6225	0.7997	0.1410	0.5832	0.7213

	AttNet(3)	0.6167	0.8409	0.1087	0.5742	0.7137
	EfficientNet-B0	0.694	0.395	0.5804	0.4043	0.5597
	ResNet18	0.6594	0.4960	0.4805	0.434	0.5894

Next, we examine the results of the models for the entity. The MLM-AttNet models achieve relatively high values of over 0.8. As for LoGTs, we observe that MLM-AttNet(1) and MLM-AttNet(1+3) achieve comparatively good results of less than 0.1. Regarding IoU, the SLM-AttNet models achieve relatively high values of over 0.6. Finally, for Dice, both SLM-AttNet models and MLM-AttNet models have values of more than 0.7.

Because both the accurate segmentation of ingredients and the preservation of their integrity are crucial for the subsequent recognition process, we consider that under Method 1, MLM-AttNet(1) and MLM-AttNet(1+3) are suitable backbones for ingredient segmentation.

7.3.2. Analysis of Method2

Here, we evaluate the performance of various models for ingredient segmentation using Method 2. The results are presented in Table 4.

Table 4. Performance of multiple ingredient segmentation using SLMs and MLMs on Foodseg103 with Method 2.

Types	Model	mPurity	mEntirety	mLoGTs	mIoU	mDice
SLM	AttNet(1)	**0.8339**	**0.8003**	**0.0552**	**0.6532**	**0.7665**
	AttNet(1+3)	0.8565	0.7391	0.1276	0.6185	0.7407
	AttNet(3+1)	0.8255	0.6618	0.1824	0.5548	0.6911
	AttNet(3)	0.8158	0.6599	0.2072	0.5749	0.7024
	EfficientNet-B0	0.7922	0.6094	0.2657	0.5392	0.6780
	ResNet18	0.7980	0.1534	0.828	0.1327	0.2121
MLM	AttNet(1)	0.8256	0.7900	0.0837	0.6540	0.7611
	AttNet(1+3)	0.8373	0.7495	0.1141	0.6199	0.7415
	AttNet(3+1)	0.7949	0.5865	0.2487	0.5001	0.6473
	AttNet(3)	0.8026	0.6055	0.2336	0.5090	0.6533
	EfficientNet-B0	0.7559	0.5342	0.2984	0.3759	0. 5160
	ResNet18	0.7314	0.5963	0.2364	0.3645	0.4965

It is important to note that the values of LoGTs are not equal to zero, despite the expectation that they should be zero according to the mechanism of Method 2. This discrepancy arises because we replace the background region pixels of the images in the FoodSeg103 dataset with pure blue prior to ingredient segmentation. Consequently, when applying k-means clustering for pixel clustering, we set the number of clusters to K+1, where K represents the number of ingredients in the image. This results in obtaining K+1 segments, including K ingredient segments and one background segment. However, in some cases, we observed that certain areas of ingredients were mistakenly segmented into the background segment. Therefore, the LoGTs are not equal to 0, as the feature vectors of pixels corresponding to some ingredients and those of pixels corresponding to the background are clustered into the same cluster using the K-means algorithm.

From Table 4, we observed that the AttNet models outperformed the ResNet-based and EfficientNet-based models across all metrics. Additionally, we noticed that various AttNet models under Method 2 exhibited superior segmentation performance compared to Method 1 in terms of Purity, LoGTs, IoU, and Dice metrics. Particularly, for Purity, the mPurity values of AttNet models under Method 2 improved by approximately 15% compared to those under Method 1. For IoU, the mIoU values showed an improvement of about 5%.

Regarding Entirety, SLM-AttNet(1) achieved the highest value of 0.80, although it was slightly lower than those of some MLM-AttNet models under Method 1. In terms of LoGTs, SLM-AttNet(1) attained the highest value of 0.055. As for IoU, which is a comprehensive metric for segmentation evaluation, SLM-AttNet(1) achieved a score of 0.6532, while MLM-AttNet(1) achieved a score of 0.6540. Both scores are nearly identical. Consequently, we believe that SLM-AttNet(1) under Method 2 serves as an optimal backbone for ingredient segmentation in terms of ingredient recognition.

Furthermore, compared to EfficientNet-B0, our SLM-AttNet(1) is more compact and requires less memory. SLM-AttNet(1) requires 22.113 MB of memory, whereas modified EfficientNet-B0 requires 70.506 MB. Moreover, as the backbone used in the ingredient segmentation framework, SLM-AttNet(1) requires less execution time than EfficientNet-B0 does. For instance, when taking an image of size 1024x1365 as input and obtaining all segmentation results using Method 2, SLM-AttNet(1) takes 1.71 seconds to execute, while EfficientNet-B0 takes 2.04 seconds. However, when using method 1, SLM-AttNet(1) takes 4.12 seconds to execute, whereas EfficientNet-B0 takes 3.59 seconds.

7.3.3. Comparison with previous work

In comparison to previous work [21], which achieves a class-wise mIoU of 0.439, our results demonstrate superior performance. Specifically, our MLM-AttNet(1) achieves the highest class-agnostic mIoU of 0.654 with Method 2, while the SLM-AttNet(1) achieves an almost same result of 0.6532 with Method 2. This indicates that our approach surpasses the previous work to some extent. Most importantly, our segmentation network uses a single-ingredient classification model as the backbone to generate masks for segmentation. This implies that we only need to train the ingredient classification model on a single-ingredient image dataset with image-level labels, thereby avoiding the need for pixel-level annotations.

7.3.4. Visualization of Segmentation results

To further investigate the performance of the ingredient segmentation framework, we present some examples of segmentation with different backbones on the MIF110 and FoodSeg103 datasets. We compare three backbones: SLM-AttNet(1), MLM-AttNet(1), and EfficientNet-B0, and compare the results using Methods 1 and 2. We chose to compare the AttNet model with the EfficientNet-based model because it performed better than ResNet18 on the entity and LoGTs metrics.

Regarding the comparison between SLM and MLM using AttNet(1), as shown in Figure 12 and Figure 13, Method 1 demonstrates a slightly better performance using MLM-AttNet(1) compared to SLM-AttNet(1). On the other hand, under Method 2, segmentation results indicate no significant difference between SLM and MLM.

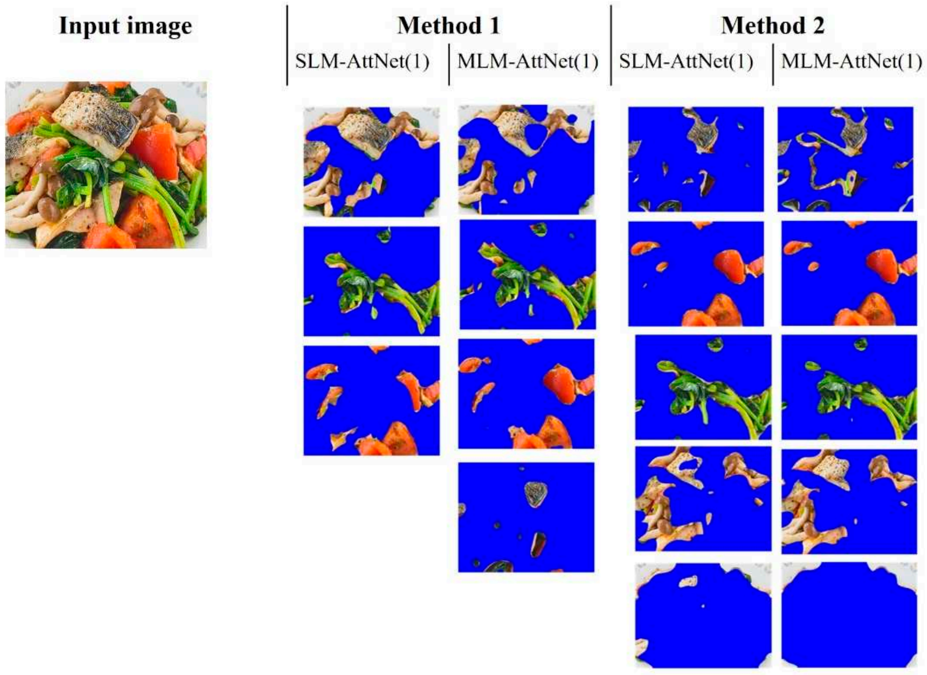

Figure 12. 4 ingredients in a dish image from MIF110 dataset. These results show that the use of Method 2 is significantly better than the use of Method 1.

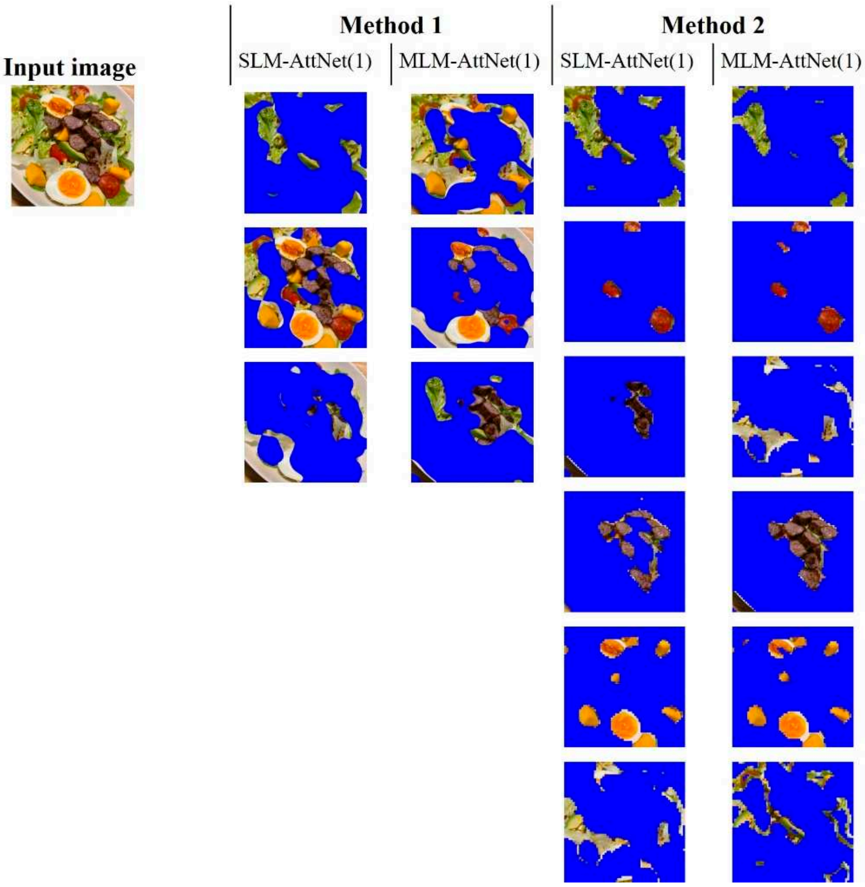

Figure 13. 6 ingredients in a dish image from MIF 110 dataset. These results show that Method 2 significantly outperforms Method1.

Regarding the comparison between AttNet(1) and EfficientNet-B0 , we compare the results from SLM-AttNet(1) and SLM-EfficientNet-B0. As shown in Figure 14-15, We found that the ingredient segmentation results generated by the AttNet(1) reserve more detailed boundaries than EfficientNet-B0 . Furthermore, we observe that the segmentation results obtained by EfficientNet-B0 suffer from missing parts of the ingredients. In Figure 14, we clearly observed this issue, where EfficientNet-B0 completely misses the parsley region. In Figure 15, under Method1, EfficientNet-B0 partially misses the sauce region, and under Method 2, EfficientNet-B0 further completely misses the apple region.

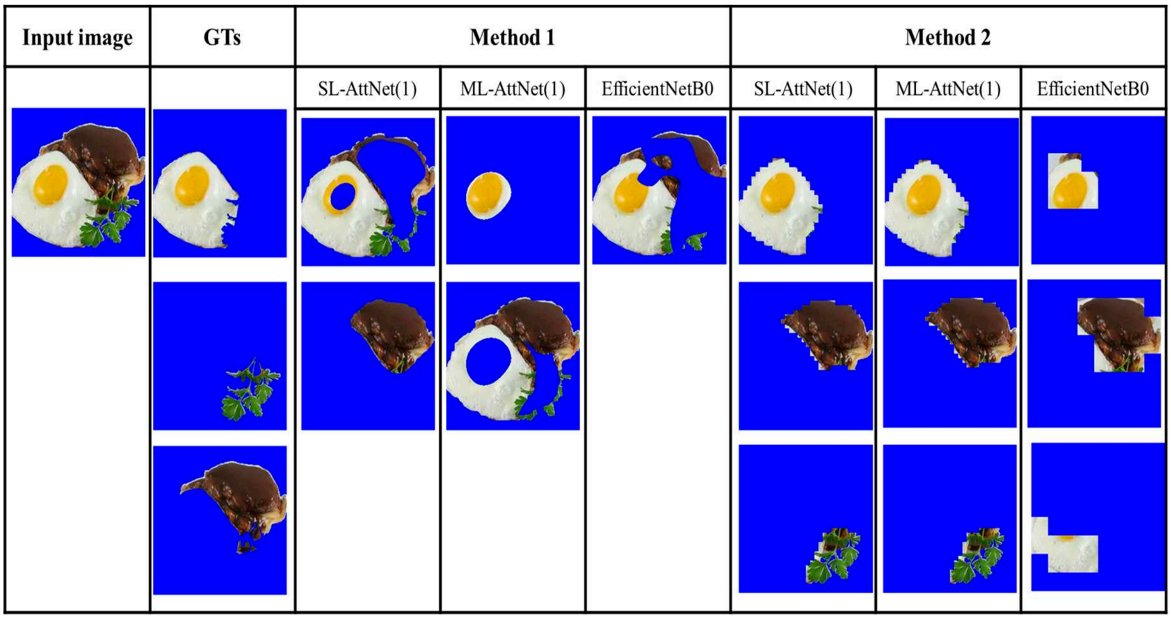

Figure 14. 3 ingredients (egg, parsley, steak) in a dish image from FoodSeg103 dataset.

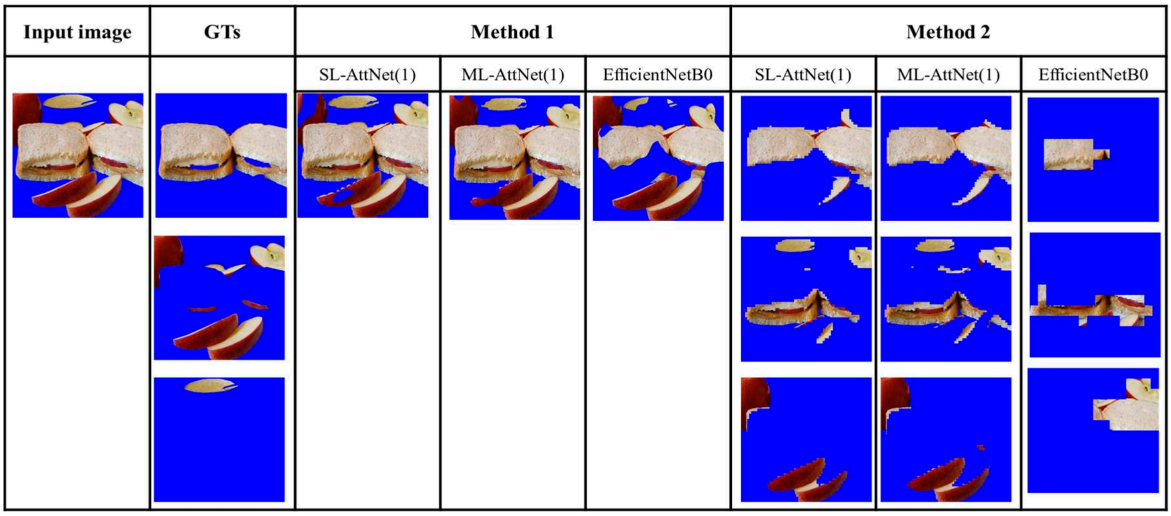

Figure 15. 3 ingredients (bread, apple, sauce) in a dish image from FoodSeg103 dataset. These results show that the proposed method has difficulty for segmenting ingredients having similar colors.

Regarding the comparison between Method 1 and Method 2, we conduct an analysis using a food image from the MIF110 dataset. In Figure 12 and Figure 13, it is evident that the segmentation results for AttNet(1) using Method 2 outperform those achieved with Method 1. As shown in Figure 12, under Method 1, the segmentation results group fish and mushrooms together into a single segment, whereas Method 2 successfully separates them to some extent. Specifically, as shown in Figure 13, models under Method 1 exhibit noticeable segmentation errors, where multiple ingredients are incorrectly segmented into the same segment.

Moving on to the analysis of segmentation results from the FoodSeg103 dataset, Figure 14 demonstrates that under Method 2, the egg is successfully segmented as a whole within the same segment. However, when employing Method 1, the segmentation of the egg is either separated or partially missed.

However, we found a drawback in using Method 2, where the eggs were segmented into different segments, as shown in Figure 13. Owing to their similar colors, the yellow parts of the eggs and mangoes were segmented into the same segment. The white part of the eggs and white plate in the background were segmented into another segment.

Finally, we compare the segmentation results of Method 2 with those of Segment Anything [24]. In our method, the SLM-AttNet (1) model is used for comparison. For Segment Anything, we use their published web demo to obtain the segmentation results by selecting the "segment everything" mode without adding any extra prompts. The segmentation results of the two multi-ingredient image examples are shown in Figure 16 and 17, respectively. We observed that both models segmented the ingredients well, and Segment Anything returned more accurate boundaries than did the proposed method. However, Segment Anything over-segmented the ingredients into several small pieces that are difficult to distinguish. In contrast, our method grouped the same ingredient into the same segment. As our objective is to identify the ingredients in the food images, it is necessary to segment them as completely as possible. Therefore, we argue that the segmentation results obtained using our method are more suitable in terms of ingredient recognition.

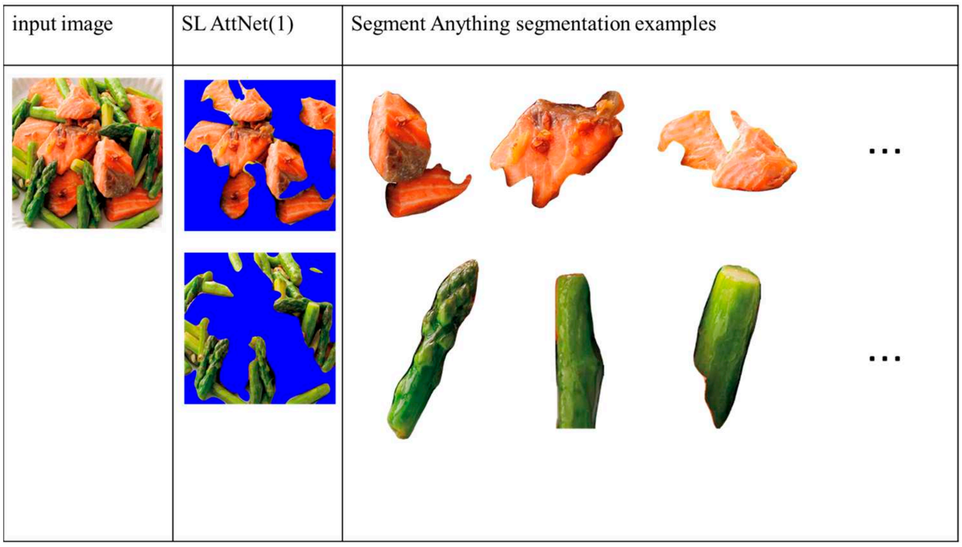

Figure 16. Comparison of segmentation results of our method with Segment Anything. Results show that Segment Anything model over-segments the ingredient into small pieces that are difficult to identify.

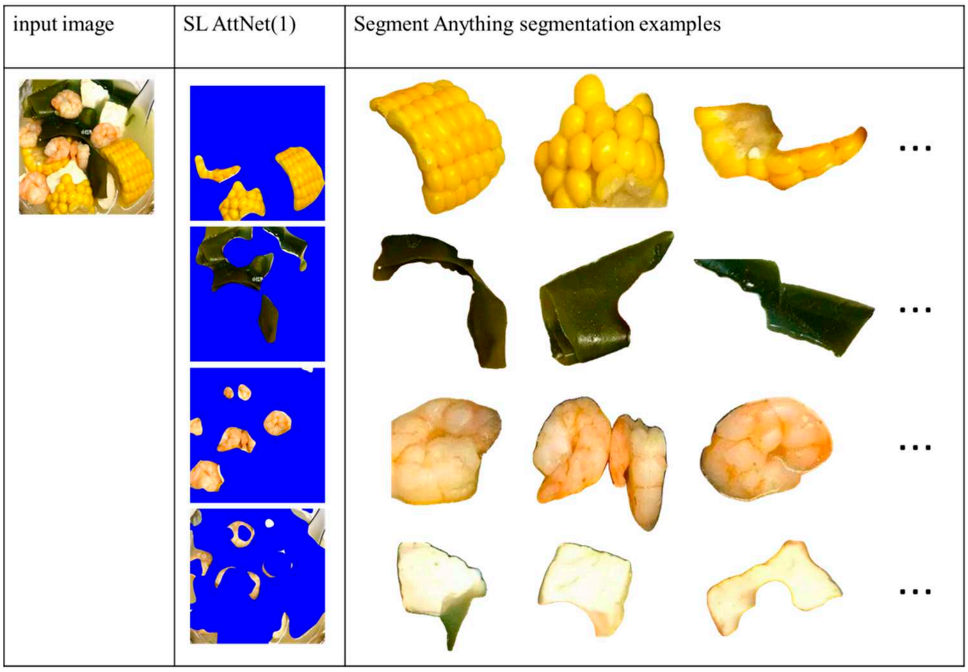

Figure 17. Comparison of segmentation results of our method with Segment Anything. Results show that Segment Anything model over-segments the ingredient into small pieces that are difficult to identify.

7.4. Evaluation on Food Segmentation

In this section, we evaluate the performance of the proposed ingredient segmentation framework for food segmentation by using the UECFoodPix COMPLETE dataset. We select SLM-AttNet (1) as the backbone and use method 2 for food segmentation. We calculate the mIoU and compare it with the results of previous studies such as UEC FoodPix [47], GourmetNet [19], and Deeplabv3+ [48]. Table 5 presents the results. Our ingredient segmentation framework performs better than UEC FoodPix but slightly worse than Deeplabv3+.

Table 5. Comparison of food segmentation performance with SOTA methods on UECFoodPix COMPLETE dataset.

Models	mIOU
UECFoodPix **Error! Reference source not found.**	55.55%
GourmetNet **Error! Reference source not found.**	65.13%
Deeplabv3+ **Error! Reference source not found.**	61.54%
Ours	60.18%

These results suggest that AttNet(1), a single-ingredient classification model, can be used as a backbone for partially addressing food segmentation. However, the proposed method is significantly inferior to GourmetNet. It appears that the single-ingredient classification model used in the ingredient segmentation framework lacks the ability to segment food. we plan to explore new methods to enhance the performance of food segmentation.

In summary, our qualitative and quantitative analyses indicate that the framework using SLM-AttNet(1) as the backbone and applying Method 2 to ingredient segmentation leads to better ingredient segments for the following ingredient recognition. However, we identify the drawbacks of the proposed approach:1) some different ingredients, or some parts of the ingredients and the background in the image may be segmented into the same segment, and 2) some parts of the background may be segmented into the ingredient segments because of their similar visual features.

9. Discussion

In this study, we introduce a hierarchical ingredient structure based on a standardized definition of ingredient categories. Using this structure, we constructed the SI110 and MIF110 datasets to train the single-ingredient classification model and evaluate the performance of multiple-ingredient segmentation in food images

Moreover, we proposed a novel CNN-based architecture with an attention mechanism for the single-ingredient classification model. In addition, we proposed an ingredient segmentation framework that utilizes a single-ingredient classification model as the backbone to extract feature maps and generate masks for ingredient segmentation. Furthermore, we explored two feature map processing methods to generate segmentation masks. This framework does not need the pixel annotations, providing a more practical and cost-effective solution for food ingredient segmentation. We assessed the segmentation performance using five metrics: IoU, Dice, Purity, Entirety, and LoGTs. Our findings indicated that employing SLM-AttNet (1) and applying Method 2 to the ingredient segmentation framework yields the best results.

In future work, our objective is to tackle the issues of different ingredients being segmented into the same segment and some parts of the background being segmented into ingredient segments.

doi:10.20944/preprints202308.1030.v1

Moreover, it is necessary to provide more results-based segmentation models using other matrices to assess the effectiveness of the proposed method.

Lastly, our proposed ingredient segmentation model lays the foundation and provides support for further advancements in ingredient recognition, nutritional assessment, and recipe recommendations.

Funding: This work was partly supported by JSPS KAKENHI Grant Numbers 22K12095.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

Acknowledgments: This work was partly supported by JSPS KAKENHI Grant Number JP22K12095. We would like to thank Kun Fu for his help with the data collection for constructing the dataset.

Appendix A

LIST OF 110 LEVEL 4 INGREDIENT CATEGORIES

Index	Ingredient Category	Index	Ingredient Category	Index	Ingredient Category	Index	Ingredient Category
C1	green caviar	C29	yam	C57	daikon	C85	other white flesh
C2	raspberries	C30	apple	C58	grape	C86	sweat potato
C3	peach	C31	green onion	C59	pineapple	C87	broccoli
C4	enoki	C32	orange	C60	asparagus	C88	broad beans
C5	avocado	C33	apricot	C61	cheese	C89	chinese chieves
C6	konpu	C34	wakame	C62	cabbage	C90	bamboo shoot
C7	sesame seeds	C35	crab	C63	cream	C91	celery stem
C8	eel	C36	corn	C64	octopus	C92	lotos
C9	yogurt	C37	green soybean	C65	peas	C93	peanuts
C10	papaya	C38	hazel nuts	C66	fig	C94	chickpea
C11	bonito	C39	bok choy	C67	kidney beans	C95	potato
C12	pitaya	C40	oyster	C68	sunflower seed	C96	pumpkin seeds
C13	chestnuts	C41	cauliflower	C69	lemon	C97	tomato
C14	pear	C42	cherry	C70	kiwi	C98	cattle
C15	purple laver	C43	lobster	C71	eggplant	C99	poultry
C16	salmon	C44	wax gourd	C72	grape fruits	C100	soybean
C17	banana	C45	almond	C73	mushroom	C101	onion
C18	snowpea	C46	blueberry	C74	celtuce	C102	oyster mushroom
C19	black rice	C47	lettuce	C75	meat product	C103	cucumber
C20	bean sprout	C48	tree ears	C76	abalone	C104	pepper
C21	wulnuts	C49	mackerels	C77	watermelon	C105	carrot
C22	tuna	C50	shimeiji	C78	kidney bean	C106	shiitake
C23	melon	C51	pumpkin	C79	okra	C107	strawberry
C24	bitter melon	C52	cashews	C80	chinese cabbage	C108	swine
C25	pistachio	C53	squids	C81	clam	C109	shrimp
C26	mantis shrimp	C54	mango	C82	egg	C110	wheat_product
C27	garlic stem	C55	millet	C83	pecan		
C28	rice	C56	spinach	C84	tofu		

References

1. Min, W., Jiang, S., Liu, L., Rui, Y., & Jain, R.C. (2018). A Survey on Food Computing. ACM Computing Surveys (CSUR), 52, 1 - 36.
2. Kagaya, H., Aizawa, K., & Ogawa, M. (2014, November). Food detection and recognition using convolutional neural network. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 1085-1088).
3. Aguilar, E., Bolaños, M., & Radeva, P. (2017). Food recognition using fusion of classifiers based on CNNs. In Image Analysis and Processing-ICIAP 2017: 19th International Conference, Catania, Italy, September 11-15, 2017, Proceedings, Part II 19 (pp. 213-224). Springer International Publishing.
4. Subhi, M.A., Ali, S.H., & Mohammed, M.A. (2019). Vision-Based Approaches for Automatic Food Recognition and Dietary Assessment: A Survey. IEEE Access, 7, 35370-35381.
5. Lo, F.P., Sun, Y., Qiu, J., & Lo, B.P. (2020). Image-Based Food Classification and Volume Estimation for Dietary Assessment: A Review. IEEE Journal of Biomedical and Health Informatics, 24, 1926-1939.
6. Martinel, N., Foresti, G. L., & Micheloni, C. (2018, March). Wide-slice residual networks for food recognition. In 2018 IEEE Winter Conference on applications of computer vision (WACV) (pp. 567-576). IEEE.
7. Zhou, F., & Lin, Y. (2016). Fine-grained image classification by exploring bipartite-graph labels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (pp. 1124-1133).
8. Min, W., Liu, L., Luo, Z., & Jiang, S. (2019, October). Ingredient-guided cascaded multi-attention network for food recognition. In Proceedings of the 27th ACM International Conference on Multimedia (pp. 1331-1339).
9. Qiu, J., Lo, F.P., Sun, Y., Wang, S., & Lo, B.P. (2019). Mining Discriminative Food Regions for Accurate Food Recognition. British Machine Vision Conference.
10. Bolaños, M., Ferrà, A., & Radeva, P. (2017). Food Ingredients Recognition Through Multi-label Learning. ArXiv, abs/1707.08816.
11. Gao, J., Chen, J., Fu, H., & Jiang, Y. (2022). Dynamic Mixup for Multi-Label Long-Tailed Food Ingredient Recognition. IEEE Transactions on Multimedia.
12. Chen, J., Zhu, B., Ngo, C., Chua, T., & Jiang, Y. (2020). A Study of Multi-task and Region-Wise Deep Learning for Food Ingredient Recognition. IEEE Transactions on Image Processing, 30, 1514-1526.
13. Xue, Y., Niu, K., & He, Z. (2021). Region-Level Attention Network for Food and Ingredient Joint Recognition. Proceedings of the 2021 4th International Conference on Algorithms, Computing and Artificial Intelligence.
14. Chen, J., Pan, L., Wei, Z., Wang, X., Ngo, C., & Chua, T. (2020). Zero-Shot Ingredient Recognition by Multi-Relational Graph Convolutional Network. AAAI Conference on Artificial Intelligence.
15. Romero-Tapiador, S., Tolosana, R., Morales, A., Espinosa-Salinas, I., Freixer, G., Fierrez, J., Vera-Rodríguez, R., Ortega-Garcia, J., Pau, E.C., & Molina, A.R. (2022). AI4Food-NutritionDB: Food Image Database, Nutrition Taxonomy, and Recognition Benchmark. ArXiv, abs/2211.07440.
16. 生鮮食品品質表示基準 (Standards for Fresh Food Quality Labeling), https://www.caa.go.jp/policies/policy/food_labeling/quality/quality_labelling_standard/pdf/kijun_01.pdf.
17. 新食品成分表 FOODS 2021 (New Food Ingredients List FOODS 2021), ISBN-13：9784809063718.
18. Aguilar, E., Remeseiro, B., Bolaños, M., & Radeva, P. (2017). Grab, Pay, and Eat: Semantic Food Detection for Smart Restaurants. IEEE Transactions on Multimedia, 20, 3266-3275
19. Sharma, U., Artacho, B., & Savakis, A. (2021). Gourmetnet: Food segmentation using multi-scale waterfall features with spatial and channel attention. Sensors, 21(22), 7504.
20. Okamoto, K., Adachi, K., & Yanai, K. (2021). Region-Based Food Calorie Estimation for Multiple-Dish Meals. Proceedings of the 13th International Workshop on Multimedia for Cooking and Eating Activities.
21. Wu, X., Fu, X., Liu, Y., Lim, E., Hoi, S.C., & Sun, Q. (2021). A Large-Scale Benchmark for Food Image Segmentation. Proceedings of the 29th ACM International Conference on Multimedia.
22. Xia, X., Liu, W., Wang, L., & Sun, J. (2023). HSIFoodIngr-64: A Dataset for Hyperspectral Food-Related Studies and a Benchmark Method on Food Ingredient Retrieval. IEEE Access, 11, 13152-13162.
23. Aslan, S., Ciocca, G., Mazzini, D., & Schettini, R. (2020). Benchmarking algorithms for food localization and semantic segmentation. International Journal of Machine Learning and Cybernetics, 11(12), 2827-2847.
24. Liang, Y., Li, J., Zhao, Q., Rao, W., Zhang, C., & Wang, C. (2022). Image Segmentation and Recognition for Multi-Class Chinese Food. 2022 IEEE International Conference on Image Processing (ICIP), 3938-3942.
25. Wang, Q., Dong, X., Wang, R., & Sun, H. (2022, June). Swin Transformer Based Pyramid Pooling Network for Food Segmentation. In 2022 IEEE 2nd International Conference on Software Engineering and Artificial Intelligence (SEAI) (pp. 64-68). IEEE.
26. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., & Girshick, R. (2023). Segment Anything. arXiv preprint arXiv:2304.02643.

27. Chen, J., & Ngo, C. (2016). Deep-based Ingredient Recognition for Cooking Recipe Retrieval. Proceedings of the 24th ACM international conference on Multimedia.
28. Min, W., Liu, L., Wang, Z., Luo, Z., Wei, X., Wei, X., & Jiang, S. (2020, October). Isia food-500: A dataset for large-scale food recognition via stacked global-local attention network. In Proceedings of the 28th ACM International Conference on Multimedia (pp. 393-401).
29. Myers et al., "Im2Calories: Towards an Automated Mobile Vision Food Diary," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1233-1241, doi: 10.1109/ICCV.2015.146.
30. Okamoto, K., & Yanai, K. (2020). UEC-FoodPix Complete: A Large-Scale Food Image Segmentation Dataset. ICPR Workshops.
31. Zhang, X., Lu, Y., & Zhang, S. (2016). Multi-Task Learning for Food Identification and Analysis with Deep Convolutional Neural Networks. Journal of Computer Science and Technology, 31, 489-500.
32. Crawshaw, M. (2020). Multi-task learning with deep neural networks: A survey. arXiv preprint arXiv:2009.09796.
33. Liang, H., Wen, G., Hu, Y., Luo, M., Yang, P., & Xu, Y. (2021). MVANet: Multi-Task Guided Multi-View Attention Network for Chinese Food Recognition. IEEE Transactions on Multimedia, 23, 3551-3561.
34. Dai, J., He, K., & Sun, J. (2016). Instance-Aware Semantic Segmentation via Multi-task Network Cascades. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3150-3158.
35. Kendall, Alex et al. “Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics.” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018): 7482-7491.
36. Li, X., Zhou, Y., Zhou, Y., & Wang, W. (2021, September). MMF: multi-task multi-structure fusion for hierarchical image classification. In International Conference on Artificial Neural Networks (pp. 61-73). Springer, Cham.
37. SSanh, Victor, Thomas Wolf, and Sebastian Ruder. "A hierarchical multi-task approach for learning embeddings from semantic tasks." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33. No. 01. 2019.
38. Dhanachandra, N., Manglem, K., & Chanu, Y. J. (2015). Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. Procedia Computer Science, 54, 764-771.
39. Zheng, X., Lei, Q., Yao, R., Gong, Y., & Yin, Q. (2018). Image segmentation based on adaptive K-means algorithm. EURASIP Journal on Image and Video Processing, 2018(1), 1-10.
40. Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep Clustering for Unsupervised Learning of Visual Features. European Conference on Computer Vision.
41. Van Gansbeke, W., Vandenhende, S., Georgoulis, S., & Van Gool, L. (2021). Unsupervised semantic segmentation by contrasting object mask proposals. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 10052-10062).
42. Z. Zhu and Y. Dai, "CNN-based visible ingredient segmentation in food images for food ingredient recognition," 2022 12th International Congress on Advanced Applied Informatics (IIAI-AAI), 2022, pp. 348-353, doi: 10.1109/IIAIAAI55812.2022.00077.
43. Tan, Mingxing and Quoc V. Le. “EfficientNetV2: Smaller Models and Faster Training.” ArXiv abs/2104.00298 (2021): n. pag.
44. Woo, Sanghyun et al. “CBAM: Convolutional Block Attention Module.” European Conference on Computer Vision (2018).
45. He, Kaiming et al. “Deep Residual Learning for Image Recognition.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 770-778.
46. Wang, Y., Liu, C., Zhu, F., Boushey, C. J., & Delp, E. J. (2016, September). Efficient superpixel based segmentation for food image analysis. In 2016 IEEE International Conference on Image Processing (ICIP) (pp. 2544-2548). IEEE.
47. Okamoto, K., & Yanai, K. (2021). UEC-FoodPIX Complete: A large-scale food image segmentation dataset. In Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part V (pp. 647-659). Springer International Publishing.
48. Marín, J., Biswas, A., Ofli, F., Hynes, N., Salvador, A., Aytar, Y., Weber, I., & Torralba, A. (2018). Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43, 187-203.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.