Lightweight Food Recognition via Aggregation Block and Feature Encoding

YANCUN YANG, School of Information and Electrical Engineering, Ludong University, China
WEIQING MIN, Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, China
JINGRU SONG, School of Information and Electrical Engineering, Ludong University, China
GUORUI SHENG, School of Information and Electrical Engineering, Ludong University, China
LILI WANG, School of Information and Electrical Engineering, Ludong University, China
SHUQIANG JIANG, Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, China

Food image recognition has recently been given considerable attention in the multimedia field in light of its possible implications on health. The characteristics of the dispersed distribution of ingredients in food images put forward higher requirements on the long-range information extraction ability of neural networks, leading to more complex and deeper models. Nevertheless, the lightweight version of food image recognition is essential for improved implementation on end devices and sustained server-side expansion. To address this issue, we present Aggregation Feature Net(AFNet), a lightweight network that is capable of effectively capturing both global and local features from food images. In AFNet, we develop a novel convolution based on a residual model by encoding global features through row-wise and column-wise information integration. Merging aggregation block with classic local convolution yields a framework that works as the backbone of the network. Based on the efficient use of parameters by the aggregation block, we constructed a lightweight food image recognition network with fewer layers and a smaller scale, assisted by a new type of activation function. Experimental results on four popular food recognition datasets demonstrate that our approach achieves state-of-the-art performance with higher accuracy and fewer FLOPs and parameters. For example, in comparison to the current state-of-the-art model of MobileViTv2, AFNet achieved 88.4% accuracy of the top-1 level on the ETHZ Food-101 dataset, with similar parameters and FLOPs but 1.4% more accuracy. The source code will be provided in supplementary materials.

CCS Concepts: • **Computing methodologies → Visual content-based indexing and retrieval**.

Additional Key Words and Phrases: Food Recognition, Lightweight, Aggregation Block, FLOPs

Authors' addresses: Yancun Yang, Harryyang@ldu.edu.cn, School of Information and Electrical Engineering, Ludong University, No.186 Hongqi Middle Road, Zhifu District, Yantai, China; Weiqing Min, weiqingmin@ict.ac.cn, Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, No.6 Kexueyuan South Road, Beijing, China; Jingru Song, ld_jr9912@163.com, School of Information and Electrical Engineering, Ludong University, No.186 Hongqi Middle Road, Zhifu District, Yantai, China; Guorui Sheng, School of Information and Electrical Engineering, Ludong University, No.186 Hongqi Middle Road, Zhifu District, Yantai, China, shengguorui@ldu.edu.cn; Lili Wang, txjiaoyanshi@163.com, School of Information and Electrical Engineering, Ludong University, No.186 Hongqi Middle Road, Zhifu District, Yantai, China; Shuqiang Jiang, sqjiang@ict.ac.cn, Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, No.6 Kexueyuan South Road, Beijing, China.

1551-6857/2023/1-ART1 $15.00
https://doi.org/10.1145/3466780

ACM Reference Format:
Yancun Yang, Weiqing Min, Jingru Song, Guorui Sheng, Lili Wang, and Shuqiang Jiang. 2023. Lightweight Food Recognition via Aggregation Block and Feature Encoding. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1, Article 1 (January 2023), 25 pages. https://doi.org/10.1145/3466780

1 INTRODUCTION

Food computing[33] has gained increased attention within multimedia and computer vision, due to its possible applications to diet, health, and the food industry[17, 36, 43, 44, 46]. For example, by determining the type, components or other characteristics of a meal, the nutritional value of a meal can be assessed, and the individual can determine his diet habits, thereby ensuring the health of the individual and the prevention of disease. The recognition of food images is integral to these application scenarios[38, 50, 57]. In the context of a food computing system whose ultimate aim is to aid people in managing their diet and health and facilitating their daily life, it is crucial to implement a system for the efficient identification of food images on end devices such as mobile phones. In addition, the large selection of foods and cooking techniques has led to a rapid expansion of images of food, which has raised the standard for the long-term expansion of image recognition on the server side. Lastly, the recognition of food images belongs to the more complex and fine-grained recognition[42], and light efforts in the field will serve as a useful reference for similar work on fine-grained classification. Nevertheless, prevalent techniques at present[18, 29] adopt deep learning-based solutions, necessitating a vast amount of parameters and a prolonged training and inference process. So the emphasis of this paper is on the lightweight of deep neural network models for food image recognition.

Fig. 1. Some samples from ETHZ Food-101[1] and Vireo Food-172[4]. Ingredients are scattered throughout the food image.

First, food image recognition possesses unique characteristics that distinguish it from general image classification tasks, which can be summarized in two main differences: (1) Image Features: In general image recognition, objects typically appear as relatively continuous color blocks with

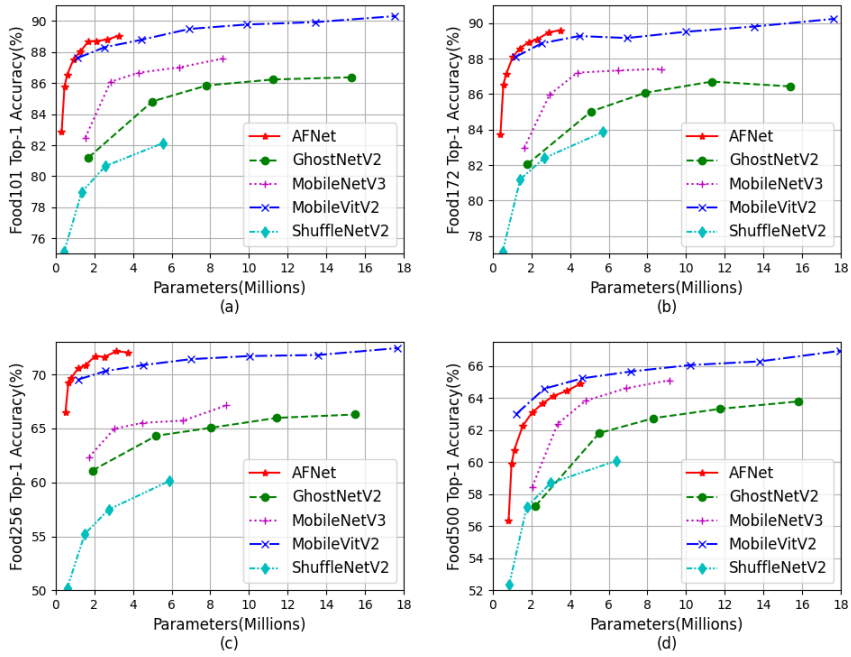


Fig. 2. Comparison in the parameter dimension with state-of-the-art CNN-based (MobileNetV3[12] & ShuffleNetV2[28] & GhostNetV2[53]) and Hybrid (MobileViTv2[31]) lightweight models across different datasets. (a): ETHZ Food-101[1]; (b): Vireo Food-172[4]; (c): UEC Food256[21]; (d): ISIA Food-500[35].

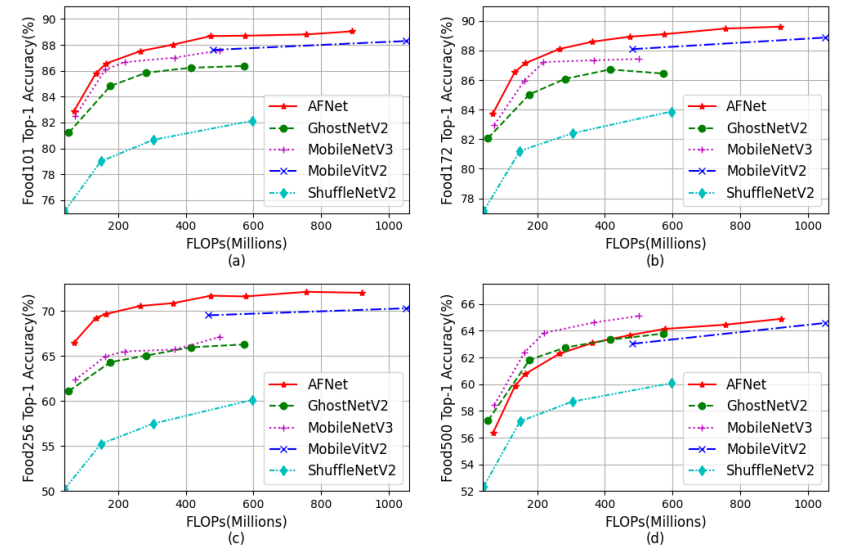


Fig. 3. Comparison in the FLOPs dimenstion with state-of-the-art CNN-based (MobileNetV3[12] & ShuffleNetV2[28] & GhostNetV2[53]) and Hybrid (MobileViTv2[31]) lightweight models across different datasets. (a): ETHZ Food-101[1]; (b): Vireo Food-172[4]; (c): UEC Food256[21]; (d): ISIA Food-500[35].

clear boundaries and extensive backgrounds. In contrast, food images often consist of multiple ingredients, with the same ingredient dispersed across different parts of the image, forming similarly colored blocks but with irregular shapes. The boundaries between different ingredients are intricate and convoluted, and the background is often minimal. Consequently, food image recognition is a complex fine-grained image recognition task. Analyzing the heatmaps, the recognition results for ordinary objects such as cats, dogs, or cups show clear focal points within the object's spatial location. However, in food image recognition heatmaps, the focal points often appear on specific key ingredients, displaying a discontinuous, multi-spot distribution. (2) Challenges in Image Recognition. The complexities of food image recognition surpass those of general image classification tasks in several ways. On one hand, when the image background is minimal and the number of categories to be recognized is relatively small, the high quality of food images makes the recognition task easier than typical image classification tasks. The interaction of multiple ingredients in food images does not significantly impact the difficulty of recognition. Our experimental results indicate that for datasets such as ETHZ Food-101[1] and Vireo Food-172[4], which feature high-quality images and fewer categories, lightweight models achieved high recognition accuracy, with some models reaching or even exceeding 90%. On the other hand, when the image background is extensive and the number of categories to be recognized increases, the intricate fine-grained features of food images exacerbate the recognition difficulty. The potential for confusion between categories rises substantially, given that different dishes may consist of similar ingredients. Experimental results show that for datasets such as UEC Food-256[21] and ISIA Food-500[35], the recognition accuracy is considerably lower than for ETHZ Food-101[1] and Vireo Food-172[4]. On the Food-256[21] dataset, the highest accuracy achieved by lightweight models was approximately 72%, while on the Food-500[35] dataset, the maximum accuracy was only around 66%.

To date, there has been limited research conducted on the topic of lightweight food image recognition. At the outset, a strategy involving a lightweight CNN was applied for the purpose of food image recognition[20, 22, 40, 52]. The key impediment is that conventional convolution models are unable to access extensive data from food images due to the scattered positioning of the ingredients, and obtaining further long-range characteristics necessitates a more intricate CNN model, thus limiting the potential for lightweight solutions. As Fig. 1 indicates, the essential factor in distinguishing between food images is the components that comprise the meal, with multiple ingredients often scattered throughout the picture. Additionally, the same component in the same dish can display distinct features in terms of size, form, and arrangement depending on the method of preparation, as is demonstrated by the two Takoyaki and Salad pictures within Fig. 1. As such, it is vital to accurately capture the far-reaching relationships between these scattered food ingredients to successfully identify a dish. Utilizing Vision Transformer (ViT)[7], extracting global details is achievable with the attentional mechanism driven by the input data. However, compared to convolutional operations, it suffers from the issue of high computational complexity. It mainly utilizes the Encoder part of Transformer, consisting of Attention and Feed Forward Network (FFN) components. The computational complexity mainly arises from the Attention part, where two matrix multiplications are required to compute the Attention matrix and the output value matrix. This non-linear complexity leads to a significant increase in computational cost as the model scales up. The goal of lightweight models is to drastically reduce parameter size and computational complexity while maintaining sufficient accuracy. Therefore, using Transformer for lightweight models poses a challenging task. Although some models like MobileVitV2[31] attempt to reduce computational costs by simplifying the operations in the Attention part, their computational complexity is usually significantly higher than that of pure CNN models. Sheng[48] endeavored to utilize the advantages of ViT's extensive global representation and CNN's potent

local representation capability. Nonetheless, the magnitude of the parameters and computations of the model is still appreciable.

Therefore, the challenges of lightweight food image recognition derive two-fold: (1) Owing to the disordered nature of the positioning of components, the long-range pixel correlation characteristics of food images are of great significance for food image recognition. Despite CNN's capacity for distinguishing local characteristics, a profoundly deep network must be constructed to represent the relationship between far-off pixel vectors. This will result in a marked augmentation of parameters and processing, which is antithetical to the need for a lightweight network. (2) ViT enables us to effectively obtain long-range pixel correlation features. Nonetheless, given the quadratic number of interactions between tokens, substantial vector dot product operations are necessary, as well as a greater volume of training data and additional cycles to secure the local correlations. Therefore, the processing ability is constrained and it is difficult to meet the lightweight requirements.

Our efforts have successfully resolved the primary issues related to lightweight food recognition, namely, the lack of long-range information expression capabilities of CNN and the complexity and difficulty of training the ViT model. We use aggregation block to capture the global information of food ingredients scattered in food images to obtain global expression and form an integrated block with local convolution. This integrated block is used as the basic structure of AFNet, which effectively improves the food image recognition accuracy. Furthermore, based on the efficient use of parameters by the aggregation block, we significantly decrease the number of network layers to lower the number of parameters and computations. We conduct comprehensive experiments on four significant databases in the food image field, the results show that compared with existing CNN-based, ViT-based, and hybrid lightweight networks, as shown in Fig. 2 and Fig. 3, in the case of equal or better recognition accuracy, our method has certain advantages in key metrics such as the number of parameters and FLOPs.

We summarize our contributions as follows:

- We designed a new type of neural network structure for food image recognition, called Aggregation Block, which implements a pure convolution block through information integration in the row direction and column direction, feature encoding, and residual model. The aggregation block can help effectively collect global information and feature positioning information.
- Based on the efficient use of parameters by the aggregation block, we constructed a lightweight food image recognition network AFNet with fewer layers and a smaller scale. Compared with similar lightweight image recognition models, it can achieve higher performance with fewer parameters and less computation.
- We designed a new activation function in the optimization of the AFNet network, called SoftReLU, which is a generalization of the Hardswish activation function. Experiments show that it can speed up the optimization of simple models and achieve high performance.

2 RELATED WORK

Our work is closely related to two research fields: (1) Lightweight CNNs, ViTs, and hybrid models, and (2) Lightweight food recognition.

2.1 Lightweight CNNs, ViTs, and Hybrid Models

ResNet [10] has been widely acknowledged as one of the top CNN architectures. Nevertheless, the most advantageous CNN models necessitate a considerable quantity of parameters and FLOPs. Lightweight CNNs that demonstrate competitive performance despite fewer parameters and FLOPs include ShuffleNetv2[28], ESPNetV2[32], EfficientNet[51], FasterNet[3], and MobileNetV2 [45] &

V3 [12]. MobileNetV3[12] is the most current iteration of a class of models constructed specifically for environments with limited resources, such as mobile devices. The basic blocks of MobileNetV3 [12] include MobileNetV2 [45] block and Squeeze-and-Excite network[15]. The frequent difficulty of CNN-based models with lightweight structures is their lack of capability to pick up on global information.

In order to acquire global information expeditiously, ViT [7] introduces transformer models for natural language processing assignments to the field of vision, particularly image recognition. The application of ViT in the realm of machine vision has prompted investigations into its capacity for efficiency. The majority of endeavors are focused on optimizing the self-attention process to raise efficiency, for instance SwinT [27], EfficientFormer [24], LightViT [16], EfficientViT [26], MiniViT [59] and TinyViT[56]. The primary challenges of ViT-based lightweight models are the complexity of training and the considerable computing expenditure due to the quadratic amount of interactions between tokens. Researchers have recently strived to develop compact hybrid models that integrate CNN and ViT for mobile vision tasks, evidencing that the combination of convolution and transformer yields augmentation in forecast accuracy as well as training solidity. Following, a considerable array of light-weight works on these models has been created, such as MobileFormer[5], CMT[8], CvT[55], BoTNet[49], Next-ViT[23], EdgeViTs [39], MobileViTv1[30] and MobileViTv2[31]. The hybrid lightweight model composed of CNN and ViT has been successful in amalgamating global and local data, yet the issue of a bulky model persists.

2.2 Lightweight Food Recognition

Recently, Min *et al.* [33] conducted an in-depth survey on the topic of food computing, which included food recognition. In the earlier years, various hand-crafted features are utilized for recognition [1, 58]. For example, Lukas *et al.* [1] utilized random forests to mine discriminative image patches as a visual representation. Owing to the advancement of deep learning technology, numerous recognition techniques founded on deep learning have arisen [11, 18, 19, 29, 34, 37, 50].

Given the necessity of lightweight food image recognition, a lot of related research work has been proposed. Early researchers use the light-weight CNN method for food image recognition[20, 22, 40, 52]. Tan *et al.*[52] recently propose a novel lightweight Neural Architecture Search (LNAS) model to self-generate a thin CNN that can be executed on mobile devices, achieving nearly 76% recognition accuracy on the ETHZ Food-101 dataset. The precision of these CNN-based lightweight food recognition models is not notably successful. ViT offers a novel approach to harvesting global characteristics of food images, Sheng *et al.* [48] tried to extract global and local features with a parallel structure composed of the ViT group and CNN, and obtained the SOTA performance. However, due to the multi-head attention mechanism of the ViT, the model size is still large. Sheng also tried to use a ViT-based lightweight food image recognition model [47], which got a high accuracy yet still suffers from high FLOPs.

In comparison to ViT-based approaches, we've developed a simplified yet efficient convolution network, utilizing the features of food images and allowing for improved regulation of parameters and computations. In this architecture, an aggregate block-based convolution is utilized to identify global features and a MobileNetV2 block is fashioned to draw out local features, resulting in SOTA performance.

3 METHOD

The overall network architecture is shown in Fig. 4, using a 3×256×256 image as input, an aggregation block together with a normal convolution as the leading head. The network's core consists of 5 layers with a collective of 9 aggregation blocks, each layer having the same resolution, except the first block's resolution which is halved. The tail uses a 1×1 convolution to expand the network to

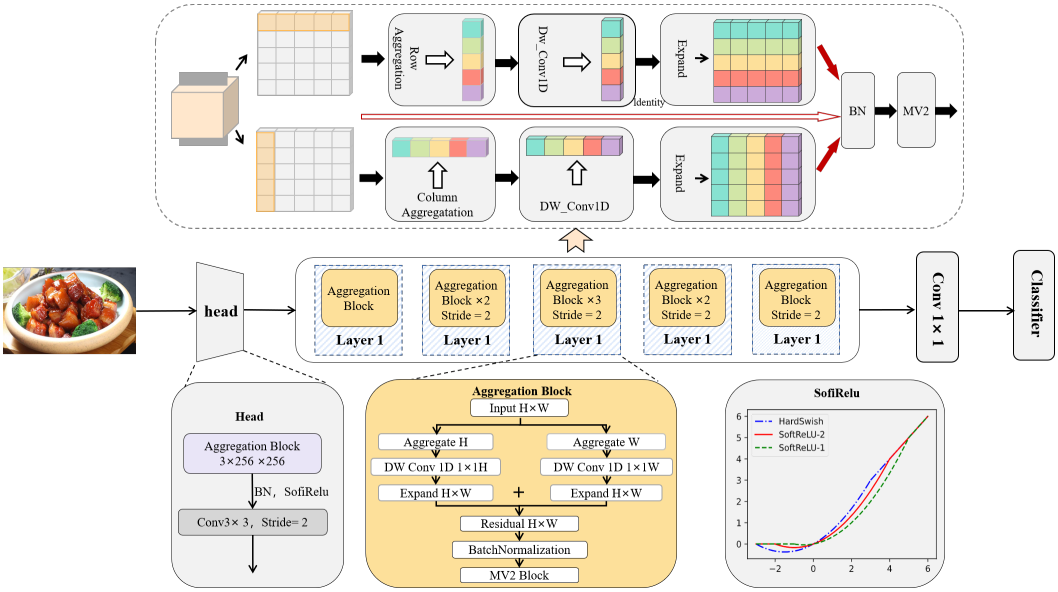


Fig. 4. Frame Work of AFNet. **Middle** is the backbone of AFNet. **Top** is the method of aggregation block. **Bottom Right** is the image of our proposed novel activation function SoftReLU.

the specified number of channels, and finally connects a classifier layer, which adjusts the graphics resolution to 1×1 through global average pooling, and then maps the channel to the classification through a linear model.

Since the main part uses the aggregation block, AFNet is much shallower than the general lightweight network, with only 9 blocks. The aggregation block constructs input data through the integration of row and column information, feature coding, and residual model, improves the utilization efficiency of parameters through data integration and maintains high performance while reducing the number of parameters and network scale.

3.1 Aggregation Block

As shown in the Fig. 4 and Table 1, the processing method of the aggregation block is to first integrate the information of the same row or the same column in the row direction and column direction respectively. Among the many optional ways of information integration, we use a relatively simple way of calculating the average value of pixels in each row or column. Through the integration of row/column information, the obtained vector contains the relevant information of the pixels of the full image. Use one-dimensional deep convolution for the obtained vector to realize learnable feature coding, and then broadcast the coding information to restore it to the size of the coded image, and then combine the feature coding in the row direction with the feature coding in the column direction. The original image matrix is added and then passed through a batch normalization layer, and finally used as the input of the MV2 block. The addition of the batch normalization layer here is based on the following considerations: since the encoding information is added to the input data, and this information is generated by the aggregation of the rows and columns, even if the original image matrix is normalized, the new input matrix will generally become denormalized, so adding a batch normalization layer is necessary. Generally, the neural network used for images will have a nonlinear activation layer after the batch normalization layer to generate a threshold-style

nonlinear output, which will obviously erase most of the added coding deviation information, so there is no nonlinearity in the aggregation block here.

The aggregation block collects and integrates global information in the row direction and the column direction, and then uses it for feature encoding. In doing so, the following three objectives are achieved:

- Collect global information. The food image is the same as the ordinary image in that it also has obvious aggregated color blocks, but the difference is also obvious. Since food is composed of various ingredients, different ingredients may be mixed and stirred during the food production process, and finally the same ingredient is distributed to different positions of the image. Therefore, it is important to collect information about food ingredients from a global perspective for food image recognition.
- Implement feature encoding. In the neural network used for food image recognition, the same ingredients may be located in different positions in different images, so using pixel position encoding on the original image cannot effectively deal with the characteristics of different distributions of the same ingredients. Therefore, it is more reasonable to encode the position of the extracted features in the middle layer of the network. Feature encoding in this way can reflect the positional relationship between different color blocks.
- Lightweight achieved through shallower neural networks. The existing image recognition neural network generally builds a deeper network by continuously reducing image resolution while increasing channels/features. Usually, to offset the computational load brought by the deep network, a smaller convolution kernel is generally selected. Therefore, only on the deeper network layer, the correlation information of long-distance related pixels on the original image (such as the same food material scattered throughout the image in the food image) can be captured. Aggregation block can effectively reduce the number of layers of the network, because it integrates the global information collected based on the row direction and column direction into the initial input, so that the neural network already has the correlation information between the distant pixels of the image at the shallow layer, instead of relying on the deepening of the number of network layers to obtain. Therefore, the aggregation block can achieve lightweight without loss of accuracy by greatly reducing the number of layers of the network.

Here we describe the calculation method and process of the aggregation block. Let the size of the input image be $h \times w \times c$, where h represents the height of the image, w represents the width, and c represents the number of channels, $M_{h\times w\times c}$ represents a matrix of vectors. let $e_l = (1, 1, \cdots 1)^T$, its elements are all 1, and its dimension is l. $E_{h\times w\times c}$ is a matrix with the same size as $M_{h\times w\times c}$ and all elements are 1, let $\cdot$ represent ordinary matrix multiplication, and $*$ represent dot multiplication. The calculation process of the encoding part of the aggregation block can be divided into 3 steps:

Row/Column Aggregation. The first step in the process is to compute the mean in the row/column direction. In the row direction, perform the following transformation $\frac{1}{w}M^{(k)}_{h\times w} \cdot e_w$, the result is a matrix of $h \times 1$. Stack all the transformed matrices to obtain the row-wise aggregation matrix $\frac{1}{w}M^{row}_{h\times c}$. Similarly, the operation in the column direction transforms $\frac{1}{h}e^T_h \cdot M^{(l)}_{h\times w}$, after stacking, the aggregation matrix $\frac{1}{h}M^{col}_{w\times c}$ can be obtained in the column direction.

Feature Encoding. Feature encoding is realized by one-dimensional deep convolution with a kernel size of 1, and in the row direction, continue to transform $\frac{1}{w}M^{row}_{h\times c} \cdot B^{row}_{h\times c}$, where $B^{row}_{h\times c}$ is the convolution parameter. Since one-dimensional depth convolution is used, there are only c parameters. In the column direction, continue with a similar transformation $\frac{1}{h}M^{col}_{w\times c} \cdot B^{col}_{w\times c}$. Where $B^{col}_{w\times c}$ is the convolution parameter. The two results here, the former is called feature encoding in the row direction, and the latter becomes feature encoding in the column direction.

Residual Combination. Expand the feature encoding in the row direction and column direction to the original image size through broadcasting, and add it to the original image matrix to obtain the output of the aggregation block encoding part, namely:

$$M_{h\times w\times c} + \left(\frac{1}{w}M^{row}_{h\times c} \cdot B^{row}_{h\times c}\right) * E_{h\times w\times c} + \left(\frac{1}{h}M^{col}_{w\times c} \cdot B^{col}_{w\times c}\right) * E_{h\times w\times c} \tag{1}$$

This output is used as input to the batch normalization layer.

From the implementation process of the aggregation block coding part, the following characteristics can be observed:

- Compared with the general image two-dimensional convolution block, the calculation of the aggregation block is less. The reason is that its parameters come from two one-dimensional deep convolutions. Since a 1×1 convolution kernel is used, the number of parameters is $2c$. According to the feature encoding mechanism described above, the calculation process is not affected by the size of the image. The amount of calculation is also mainly from one-dimensional convolution. Under the conditions of ignoring aggregation/mean operations and broadcasting operations, the number of addition and multiplication operations is $2c(h + w)$. Compared with ordinary two-dimensional convolution, the amount of calculation is very small.
- The residual model enables automatic learning of the encoding module. Taking a single-channel image as an example, the transformation formula at this time is:

$$M_{h\times w} + \frac{\beta^{row}}{w}M_{h\times w} \cdot e_w \cdot e_w^T + \frac{\beta^{col}}{h} \cdot e_h \cdot e_h^T \cdot M_{h\times w} \tag{2}$$

  The feature encoding in the row direction and column direction is controlled by parameters β^{row} and β^{col}, respectively. They are automatically learned through the network, which can automatically control the influence of the aggregation information in the row direction and column direction on their respective outputs. When the number of channels is greater than 1, the influence of multi-channel aggregation information on the output can be automatically coordinated.

3.2 SoftReLU Activation Function

Since the use of the aggregation block reduces the number of network parameters and network scale, a more aggressive activation function can be used in network optimization. We use the new activation function SoftReLU, which is actually an extension of the Hardswish activation function, defined in the following way :

$$Soft\,\mathrm{Re}\,LU = \begin{cases} 0 & x \le -\lambda \\ x & x \ge 6 - \lambda \\ x\,(x+\lambda)\,/6 & otherwise \end{cases} \tag{3}$$

The activation function contains a hyperparameter λ, which can be adjusted to allow a derivative similar to ReLU but greater than 1. On the interval $(-\lambda, 6-\lambda)$, the derivative function is $f'(x) = \frac{2x+\lambda}{6}$, in Hardswish, $\lambda = 3$, the variation range of the derivative is $\left(-\frac{1}{2}, \frac{3}{2}\right)$. when $\lambda = 2$, the variation range of the derivative is $\left(-\frac{1}{3}, \frac{5}{3}\right)$, when $\lambda = 1$, the variation range of the derivative is $\left(-\frac{1}{6}, \frac{11}{6}\right)$. When the derivative value is small, the exploration range of the model becomes smaller during the optimization process, and the convergence becomes slower. When the derivative value is large, more oscillations will occur during the model optimization process, making it difficult to converge. In the ablation experiment, we verified the effects of $\lambda = 3$, $\lambda = 2$, and $\lambda = 1$ on several

Table 1. Structure of AFB. C: number of channels; H: height of input/output; W: width of input/output; AOR: average of rows; AOC: average of columns;BEA: broadcast elementwise addition; EA: elementwise addition; Dwise3×3: depthwise convolution with kernel 3×3; EC: number of expanded channels.

Component	Input	Operator	#Out
Row/Column Aggregation	$C \times H \times W$	AOR	$C \times W$
	$C \times H \times W$	AOC	$C \times H$
	$C \times W$	Conv1d	$C \times W$
	$C \times H$	Conv1d	$C \times H$
	$C \times W \times W$	Unsqueeze(2)	$C \times 1 \times W$
	$C \times H \times W$	Unsqueeze(3)	$C \times H \times 1$
	$C \times 1 \times W$	BEA	$C \times H \times W$
	$C \times H \times 1$	BEA	$C \times H \times W$
	$C \times H \times W$(origin)	EA	$C \times H \times W$
	$C \times H \times W$(aggregated)	EA	$C \times H \times W$
	$C \times H \times W$	BatchNorm2d	$C \times H \times W$
MobileNetV2 Block	$C \times H \times W$	Conv1 × 1, ReLU6	$EC \times H \times W$
	$EC \times H \times W$	Dwise3 × 3, ReLU6	$EC \times H/s \times W/s$
	$EC \times H/s \times W/s$	Conv1 × 1, Linear	$C \times H/s \times W/s$
	$C \times H \times W(origin)$	EA(s=1)	$C \times H \times W$
	$C \times H/s \times W/s(reverted)$	EA(s=1)	$C \times H \times W$

typical databases, and found that $\lambda = 2$ obtained better results in most cases. Therefore, in various comparative experiments, we used the setting of $\lambda = 2$.

3.3 Network Specification

The detailed network specification is shown in Table 2. The network first obtains a 128×128 image plane through a 3×3 local convolution and then passes through a series of aggregation block groups. At the tail of the network, the number of channels is expanded by 1×1 convolution, then global pooling is performed to obtain the single-pixel output, and finally, a fully connected layer is used to map to the number of classes.

3.4 Discussion

The block structure design of AFNet primarily takes into account features such as the dispersed distribution of ingredients and fine-grain in food images. Through row-column integration, it gathers relevant information dispersed across the image plane for the same ingredients to enhance the network's representational capacity. Unlike food images where key features consist of dispersed patches of the same ingredients, ordinary images often exhibit larger backgrounds and stronger continuity in the color blocks of recognized objects. Considering AFNet's structure, it is expected to yield acceptable results on general image datasets but is anticipated to perform even better on food images. Extensive experiments conducted on four major food image datasets as well as common image datasets demonstrate that AFNet performs reasonably well on general image recognition tasks and exhibits outstanding performance on food image recognition tasks. Its performance surpasses all current mainstream lightweight recognition methods.

The second purpose of constructing an aggregation block is to use parameters more effectively to reduce computational costs. When the image resolution is large, using a fully connected network will bring a heavy parameter burden while making the optimization process more prone to oscillation or chaos. At the same time, the presentation form of various objects on the image is often some

aggregated color blocks, so the stacking network based on ordinary convolutional blocks is widely used in the image recognition network to reduce the number of parameters and calculation costs. In the main CNN-based lightweight model, the researchers further improved the convolution to reduce the amount of parameters and calculations, such as the depth convolution used in MobileNetV1[13], the bottleneck model used in MobileNetV2[45], and the low-cost channel expansion operation used in GhostNetV2[53] are all based on deep convolution and adopt different strategies to reduce the number of parameters and calculations. Aggregation block also uses deep convolution in the global information collection stage to reduce the amount of computation. However, unlike the previous operation directly on the original data and using two-dimensional deep convolution, AFNet operates on the integrated data in the row or column direction and uses 1D convolution. The parameters and computational complexity of one-dimensional convolution are linear, so the parameters and calculations added to the network model are very small.

The third purpose of using the aggregation block is to collect the global correlation information of the image. Although local correlation is the decisive factor in image recognition, the acquisition of global correlation information is still necessary. When there is no clear global information collection and processing mechanism, such as MobileNetV1[13] and MobileNetV2[45], the model realizes the reflection of global correlation through the overall full-level network structure. In almost all image recognition network hierarchies, features are extracted by continuously shrinking feature maps and increasing channels, so that the correlation of global information is generally reflected at the back of the network, so it is usually necessary to lay out deeper network levels. Therefore, some lightweight models are committed to reducing the network scale by adding a mechanism to collect global information in the backbone blocks of the network: for example, in MobileNetV3[12], the Squeeze Excitation block is introduced in some layers, which greatly reduces the number of parameters and calculations of the model; GhostNetV2[53] also introduces Squeeze Excitation blocks in some layers, and constructs an attention block based on average pooling in some layers to integrate global information. In related research on vision transformer, by introducing transformer blocks that are effective in natural language processing into image processing, the global correlation can be effectively expressed. However, the pure transformer model is not effective in image-related tasks, therefore most researchers combine it with convolutional blocks, but it is always difficult to solve the problem of the transformer's huge amount of computation.

The aggregation block introduces a non-transformer global information collection mechanism, also it is different from several previous mechanisms (such as Squeeze Excitation), the main difference lies in two points: (1) The previous mechanism including the transformer integrated the global information on the two-dimensional image, while the aggregation block integrated the information in the row direction and the column direction respectively, and finally formed not an overall global correlation description but the correlation on each row and column, which helps the network to use data more flexibly, without having to reorganize the information before using it as in the previous mechanism; (2) In the previous non-transformer method, when the global information and local information are fused, the activation function is generally used to construct a threshold mechanism to determine whether the global information is integrated. The aggregation block uses the residual model (does not use the activation function), and the integration of global information is reflected in the finally learned block parameters, this processing method can reduce the amount of calculation, and at the same time make the gradient change more robust, which is conducive to the use of more aggressive activation functions.

Another point to note is that the feature encoding of AFNet is different from the position encoding used in ViT. The use of positional encoding in the transformer is a very necessary mechanism in natural language processing. Position encoding usually uses two ideas, one is to construct random data to represent the position, and the other is to construct a set of position parameters for the

Table 2. Network specification. AFB: Aggregation Feature Block; Exp Ratio: Expansion Ratio in MobileNetV2[45] block.

Component	Input	Operator	Exp Ratio	#Out	Stride
Stem	$256 \times 256 \times 3$	AB-Single 3×3	-	-	-
	$256 \times 256 \times 3$	Conv2D 3×3	-	16	2
Layer 1	$128 \times 128 \times 16$	AFB 3×3	1	16	1
Layer 2	$128 \times 128 \times 16$	AFB 3×3	4	24	2
	$64 \times 64 \times 24$	AFB 3×3	3	24	1
Layer 3	$64 \times 64 \times 24$	AFB 3×3	3	40	2
	$32 \times 32 \times 40$	AFB 3×3	3	40	1
	$32 \times 32 \times 40$	AFB 3×3	3	40	1
Layer 4	$32 \times 32 \times 40$	AFB 3×3	6	80	2
	$16 \times 16 \times 80$	AFB 3×3	6	80	1
Layer 5	$16 \times 16 \times 80$	AFB 3×3	6	160	2
Head	$8 \times 8 \times 160$	Conv2D 1×1	-	1280	1
Classifier	$8 \times 8 \times 1280$	Global Avg Pool	-	1280	1
	$1 \times 1 \times 1280$	Linear	-	n classes	-

network to learn by itself. The aggregation block first integrates information in the row or column direction, and then through one-dimensional deep convolution, the network learns the form of integrating this integrated information into local information by itself, which we call this process feature encoding, that is, the network will learn a method that encodes the information collected on rows and columns into appropriate features for input to the network.

4 EXPERIMENT

4.1 Datasets

In order to assess the proposed model, we carry out experiments on four food datasets: ETHZ Food-101[1], Vireo Food-172[4], UEC Food-256[21] and ISIA Food-500[35]. ETHZ Food-101 has 101 categories, with 75,750 images used for training and 25,250 for validation. Vireo Food-172 has 172 Categories, 66,071 images are used for training, and 44,170 images are used to validate. UEC Food-256 contains 256 distinct classes, with 22,095 images allocated for training and 9,300 images allocated for validation. The ISIA Food-500 dataset comprises 500 types of food from Wikipedia, with 239,378 images employed for training and 120,142 images used for validation.

4.2 Training Settings

We train our models using an input image resolution 256×256, a batch size of 256, and SGD[2] optimizer with 0.9 momentum. We use the initial learning rate of 0.1 for the first 3,000 iterations of linear warm-up and then a cosine schedule with the learning rate ranging from 0.0004 to 0.8. Furthermore, we use the same data augmentation method as MobileViTv2[31] for image preprocessing.

The model weights were initialized using PyTorch's default parameter randomization method, with no pre-trained models utilized; all models were trained from scratch. The focus was on constructing an efficient foundational model for food image recognition, emphasizing the intrinsic capabilities of the models. In image recognition tasks, using parameters pre-trained on other datasets generally enhances model performance, and this is likely true in our work as well. However, even when pre-trained models are derived from the same dataset, variations in image augmentation

techniques and hyperparameter settings can result in different outcomes. This variability makes it challenging to conduct a fair comparison of direct model performance. In related work on lightweight image recognition, the primary concern is typically the comparison of independently trained model results.

4.3 Experiment Results

Results on ETHZ Food-101. Results from ETHZ Food-101 are displayed in Table 3. The results have been categorized based on a similar number of parameters. Our model is superior to all others in six parameter ranges. Among all models with less than 1M parameters, our model achieves 87.3% top-1 accuracy which is 13% higher than ShuffleNetV2[28]. Among all models with 1M-2M parameters, our model achieves 88.4% top-1 accuracy which is 7.2%/6.0%/1.4% higher than GhostNetV2[53], MobileNetV3[12], and MobileViTv2[31] respectively. In around 2-3M, 3-4M parameter budget models, our model's top-1 accuracy is 88.7% /88.8%/88.6%, which is at least 1 percentage point higher than the accuracy achieved by the current mainstream lightweight models like MobileViTv2[31], MobileNetV3[12], GhostNetV2[53] and EfficientNet[51]. Our model also achieves the highest top-1 accuracy of 88.4% in the parameter range of 3-5M, surpassing MobileViTv2[31], MobileNetV3[12] and MobileNetV2[12] by 0.8%, 2.2% and 1.9%, respectively. In the parameter size range of 5 to 10M, MobileViTv2[31] shows the best performance, but compared with our proposed AFNet, the performance is comparable but the parameter size is 6.6M higher, and the FLOPs are four times higher. We also compare with recent light-weight food recognition networks, the results show that the recognition accuracy of our network (86.4%) is much higher than that of LNAS-NET[52](75.9%) and LTBDNN(TD-192)[48](76.8%) in the case of much fewer parameters.

On the other hand, although AFNet surpasses other state-of-the-art lightweight models in terms of accuracy, it still falls short compared to our prior work, EHFR-Net [47]. AFNet's primary advantage lies in its lower computational cost. The smallest model, AFNet-0.5, achieves nearly 83% accuracy with just 0.3M parameters and 69M FLOPs. While AFNet-1.75 has a lower accuracy compared to EHFR-Net -0.75, it consumes half the computational resources. Similarly, the AFNet-2.0 model has an accuracy approximately 2% lower than that of EHFR-Net-1.0, yet the latter's computational cost is more than double that of the former. AFNet maintains control over parameter count and computational load, whereas EHFR-Net explores a wider range of parameters, resulting in a significant increase in FLOPs.

The design philosophy of the AFNet model differs from that of hybrid models based on Transformers, such as EHFR-Net [47] and MobileViT[31]. These latter models combine Transformers with convolutional networks, leveraging the capabilities of the Transformer Encoder to enhance accuracy. However, an unavoidable drawback is the generally high computational cost associated with Transformers. In contrast, AFNet employs a purely convolutional design, utilizing Aggregate Blocks to achieve lower computational cost with fewer parameters. This approach aims to create a lightweight model that balances accuracy and computational efficiency more effectively.

Results on Vireo Food-172. Table 4 presents results on VireoFood-172. Compared to MobileViTv2[31] in 0-1M, 1-2M, 2-3M, and 3-4M parameter range, our model achieves better top-1 accuracy of 88.0%, 88.9%, 89.4%, 89.5 with much fewer parameters. Our analysis concludes that the superior results on 172 are due to the data set comprising a larger selection of Chinese dishes, as well as its ingredients being more varied.

Results on UEC Food256. As seen in Table 5, the results are similar to the other two datasets. Our models achieve the highest top-1 accuracy in most parameter ranges. Compared to MobileViTv2[31], our model has fewer parameters and FLOPs. Compared to MobileNetV3 [12], ShuffleNetV2[28] and

Table 3. Performance comparison on ETHZ Food-101. AFNet-x: x denotes the width multiplier on the base model.

Method	Top-1 Acc.	#Params↑	#FLOPs
AFNet -0.5	82.9%	0.3M	69.1M
ShuffleNetV2 -0.5[28]	74.3%	0.5M	41.6M
AFNet -0.75	85.8%	0.5M	134M
AFNet -1.0	86.4%	0.6M	164M
AFNet -1.25	**87.3%**	0.9M	265M
MobileViTv2 -0.5[31]	87.0%	1.1M	480M
AFNet -1.5	87.8%	1.3M	363M
ShuffleNetV2 -1.0[28]	78.0%	1.4M	149M
MobileNetV3 -0.5[12]	82.4%	1.5M	73M
AFNet -1.75	**88.4%**	1.7M	474M
GhostNetV2 -0.5[53]	81.2%	1.7M	54M
EHFR-Net -0.75 [47]	**90.4%**	1.8M	**981.9M**
AFNet -2.0	88.3%	2.1M	576M
MobileViTv2-0.75[31]	87.2%	2.5M	1051M
ShuffleNetV2 -1.5[28]	80.3%	2.6M	304M
AFNet -2.25	**88.7%**	2.7M	756M
MobileNetV3 -0.75[12]	85.5%	2.8M	162M
EHFR-Net -1.0 [47]	**90.7%**	2.8M	**1238.5M**
AFNet -2.5	**88.8%**	3.3M	892M
MobileOne S1[54]	87.5%	3.7M	857M
MobileNetV3 -1.0[12]	86.2%	4.3M	219M
MobileViTv2-1.0[31]	87.6%	4.4M	1843M
EHFR-Net -1.25 [47]	**91.1%**	4.5M	**2104.5M**
EfficientNet B0[51]	85.2%	4.7M	567M
GhostNetV2 -1.0[53]	83.6%	5.0M	177M
ShuffleNetV2 -2.0[28]	82.0%	5.6M	596M
MobileOne S2[54]	88.0%	6.0M	1337M
MobileNetV3 -1.25[12]	86.2%	6.4M	367M
EHFR-Net -1.5 [47]	91.3%	6.4M	2985.5M
MobileViTv2 -1.25[31]	88.3%	6.9M	2856M
GhostNetV2 -1.3[53]	84.8%	7.8M	283M
MobileOne S3[54]	88.5%	8.3M	1942M
MobileNetV3 -1.5[12]	86.5%	8.6M	500M
MobileViTv2-1.5[31]	**88.6%**	9.9M	4089M
UniNet B0[25]	81.2%	10.3M	554M
UniNet B1[25]	84.1%	10.3M	1117M
GhostNetV2 -1.6[53]	85.5%	11.2M	415M
MobileOne S4[54]	89.9%	13.1M	3041M
MobileViTv2 -1.75[31]	88.9%	13.4M	5544M
GhostNetV2 -1.9[53]	85.7%	15.3M	573M
MobileViTv2-2.0[31]	**89.5%**	17.5M	7218M
LNAS-NET[52]	75.9%	1.8M	-
LTBDNN(TD-192)[48]	76.8%	12.2M	-
AFNet -1.0	**86.4%**	0.6M	164M

Table 4. Performance comparison on Vireo-172. AFNet-x: x denotes the width multiplier on the base model.

Method	Top-1 Acc.	#Params↑	#FLOPs
AFNet -0.5	83.7%	0.4M	69M
ShuffleNetV2 -0.5[28]	77.0%	0.5M	42M
AFNet -0.75	86.5%	0.6M	134M
AFNet -1.0	87.1%	0.7M	165M
AFNet -1.25	**88.0%**	1.0M	265M
MobileViTv2 -0.5[31]	87.3%	1.2M	480M
AFNet -1.5	87.8%	1.3M	363M
ShuffleNetV2 -1.0[28]	81.0%	1.4M	149M
MobileNetV3 -0.5[12]	83.0%	1.6M	73M
GhostNetV2 -0.5[53]	81.8%	1.8M	54M
AFNet -1.75	**88.9%**	1.9M	474M
AFNet -2.0	89.0%	2.3M	576M
MobileViTv2-0.75[31]	88.0%	2.5M	1051M
ShuffleNetV2 -1.5[28]	82.4%	2.7M	304M
AFNet -2.25	**89.4%**	2.9M	756M
MobileNetV3 -0.75[12]	85.9%	2.9M	162M
AFNet -2.5	**89.5%**	3.5M	919M
MobileOne S1[54]	88.2%	3.8M	857M
MobileNetV3 -1.0[12]	86.7%	4.4M	219M
MobileViTv2-1.0[31]	88.2%	4.5M	1843M
EfficientNet B0[51]	83.6%	4.8M	567M
GhostNetV2 -1.0[53]	84.7%	5.1M	177M
ShuffleNetV2 -2.0[28]	83.8%	5.7M	597M
MobileOne S2[54]	88.3%	6.2M	1337M
MobileNetV3 -1.25[12]	86.9%	6.5M	367M
MobileViTv2 -1.25[31]	87.9%	6.9M	2856M
GhostNetV2 -1.3[53]	85.7%	7.9M	283M
MobileOne S3[54]	88.9%	8.5M	1942M
MobileNetV3 -1.5[12]	86.5%	8.7M	500M
MobileViTv2-1.5[31]	**88.6%**	10.0M	4089M
UniNet B0[25]	82.7%	10.4M	554M
UniNet B1[25]	84.8%	10.4M	1117M
GhostNetV2 -1.6[53]	86.2%	11.3M	415M
MobileOne S4[54]	90.1%	13.3M	3041M
MobileViTv2 -1.75[31]	89.1%	13.5M	5544M
GhostNetV2 -1.9[53]	86.0%	15.4M	573M
MobileViTv2-2.0[31]	**89.4%**	17.6M	7218M

EfficientNet B0[51], our model achieves much higher top-1 accuracy with fewer parameters but slightly more FLOPs.

Results on ISIA Food-500. Table 6 presents experimental results on dataset ISIA Food-500. Because of its wide range, large scale, and offering of both Chinese and Western food, it is harder for food recognition in Food-500. Even so, our proposed AFNet still achieves competitive results: compared with SOTA ViT-based lightweight network MobileViTv2, the FLOPs are greatly reduced with almost the same recognition rate. Compared to the SOTA CNN-based light-weight network

Table 5. Performance comparison on UEC Food-256. AFNet-x: x denotes the width multiplier on the base model.

Method	Top-1 Acc.	#Params↑	#FLOPs
AFNet -0.5	65.5%	0.5M	69M
ShuffleNetV2 -0.5[28]	50.2%	0.6M	42M
AFNet -0.75	68.7%	0.7M	134M
AFNet -1.0	**69.3%**	0.8M	165M
AFNet -1.25	70.0%	1.2M	266M
MobileViTv2 -0.5[31]	69.1%	1.2M	466M
AFNet -1.5	70.4%	1.6M	363M
ShuffleNetV2 -1.0[28]	55.2%	1.5M	149M
MobileNetV3 -0.5[12]	62.1%	1.7M	74M
GhostNetV2 -0.5[53]	61.1%	1.9M	54M
AFNet -1.75	**71.4%**	2.1M	474M
AFNet -2.0	**70.7%**	2.5M	576M
MobileViTv2-0.75[31]	69.8%	2.6M	1052M
ShuffleNetV2 -1.5[28]	57.5%	2.7M	304M
MobileNetV3 -0.75[12]	64.9%	3.0M	162M
AFNet -2.25	71.8%	3.1M	756M
AFNet -2.5	**71.8%**	3.8M	919M
MobileOne S1[54]	68.1%	3.9M	857M
MobileNetV3 -1.0[12]	65.5%	4.5M	219M
MobileViTv2-1.0[31]	70.0%	4.5M	1843M
EfficientNet B0[51]	64.0%	4.9M	567M
GhostNetV2 -1.0[53]	63.9%	5.2M	177M
ShuffleNetV2 -2.0[28]	60.1%	5.9M	597M
MobileOne S2[54]	68.3%	6.4M	1337M
MobileNetV3 -1.25[12]	65.7%	6.6M	367M
MobileViTv2 -1.25[31]	71.2%	7.0M	2856M
GhostNetV2 -1.3[53]	65.0%	8.0M	283M
MobileOne S3[54]	69.2%	8.7M	1942M
MobileNetV3 -1.5[12]	67.1%	8.8M	501M
MobileViTv2-1.5[31]	**71.2%**	10.0M	4090M
UniNet B0[25]	58.7%	10.5M	554M
UniNet B1[25]	61.5%	10.5M	1117M
GhostNetV2 -1.6[53]	65.5%	11.4M	415M
MobileOne S4[54]	71.7%	13.4M	3041M
MobileViTv2 -1.75[31]	71.4%	13.6M	5544M
GhostNetV2 -1.9[53]	66.1%	15.5M	573M
MobileViTv2-2.0[31]	**71.5%**	17.7M	7219M

MobileNetV3[12], our model has significantly better performance with similar parameters: AFNet-2.5 obtain 63.7% top-1 accuracy, which is +3.2% higher than that of MobileNetV3[12](60.5%) with a similar amount of parameters.

Our experimental results confirm that for food recognition tasks, hybrid structure networks incorporating global information collection components outperform pure CNN networks in achieving higher performance. Our AFNet model was compared with several state-of-the-art (SOTA) pure

Table 6. Performance comparison on ISIA Food-500[35].

Method	Top-1 Acc.	#Params↑	#FLOPs
AFNet -0.5	56.3%	0.8M	70M
ShuffleNetV2 -0.5[28]	51.7%	0.9M	42M
AFNet -0.75	**59.9%**	1.0M	135M
AFNet -1.0	60.7%	1.1M	165M
MobileViTv2 -0.5[31]	**61.9%**	1.2M	480M
AFNet -1.25	61.6%	1.6M	266M
AFNet -1.5	62.6%	2.0M	364M
MobileNetV3 -0.5[12]	58.5%	2.1M	364M
GhostNetV2 -0.5[53]	56.9%	2.2M	55M
AFNet -1.75	62.6%	2.6M	475M
MobileViTv2-0.75[31]	**62.2%**	2.7M	1052M
ShuffleNetV2 -1.5[28]	56.2%	3.0M	304M
AFNet -2.0	63.2%	3.2M	577M
MobileNetV3 -0.75[12]	60.5%	3.4M	162M
AFNet -2.25	63.4%	3.8M	757M
AFNet -2.5	**63.7%**	4.5M	920M
MobileViTv2-1.0[31]	63.0%	4.6M	1844M
MobileNetV3 -1.0[12]	63.3%	4.8M	219M
EfficientNet B0[51]	60.1%	5.2M	567M
GhostNetV2 -1.0[53]	60.5%	5.5M	177M
ShuffleNetV2 -2.0[28]	59.0%	6.4M	597M
MobileNetV3 -1.25[12]	62.5%	6.9M	367M
MobileViTv2 -1.25[31]	**63.3%**	7.2M	2856M
GhostNetV2 -1.3[53]	61.0%	8.3M	283M
MobileNetV3 -1.5[12]	62.9%	9.1M	501M
MobileViTv2-1.5[31]	63.6%	10.2M	4090M
GhostNetV2 -1.6[53]	61.3%	11.8M	416M
MobileViTv2 -1.75[31]	61.9%	13.8M	5544M
GhostNetV2 -1.9[53]	61.9%	15.8M	573M
MobileViTv2-2.0[31]	**63.6%**	18.0M	7219M

CNN lightweight models, and the experimental results corroborate the limitations of pure CNN networks.

Food images differ from ordinary images mainly in several aspects: (1) Food images possess fine-grained features, with the same ingredient typically scattered throughout the entire image, composed of multiple separate color blocks, whereas ordinary images such as cats, dogs, or cups usually consist of continuous color blocks on the image plane; (2) In existing food datasets, dishes are typically centered in the image and occupy most of the image plane with a small background, whereas in ordinary image datasets, target objects may be located anywhere in the image and have a larger background. Therefore, in food image recognition, the influence of image background is smaller; (3) Foods are often composed of multiple ingredients, and during recognition, factors such as the proportion, area, and irregular shape of different ingredients influence the recognition process. The color mixing and shape composition in food images are more diverse, whereas in ordinary images, different target objects often have a certain distance or clearer boundaries.

These characteristics of food images make the collection of global correlation information particularly important for food image recognition tasks. Pure CNN models require deeper networks to

Table 7. Performance comparison on ImageNet-1K[6] with Comparable Parameter Ranges.

Method	Top-1 Acc.	#Params	#FLOPs
MobileNetV1[14]	70.6%	4.2M	575M
ShuffleNet -2.0[60]	73.7%	5.4M	524M
NasNet-A[41]	74.0%	5.3M	564M
MobileNetV3 Large -1.0[12]	75.2%	5.4M	219M
GhostNet -1.0[9]	73.9%	5.2M	141M
GhostNetV2 -1.0[53]	75.3%	6.1M	167M
ShuffleNetV2 -2.0[28]	74.5%	5.5M	557M
EfficientNet B0[51]	77.1%	5.3M	390M
MobileViT S[30]	78.4%	5.6M	2000M
MobileViTv2 -1.0[31]	78.1%	4.9M	1800M
MobileOne S1[54]	75.9%	4.8M	825M
MobileFormer-96M[5]	72.8%	4.6M	96M
AFNet -2.0	**74.5%**	4.4M	578M

Table 8. Performance comparison on ImageNet-1K[6] with Comparable FLOPs Ranges.

Method	Top-1 Acc.	#Params	#FLOPs
MobileNetV2 -1.4[45]	74.7%	6.9M	585M
GhostNetV2 -1.6[53]	77.8%	12.3M	399M
ShuffleNetV2 -2.0[28]	74.5%	5.5M	557M
MobileViT XXS[30]	69.0%	1.3M	400M
MobileViTv2 -0.5[31]	70.2%	1.4M	500M
MobileFormer-508M[5]	79.3%	14.0M	508M
AFNet -2.0	**74.5%**	4.4M	578M

better extract the correlation between distant pixels on the image plane, while networks incorporating global information collection components do not. However, very deep CNN networks can lead to the inability to achieve lightweight goals. From our experimental results, it can be seen that the performance of AFNet, MobileViTv2, and MobileNetV3 is significantly higher than that of pure CNN networks. Considering both parameter count and computational complexity, AFNet achieves the best performance.

Results on ImageNet-1K. From Table 7, it can be observed that AFNet outperforms most comparison models, achieving higher performance with similar parameter counts. The MobileVit[30] model exhibits notably higher accuracy than other models, but its FLOPs are several times higher than those of other models. AFNet's performance is comparable to that of MobileNetV3[12] and MobileOne[54], with MobileNetV3[12] achieving higher accuracy with significantly fewer FLOPs, albeit at the expense of higher parameter count compared to AFNet. MobileOne[54] achieves significantly higher accuracy than AFNet, but its FLOPs are also higher.

Table 8 lists several typical models with FLOPs similar to AFNet 2.0. In terms of accuracy, AFNet performs moderately well, significantly outperforming MobileNetV2[45] and ShuffleNetV2[28] while having substantially fewer parameters at similar levels of accuracy and FLOPs. It achieves lower accuracy compared to MobileFormer[5] and GhostNetV2[53], but the parameter count of the latter two models is 2-3 times higher than that of AFNet.

Overall, AFNet demonstrates competitive performance on general image recognition datasets while better balancing parameter count and FLOPs. However, due to its block structure designed for food image features, experiments on food datasets reveal that AFNet performs exceptionally well, outperforming all other lightweight models.

4.4 Qualitative Analysis

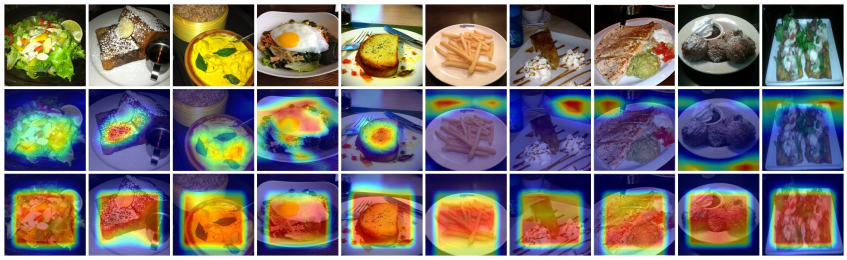
Different from the image recognition mechanism of the traditional local convolution, the network including the aggregation block tends to collect similar color patch information globally in the image plane. Fig. 5 shows the comparison by the method provided by Grad-CAM: results are obtained using and without using aggregation block. In Fig. 5, the first row is the original image, the second row is heat maps generated without using an aggregation block, and the third row is heat maps generated by using an aggregation block. As can be seen from Fig. 5: (1) Using only local convolution tends to identify locally clustered patches, which can be well-focused when they appear in food images; When the background is relatively monotonous and contains similar color blocks, the local convolution will also focus on the background incorrectly and cause recognition failure. (2) Convolution with aggregation block tends to collect similar color patches globally, and its focal area tends to be wider than local convolution, covering multiple color patches at the same time. (3) As shown in the last five columns of each figure, in some cases, the convolution without aggregation block fails to recognize because it focuses on the background, while the mechanism with aggregation block can correctly lock the target area for correct recognition. In summary above results show that the aggregation block is more suitable to the scattered-color features of food images and can achieve better recognition results.

It is readily apparent from Fig. 5 that a phenomenon is observable: the visual effect of a square shape is present in the third row of Fig. 5(a), (c), and (d), while absent in Fig. 5(b). Here, we provide an analysis and explanation of this phenomenon: The Aggregate Block collects related information of homogeneous color blocks dispersed in different regions of the image through row-column integration. If there are many blocks of the same color in the same row or column, stronger related information will be generated in that row or column. This information is then attached to the image plane through broadcasting mechanism, making it easy to identify rectangular or bar-shaped focus areas in the heatmap, as illustrated in Fig. 5(a), (c), and (d). However, when the background influence on food images is minimal and the dishes occupy the main position in the image, or when the distribution of food color blocks is more concentrated, AFNet's heatmap can still form irregularly shaped high-quality focuses. This is likely due to the use of residual models in the design of our network blocks, which automatically learn the proportion of local information and global-related information. Fig. 5(b) presents the experimental results on the Food172 dataset, which is known for its high image quality, where many images satisfy the characteristics mentioned above, thus, no rectangular visual effects are observed. To demonstrate that AFNet can indeed generate high-quality focus, we included some instances of this situation in Fig. 5(b). In fact, both scenarios exist in the heatmap of the Food172 dataset. Fig. 5b depicts the heatmap from the original paper (which can achieve irregular focus), while Fig. 6 shows heatmaps created from a different set of instances (rectangular focus blocks), exhibiting a similar rectangular appearance as in Fig. 5(a), (c), and (d).

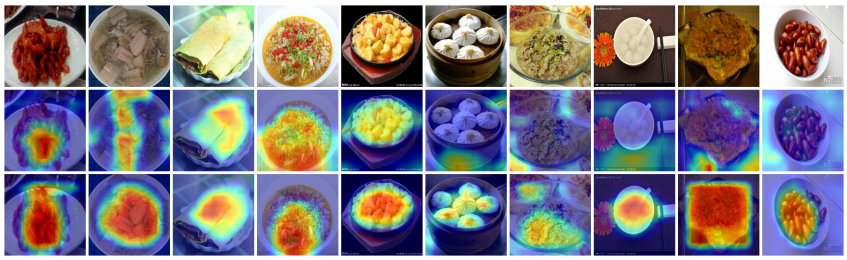
4.5 Ablation Study

In this section, we ablate important design elements in the proposed model using image classifications on four datasets.

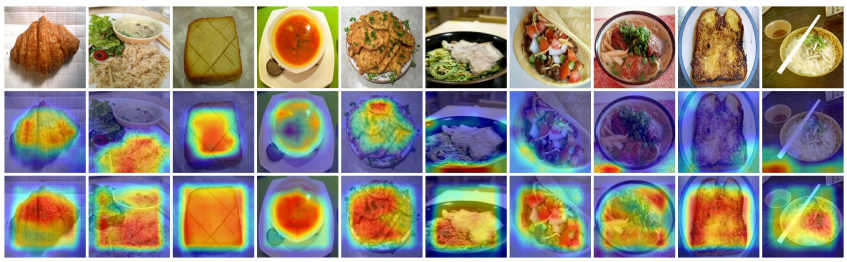
Effectiveness of Aggregation Block. Ablations of the Aggregation Block effect on four datasets are reported in Table 9. The models with aggregation blocks obtains more higher top-1 accuracy:



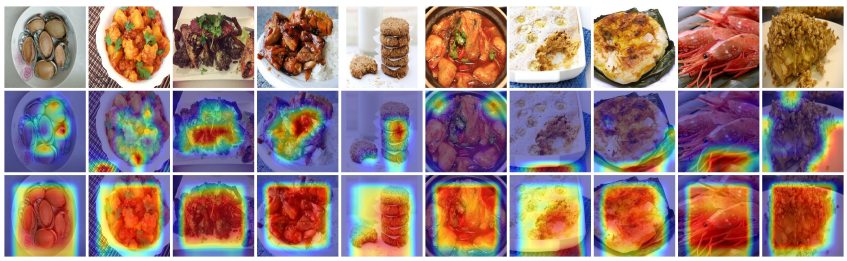
(a) Examples from dataset ETHZ Food-101.



(b) Examples from dataset Vireo Food-172.



(c) Examples from dataset UEC Food256.



(d) Examples from dataset ISIA Food-500.

Fig. 5. Visualization of experimental results comparison. (a)(b)(c)(d): Examples from dataset Food101, 172, 256, 500. The first row is the original image, the second row is the heat map without aggregation, and the third row is the heat map with aggregation; the first five columns have achieved good recognition results, and the last five columns only use aggregation recognition correct, if aggregation is not used, the recognition will fail.

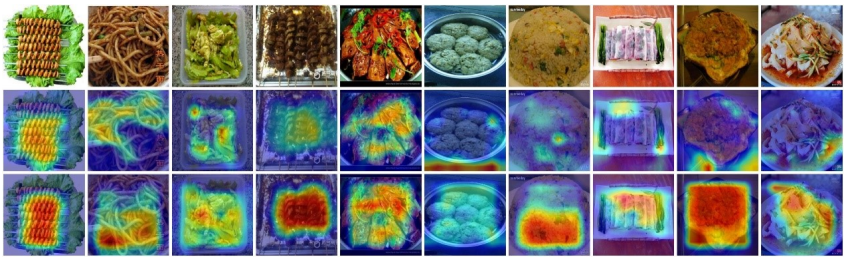


Fig. 6. Another Visualization of experimental results from Vireo Food-172.

Table 9. Ablation study. Comparison of AFNet variants with and without Aggregation Block/SoftReLU activation function when trained on Food-101, Food-172, Food256, Food-500 dataset.

Dataset	Method	Top-1 Acc.	#Params	#FLOPs
Food-101	w/o Aggregation Block	84.75%	0.61M	162.17M
	w/ Hardswish	86.21%	0.61M	164.36M
	w/ ReLU	85.62%	0.61M	164.36M
	AFNet-1.0	**86.43%**	0.61M	164.36M
Food-172	w/o Aggregation Block	86.38%	0.70M	162.26M
	w/ Hardswish	87.08%	0.70M	164.45M
	w/ ReLU	86.75%	0.70M	164.45M
	AFNet-1.0	**87.10%**	0.70M	164.45M
Food256	w/o Aggregation Block	68.29%	0.81M	162.37M
	w/ Hardswish	68.55%	0.81M	164.56M
	w/ ReLU	68.51%	0.81M	164.56M
	AFNet-1.0	**69.31%**	0.81M	164.56M
Food-500	w/o Aggregation Block	59.64%	1.12M	162.37M
	w/ Hardswish	60.72%	1.12M	164.56M
	w/ ReLU	60.21%	1.12M	164.56M
	AFNet-1.0	**60.74%**	1.12M	164.56M

86.43%(Food-101), 87.10%(Food-172), 69.31%(Food-256) and 60.74%(Food-500) compared to models without aggregation blocks: 84.75%(Food-101), 86.38%(Food-172), 68.29%(Food-256) and 59.64%(Food-500). That indicates the aggregation block is effective to improve model accuracy by gathering long-range features.

Activation Function. By virtue of the aggregation block cutting down on the number of network parameters and size, a more ambitious activation function named SoftReLU has been adopted during network optimization in our work. Here we compared the effectiveness of the proposed activation function SoftReLU with two typical activation functions Hardswish and ReLU. As seen in Table 9, the models using SoftReLU achieve higher top-1 accuracy: 86.43%(Food-101), 87.10%(Food-172), 69.31%(Food-256) and 60.74%(Food-500), compared to models using Hardswish and ReLU: 86.21%/85.62%(Food-101), 87.08%/86.75%(Food-172), 68.55%/68.51%(Food-256) and 60.72%/60.21%(Food-500). The results show that the SoftReLU activation function helps to find better solutions.

5 CONCLUSIONS

With the intention of focusing on the characteristics of food images, a lightweight CNN-based model AFNet is proposed for the recognition of food images. The model can proficiently collect both local and global features from food images. We developed a convolution based on the residual model, which encodes global features by combining information from the row and the column. Combining the aggregation block with classic local convolution creates a framework that serves as the backbone of the network. On the basis of the efficient utilization of the parameters by aggregation blocks, we have developed a lightweight network of image recognition for food, with fewer layers and smaller scales, and supported by the new activation function SoftReLU. Experimental results on four popular databases of food images show that our method achieves the best performance compared to existing CNN-based, ViT-based, and hybrid lightweight network models.

REFERENCES

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101 - Mining Discriminative Components with Random Forests. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI (Lecture Notes in Computer Science, Vol. 8694)*, David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer, 446–461. https://doi.org/10.1007/978-3-319-10599-4_29

[2] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. 2018. Optimization Methods for Large-Scale Machine Learning. *SIAM Rev.* 60, 2 (2018), 223–311. https://doi.org/10.1137/16M1080173

[3] Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S-H Gary Chan. 2023. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12021–12031.

[4] Jingjing Chen and Chong-Wah Ngo. 2016. Deep-based Ingredient Recognition for Cooking Recipe Retrieval. In *Proceedings of the 2016 ACM Conference on Multimedia Conference, MM 2016, Amsterdam, The Netherlands, October 15-19, 2016*, Alan Hanjalic, Cees Snoek, Marcel Worring, Dick C. A. Bulterman, Benoit Huet, Aisling Kelliher, Yiannis Kompatsiaris, and Jin Li (Eds.). ACM, 32–41. https://doi.org/10.1145/2964284.2964315

[5] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. 2022. Mobile-Former: Bridging MobileNet and Transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 5260–5269. https://doi.org/10.1109/CVPR52688.2022.00520

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[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. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=YicbFdNTTy

[8] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. 2022. CMT: Convolutional Neural Networks Meet Vision Transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 12165–12175. https://doi.org/10.1109/CVPR52688.2022.01186

[9] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. 2020. GhostNet: More Features From Cheap Operations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 1577–1586. https://doi.org/10.1109/CVPR42600.2020.00165

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 770–778. https://doi.org/10.1109/CVPR.2016.90

[11] Shota Horiguchi, Sosuke Amano, Makoto Ogawa, and Kiyoharu Aizawa. 2018. Personalized Classifier for Food Image Recognition. *IEEE Trans. Multim.* 20, 10 (2018), 2836–2848. https://doi.org/10.1109/TMM.2018.2814339

[12] Andrew Howard, Ruoming Pang, Hartwig Adam, Quoc V. Le, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, and Yukun Zhu. 2019. Searching for MobileNetV3. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 1314–1324. https://doi.org/10.1109/ICCV.2019.00140

[13] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR* abs/1704.04861 (2017). arXiv:1704.04861 http://arxiv.org/abs/1704.04861

[14] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[15] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-Excitation Networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 7132–7141. https://doi.org/10.1109/CVPR.2018.00745

[16] Tao Huang, Lang Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. 2022. LightViT: Towards Light-Weight Convolution-Free Vision Transformers. *CoRR* abs/2207.05557 (2022). https://doi.org/10.48550/arXiv.2207.05557 arXiv:2207.05557

[17] Akihisa Ishino, Yoko Yamakata, Hiroaki Karasawa, and Kiyoharu Aizawa. 2021. RecipeLog: Recipe Authoring App for Accurate Food Recording. In *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*, Heng Tao Shen, Yueting Zhuang, John R. Smith, Yang Yang, Pablo César, Florian Metze, and Balakrishnan Prabhakaran (Eds.). ACM, 2798–2800. https://doi.org/10.1145/3474085.3478563

[18] Shuqiang Jiang, Weiqing Min, Linhu Liu, and Zhengdong Luo. 2020. Multi-Scale Multi-View Deep Feature Aggregation for Food Recognition. *IEEE Trans. Image Process.* 29 (2020), 265–276. https://doi.org/10.1109/TIP.2019.2929447

[19] Hokuto Kagaya, Kiyoharu Aizawa, and Makoto Ogawa. 2014. Food Detection and Recognition Using Convolutional Neural Network. In *Proceedings of the ACM International Conference on Multimedia, MM '14, Orlando, FL, USA, November 03 - 07, 2014*, Kien A. Hua, Yong Rui, Ralf Steinmetz, Alan Hanjalic, Apostol Natsev, and Wenwu Zhu (Eds.). ACM, 1085–1088. https://doi.org/10.1145/2647868.2654970

[20] Yoshiyuki Kawano and Keiji Yanai. 2013. Real-Time Mobile Food Recognition System. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2013, Portland, OR, USA, June 23-28, 2013*. IEEE Computer Society, 1–7. https://doi.org/10.1109/CVPRW.2013.5

[21] Yoshiyuki Kawano and Keiji Yanai. 2014. FoodCam-256: A Large-scale Real-time Mobile Food RecognitionSystem employing High-Dimensional Features and Compression of Classifier Weights. In *Proceedings of the ACM International Conference on Multimedia, MM '14, Orlando, FL, USA, November 03 - 07, 2014*, Kien A. Hua, Yong Rui, Ralf Steinmetz, Alan Hanjalic, Apostol Natsev, and Wenwu Zhu (Eds.). ACM, 761–762. https://doi.org/10.1145/2647868.2654869

[22] Yoshiyuki Kawano and Keiji Yanai. 2015. FoodCam: A real-time food recognition system on a smartphone. *Multim. Tools Appl.* 74, 14 (2015), 5263–5287. https://doi.org/10.1007/s11042-014-2000-8

[23] Jiashi Li, Xin Xia, Wei Li, Huixia Li, Xing Wang, Xuefeng Xiao, Rui Wang, Min Zheng, and Xin Pan. 2022. Next-ViT: Next Generation Vision Transformer for Efficient Deployment in Realistic Industrial Scenarios. *CoRR* abs/2207.05501 (2022). https://doi.org/10.48550/arXiv.2207.05501 arXiv:2207.05501

[24] Yanyu Li, Geng Yuan, Yang Wen, Eric Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. 2022. EfficientFormer: Vision Transformers at MobileNet Speed. *CoRR* abs/2206.01191 (2022). https://doi.org/10.48550/arXiv.2206.01191 arXiv:2206.01191

[25] Jihao Liu, Xin Huang, Guanglu Song, Hongsheng Li, and Yu Liu. 2022. UniNet: Unified Architecture Search with Convolution, Transformer, and MLP. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXI (Lecture Notes in Computer Science, Vol. 13681)*, Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer, 33–49. https://doi.org/10.1007/978-3-031-19803-8_3

[26] Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. 2023. EfficientViT: Memory Efficient Vision Transformer with Cascaded Group Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14420–14430.

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 9992–10002. https://doi.org/10.1109/ICCV48922.2021.00986

[28] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. 2018. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV (Lecture Notes in Computer Science, Vol. 11218)*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer, 122–138. https://doi.org/10.1007/978-3-030-01264-9_8

[29] Niki Martinel, Gian Luca Foresti, and Christian Micheloni. 2018. Wide-Slice Residual Networks for Food Recognition. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*. IEEE Computer Society, 567–576. https://doi.org/10.1109/WACV.2018.00068

[30] Sachin Mehta and Mohammad Rastegari. 2022. MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=vh-0sUt8HlG

[31] Sachin Mehta and Mohammad Rastegari. 2022. Separable Self-attention for Mobile Vision Transformers. *CoRR* abs/2206.02680 (2022). https://doi.org/10.48550/arXiv.2206.02680 arXiv:2206.02680

[32] Sachin Mehta, Mohammad Rastegari, Linda G. Shapiro, and Hannaneh Hajishirzi. 2019. ESPNetv2: A Light-Weight, Power Efficient, and General Purpose Convolutional Neural Network. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 9190–9200. https://doi.org/10.1109/CVPR.2019.00941

[33] Weiqing Min, Shuqiang Jiang, Linhu Liu, Yong Rui, and Ramesh C. Jain. 2019. A Survey on Food Computing. *ACM Comput. Surv.* 52, 5 (2019), 92:1–92:36. https://doi.org/10.1145/3329168

[34] Weiqing Min, Linhu Liu, Zhengdong Luo, and Shuqiang Jiang. 2019. Ingredient-Guided Cascaded Multi-Attention Network for Food Recognition. In *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, Laurent Amsaleg, Benoit Huet, Martha A. Larson, Guillaume Gravier, Hayley Hung, Chong-Wah Ngo, and Wei Tsang Ooi (Eds.). ACM, 1331–1339. https://doi.org/10.1145/3343031.3350948

[35] Weiqing Min, Linhu Liu, Zhiling Wang, Zhengdong Luo, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. 2020. ISIA Food-500: A Dataset for Large-Scale Food Recognition via Stacked Global-Local Attention Network. In *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*, Chang Wen Chen, Rita Cucchiara, Xian-Sheng Hua, Guo-Jun Qi, Elisa Ricci, Zhengyou Zhang, and Roger Zimmermann (Eds.). ACM, 393–401. https://doi.org/10.1145/3394171.3414031

[36] Weiqing Min, Zhiling Wang, Yuxin Liu, Mengjiang Luo, Liping Kang, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. 2023. Large Scale Visual Food Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 8 (2023), 9932–9949. https://doi.org/10.1109/TPAMI.2023.3237871

[37] Weiqing Min, Zhiling Wang, Yuxin Liu, Mengjiang Luo, Liping Kang, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. 2023. Large Scale Visual Food Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023), 1–18. https://doi.org/10.1109/TPAMI.2023.3237871

[38] Kei Nakamoto, Sosuke Amano, Hiroaki Karasawa, Yoko Yamakata, and Kiyoharu Aizawa. 2022. Prediction of Mental State from Food Images. In *CEA++@MM 2022: Proceedings of the 1st International Workshop on Multimedia for Cooking, Eating, and related APPlications, Lisboa, Portugal, 10 October 2022*, Yoko Yamakata, Atsushi Hashimoto, and Jingjing Chen (Eds.). ACM, 21–28. https://doi.org/10.1145/3552485.3554937

[39] Junting Pan, Adrian Bulat, Fuwen Tan, Xiatian Zhu, Lukasz Dudziak, Hongsheng Li, Georgios Tzimiropoulos, and Brais Martínez. 2022. EdgeViTs: Competing Light-weight CNNs on Mobile Devices with Vision Transformers. *CoRR* abs/2205.03436 (2022). https://doi.org/10.48550/arXiv.2205.03436 arXiv:2205.03436

[40] Parisa Pouladzadeh and Shervin Shirmohammadi. 2017. Mobile Multi-Food Recognition Using Deep Learning. *ACM Trans. Multim. Comput. Commun. Appl.* 13, 3s (2017), 36:1–36:21. https://doi.org/10.1145/3063592

[41] Xu Qin and Zhilin Wang. 2019. Nasnet: A neuron attention stage-by-stage net for single image deraining. *arXiv preprint arXiv:1912.03151* (2019).

[42] Javier Ródenas, Bhalaji Nagarajan, Marc Bolaños, and Petia Radeva. 2022. Learning Multi-Subset of Classes for Fine-Grained Food Recognition. In *Proceedings of the 7th International Workshop on Multimedia Assisted Dietary Management on Multimedia Assisted Dietary Management, MADiMa 2022, Lisboa, Portugal, 10 October 2022*, Stavroula G. Mougiakakou, Giovanni Maria Farinella, Keiji Yanai, and Dario Allegra (Eds.). ACM, 17–26. https://doi.org/10.1145/3552484.3555754

[43] Ali Rostami, Nitish Nagesh, Amir Rahmani, and Ramesh C. Jain. 2022. World Food Atlas for Food Navigation. In *Proceedings of the 7th International Workshop on Multimedia Assisted Dietary Management on Multimedia Assisted Dietary Management, MADiMa 2022, Lisboa, Portugal, 10 October 2022*, Stavroula G. Mougiakakou, Giovanni Maria Farinella, Keiji Yanai, and Dario Allegra (Eds.). ACM, 39–47. https://doi.org/10.1145/3552484.3555748

[44] Ali Rostami, Vaibhav Pandey, Nitish Nag, Vesper Wang, and Ramesh C. Jain. 2020. Personal Food Model. In *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*, Chang Wen Chen, Rita Cucchiara, Xian-Sheng Hua, Guo-Jun Qi, Elisa Ricci, Zhengyou Zhang, and Roger Zimmermann (Eds.). ACM, 4416–4424. https://doi.org/10.1145/3394171.3414691

[45] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

[46] 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), 136309.

[47] Guorui Sheng, Weiqing Min, Xiangyi Zhu, Liang Xu, Qingshuo Sun, Yancun Yang, Lili Wang, and Shuqiang Jiang. 2024. A Lightweight Hybrid Model with Location-Preserving ViT for Efficient Food Recognition. *Nutrients* 16, 2 (2024), 200.

[48] Guorui Sheng, Shuqi Sun, Chengxu Liu, and Yancun Yang. 2022. Food recognition via an efficient neural network with transformer grouping. *Int. J. Intell. Syst.* 37, 12 (2022), 11465–11481. https://doi.org/10.1002/int.23050

[49] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. 2021. Bottleneck Transformers for Visual Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 16519–16529. https://doi.org/10.1109/CVPR46437.2021.01625

[50] Ghalib Ahmed Tahir and Chu Kiong Loo. 2021. A comprehensive survey of image-based food recognition and volume estimation methods for dietary assessment. In *Healthcare*, Vol. 9. MDPI, 1676.

[51] Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 6105–6114. http://proceedings.mlr.press/v97/tan19a.html

[52] Ren Zhang Tan, XinYing Chew, and Khai Wah Khaw. 2021. Neural Architecture Search for Lightweight Neural Network in Food Recognition. *Mathematics* 9, 11 (2021), 1245.

[53] Yehui Tang, Kai Han, Jianyuan Guo, Chang Xu, Chao Xu, and Yunhe Wang. 2022. GhostNetV2: Enhance Cheap Operation with Long-Range Attention. *CoRR* abs/2211.12905 (2022). https://doi.org/10.48550/arXiv.2211.12905 arXiv:2211.12905

[54] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. 2023. MobileOne: An Improved One millisecond Mobile Backbone. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 7907–7917. https://doi.org/10.1109/CVPR52729.2023.00764

[55] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. 2021. CvT: Introducing Convolutions to Vision Transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 22–31. https://doi.org/10.1109/ICCV48922.2021.00009

[56] Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. 2022. TinyViT: Fast Pretraining Distillation for Small Vision Transformers. *CoRR* abs/2207.10666 (2022). https://doi.org/10.48550/arXiv.2207.10666 arXiv:2207.10666

[57] Yoko Yamakata, Akihisa Ishino, Akiko Sunto, Sosuke Amano, and Kiyoharu Aizawa. 2022. Recipe-oriented Food Logging for Nutritional Management. In *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, João Magalhães, Alberto Del Bimbo, Shin'ichi Satoh, Nicu Sebe, Xavier Alameda-Pineda, Qin Jin, Vincent Oria, and Laura Toni (Eds.). ACM, 6898–6904. https://doi.org/10.1145/3503161.3549203

[58] Shulin Yang, Mei Chen, Dean Pomerleau, and Rahul Sukthankar. 2010. Food recognition using statistics of pairwise local features. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*. IEEE Computer Society, 2249–2256. https://doi.org/10.1109/CVPR.2010.5539907

[59] Jinnian Zhang, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. 2022. MiniViT: Compressing Vision Transformers with Weight Multiplexing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 12135–12144. https://doi.org/10.1109/CVPR52688.2022.01183

[60] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6848–6856.