applied sciences

MDPI

Article

φunit: A Lightweight Module for Feature Fusion Based on Their Dimensions

Zhengyu Long [1,2], Rigui Zhou [1,2,*], Yaochong Li [1,2], Pengju Ren [1,2], Xue Yang [1,2] and Shuo Cai [1,2]

[1] College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China
[2] Research Center of Intelligent Information Processing and Quantum Intelligent Computing, Shanghai 201306, China
* Correspondence: rgzhou@shmtu.edu.cn

Abstract: With the popularity of mobile devices, lightweight deep learning models have important value in various application scenarios. However, how to effectively fuse the feature information from different dimensions while ensuring the model's lightness and high accuracy is a problem that has not been fully solved. In this paper, we propose a novel feature fusion module, called φunit, which can fuse the features extracted by different dimensional networks according to the order of feature information with a small computational cost, avoiding the problems of information fragmentation caused by simple feature stacking in traditional information fusion. Based on φunit, this paper further builds an extremely lightweight model φNet, which can achieve performance close to the highest accuracy on several public datasets under the condition of very limited parameter scale. The core idea of φunit is to use deconvolution to reduce the discrepancy among the features to be fused, and to lower the possibility of feature information fragmentation after fusion by fusing the features from different dimensions sequentially. φNet is a lightweight network composed of multiple φunits and bottleneck modules, with a parameter scale of only 1.24 M, much smaller than traditional lightweight models. This paper conducts experiments on public datasets, and φNet achieves an accuracy of 71.64% on the food101 dataset, and an accuracy of 75.31% on the random 50-category food101 dataset, both higher than or close to the highest accuracy. This paper provides a new idea and method for feature fusion of lightweight models, and also provides an efficient model selection for deep learning applications on mobile devices.

Keywords: image classification; lightweight model; φunit; φNet

check for updates

Citation: Long, Z.; Zhou, R.; Li, Y.; Ren, P.; Yang, X.; Cai, S. φunit: A Lightweight Module for Feature Fusion Based on Their Dimensions. *Appl. Sci.* **2023**, *13*, 12621. https://doi.org/10.3390/app132312621

Academic Editors: Huaqing Li and Qingguo Lü

Received: 11 October 2023
Revised: 16 November 2023
Accepted: 21 November 2023
Published: 23 November 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Computer vision is an important branch of artificial intelligence, aiming to transform visual information into understandable content. This field covers multiple emerging domains such as object detection [1–3], semantic segmentation [4–6], face recognition [7–9], pose estimation [10–12], etc., and the real-world applications of these domains have brought great convenience to daily life and production. To make artificial intelligence better serve society, improving the accuracy of model prediction performance is the core issue of this field. Common improvement strategies include expanding network width and establishing connections from forward to backward [13]. For example, LeNet [14], ZFNet [15] and VGGNet [16] all improve accuracy by increasing the depth or width of convolutional layers in different ways. However, as the network depth increases, the network may have problems such as gradient explosion, and the computation is huge, which cannot be well deployed and used in many application devices. Therefore, researchers turned to more sophisticated model designs, such as residual structure and multi-branch structure. Residual structure triggered the development of a series of excellent networks, such as Mobinetv2 [17], DenseNet [18], ResNeXt [19], WideResNet [20] and PyramidNet [21]. The proposal of Inception was to achieve better performance by reducing the theoretical

complexity, and its branch-like topology structure could also ensure that it could learn more spatial correlation features [22], such as GoogleNet [23] and its derived networks SqueezeNet [24] and Xception [25]. However, these networks still have limitations in information extraction ability. To distinguish the importance of different information, researchers introduced channel attention mechanism, such as SE [26], CBAM [27] and ECA [28], which improved model performance by recalibrating feature responses.

As CNN gradually encountered bottlenecks in the pursuit of high accuracy, the emergence of ViT [29] aroused researchers' strong interest in transformer. Transformer's completely different network structure and higher accuracy from CNN attracted a large number of researchers to explore it in depth. However, the huge computation and parameter amount of transformer limited its application on mobile devices with limited resources (such as smartphones and unmanned driving devices). Although there are some methods, such as network pruning [30–33], knowledge distillation [34–36], structure reparameterization [37], model quantization [38,39], etc., that can reduce the parameter amount and computation of the model without significantly reducing the accuracy, these methods are limited by the model itself and cannot solve the problem fundamentally. Therefore, considering the demand of deploying and running lightweight models on mobile devices, developing efficient network structures based on CNN to improve the feature information extraction density of CNN modules, and reducing the parameter amount and computation of the network, is promising work [40].

In conventional network architectures, feature information from different network levels is fused by first determining whether convolution is needed to align the feature shapes, and then applying concatenation or addition operation. These operations are commonly used across different feature extraction layers to aggregate features from various levels [41]. However, they merely combine features without taking into account the spatial relationship among different information sources. This may cause the problem of feature information discontinuity in the fused feature information, which affects the inference quality of the final model. Figure 1a–c illustrates the common information fusion methods. Inspired by classical networks such as U-Net [41], Ghost [40] and MobinetV2 [17], an efficient network hierarchical connection structure, called φunit (as shown in Figure 1d) was proposed. Based on the bottleneck module of MobinetV2 [17] and the connection module proposed in this paper, an efficient network φNet was built. To explore the effectiveness of φunit and the performance of φNet, this paper conducts extensive ablation experiments, comparative experiments and validation evaluations, and tests with different datasets. The results of various experiments confirm the performance advantages of the novel structure proposed in this paper.

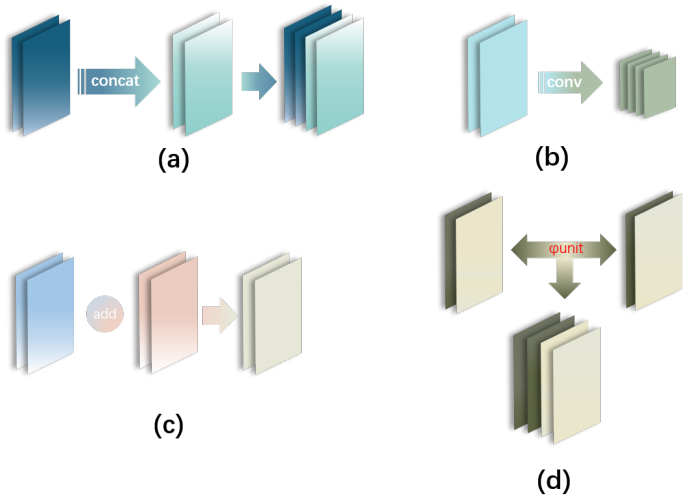


Figure 1. Different feature fusion methods: (**a**) concatenation of feature information; (**b**) convolutional modification of feature information; (**c**) summation of feature information fusion; (**d**) sequential fusion of feature information.

The key contributions of this work are summarized as follows:

- The φunit information fusion module is proposed, which considers the spatial feature relationship, a factor that is often neglected in the traditional model building strategy.
- Based on the φunit, combined with the relevant research results of lightweight models, this paper constructs a network model φNet that can achieve a better balance among model parameter amount, computation and accuracy.
- The deconvolution operation is introduced, which is used to reduce the difference between feature information, and it is applied to the image classification domain.

This paper is organized as follows. Section 2 gives a brief overview of the related work in the field. In Section 3, we introduce φunit and φNet and explain the network structure and related functions. Section 4 conducts a series of experiments and analyzes the results. Section 5 concludes the whole research process.

2. Related Work

We present a review of the state-of-the-art research on model lightweighting, which covers both model compression and pre-designed models, in this section.

2.1. Model Compression

Model lightweighting aims to reduce the computation cost of a model without sacrificing its accuracy [30]. This chapter introduces four methods for achieving this goal: knowledge distillation [34–36], using a better teacher network to guide the student network, reducing the difference between the teacher network and the student network in inference, and making the student network close to or even surpass the performance of the teacher network in actual performance ability; a quantized neural network [38,39], a method that compresses the model size and computation by quantizing the model weights to a smaller numerical range; structural reparameterization [37], a strategy that converts a multi-branch network with strong representation ability into a serial structure with less parameters and faster inference; and a pruning algorithm [30–33], an approach that eliminates redundant neurons and activation units from the model to improve its efficiency.

2.2. Pre-Designed Models

The need for lightweight models that are suitable for mobile devices increases with the advent of the artificial intelligence era. However, this is a huge challenge, because the storage and computation power of mobile devices are limited, making many large and high-precision models inapplicable. To overcome this challenge, some lightweight models have emerged in recent years [17,24,25,40,42–44]. One of them is SqueezeNet [24], which builds an efficient neural network model by using Fire Module to combine 1×1 and 3×3 convolutions. Xception [25] is an efficient network model that uses depthwise separable convolution to build with low computational complexity and a small number of parameters. ShuffleNet [42] uses a technique called feature shuffling fusion, which randomly rearranges the features after grouped convolution, thereby achieving cross-fusion of features and keeping the number of channels unchanged. MobinetV2 [17] proposes an efficient information extraction inverted residual module, aiming to improve the efficiency of information extraction. MobinetV3 [43] is redesigned based on MobinetV2, mainly optimizing the time-consuming modules and adding SE [26] module. In addition, MobinetV3 also uses the method of neural architecture search to search for the optimal parameter settings, to further improve the performance and efficiency of the model. Through these improvements, MobinetV3 achieves better results in terms of computational efficiency, model performance and parameter configuration. GhostNet [40] uses cheap convolution, instead of traditional expensive convolution operations, to significantly reduce the cost of redundant feature generation. This strategy enables GhostNet to effectively reduce the computational resources and storage requirements while maintaining high performance. GhostNetV2 [44] introduces a decoupled fully connected attention (DFC) mechanism

to further enhance the model's feature extraction ability. The DFC mechanism enables GhostNetV2 to more precisely capture key features and effectively allocate attention in images. Furthermore, the feature extraction performance and energy efficiency of the model are significantly improved by the redesigned Ghost bottleneck unit, which optimizes the model's structure and parameter settings. Touvron et al. proposed ResMLP, a fully MLP-based image classification network that does not use convolution or self-attention mechanism, but updates the representation of image blocks through linear layers and feedforward layers. And they found that MLP has a powerful expressive ability in features, and based on their nature as nonlinear transformations, they constructed a ResMLP [45] network based on ViT as a template in the process of optimizing the number and size of MLP. Park et al. explored the mechanism of ViT in depth, and found that CNN and MSA have a smooth complementary effect in the spatial aspect, and gave some criteria for the combination design, according to their respective nature as high-pass filter and low-pass filter, and in the process of optimizing the proportion of MSA and CNN in the model, they successfully constructed AlterNet [46], an efficient network based on ResNet-50. Based on the analysis of the working mechanism of the SE module, a novel network structure named SAENet [22] was proposed. It used the inception module to extract spatial feature information from the topological branch structure, and passed the important feature information to the next layer of the network through the fully connected layer. In this way, each network node could obtain the global representation of the previous network nodes, thus improving the effect of the whole network.

3. Materials and Methods

This section provides a comprehensive presentation of the φNet architecture, and a detailed explanation of the φunit module and its constituent elements that form the core of the network. The functions and roles of the associated modules are demonstrated through comparative analysis, while their crucial significance within the model is emphasized. A brief overview of the datasets used for training and testing is also provided in this section.

3.1. Transposed Convolution

Transposed convolution, also referred to as deconvolution, is a widely used operation in convolutional neural networks. It aims to upsample images or feature maps and increase their spatial resolution. Transposed convolution is the inverse of ordinary convolution in terms of computation. In ordinary convolution, output feature maps are obtained by sliding a smaller filter window across the input image. In transposed convolution, a larger filter window is used to slide across the output feature maps and recover a larger input image, achieving upsampling [47,48]. Transposed convolution is a common technique in image super-resolution tasks, which allows the conversion of low-resolution images into high-resolution ones [49]. In semantic segmentation networks with encoder–decoder structures, transposed convolution functions in the decoder part to gradually restore the high-resolution of images and obtain semantic segmentation outcomes [50]. In addition, transposed convolution is also widely utilized in the generator modules of generative adversarial networks, which can amplify feature maps and generate high-resolution images [51]. Transposed convolution is mainly used to upsample and recover the resolution, which is widely applied in image processing, semantic understanding and generative models. However, transposed convolution suffers from the issues of feature information loss during the recovery process, which limits its usage in the classification field. Nevertheless, taking into account that transposed convolution can recover feature information, and matches well with the concept of "feature information fusion" in this study. In this work, transposed convolution is applied to the image classification field.

3.2. φunit

A module is a collection of layers with a single function in classical convolutional neural networks, which researchers often define. The bottleneck module in ResNet [13]

and the fire module in SqueezeNet [24] are examples of modules. Several modules are repeated to form a group [52].These groups are stacked together to form an efficient network structure. Through this modular design, the network can be better reused and scaled, thereby achieving more effective feature extraction. This construction strategy is widely used in many classic networks [14,16,17,40,43]. Nonetheless, the issue of inter-group connections remains underexplored.

In serial networks, there are generally two common strategies to facilitate the inter-group feature information transmission and reduce the computation and operation of the overall model, as shown in Equation (1). By performing the downsampling operation ∇ or the convolution operation ψ on the feature map, f_l obtained from the low-dimensional network, the dimension or size of f_l was reduced, and then the processed result f_s was connected to the subsequent high-dimensional network, with f_s as the input condition of the high-dimensional network. However, such a strategy, in the view of this paper, might often lead to the loss of relevant shallow feature key information of the low-dimensional network inference result when it served as the input condition of the high-dimensional network, thus affecting the final inference effect of the model.

$$f_s = \begin{cases} \psi(f_l), & \text{convolution operation} \\ \nabla(f_l), & \text{downsampling operation} \end{cases} \tag{1}$$

In parallel networks, there are two common ways to fuse the inference results of multiple parallel network branches, as shown in Equation (2): one is to perform the Concat concatenation operation τ, and the other is to perform the bitwise addition operation ξ on the features inferred from different branches. After this operation, the overall summary feature f_p of the low-dimensional feature information l_1, l_2, l_3……, l_n is obtained, which serves as a condition to be passed to the high-dimensional network. However, in this work's view, these conventional information fusion methods are just simple information concatenation operations, which do not take into account the dimensionality of the information and the continuity of the information after fusion. The image information after concatenation may cause the consequence of feature information discontinuity due to the large gap between the information, which may have a negative impact on the final model inference result.

$$f_p = \begin{cases} \tau(l_1 : l_2 : l_3 \ldots\ldots : l_n), & \text{Concat concatenation operation} \\ \xi(l_1 : l_2 : l_3 \ldots\ldots : l_n), & \text{bitwise addition operation} \end{cases} \tag{2}$$

To solve the connection and information fusion problem between groups, a connection strategy named φ unit is introduced in this work. The schematic diagram is presented in Figure 2. The φunit module offers two advantages: first, it can reduce the difference of feature information among different groups, and ensure the consistency of the information after fusion; second, it can relieve the forgetting problem of high-dimensional networks for feature extraction from low-dimensional networks caused by the model depth increase. The information fusion process is demonstrated in the following equation:

$$f_\varphi = \nabla(f_l) \odot \gamma(f_h) \tag{3}$$

The equations demonstrates that f_h is the high-dimensional network's extracted feature information. To enhance the feature information similarity across different dimensions and retain more high-dimensional feature information, in this work, the γ operation is applied to deconvolve f_h, reducing the information discontinuity risk. This operation only adjusts the channels of f_h to match the downsampling ∇ result of the low-dimensional feature information f_l. Then, the feature information fusion operation $\odot$ is performed. The fused feature information f_φ has twice the dimension of f_l and half the size. This operation not only addresses the information discontinuity issue in traditional fusion operations, but also preserves shallow network features and lift network feature dimensions. Moreover, the aforementioned operation also augments the informational content of pixel

units at the same spatial location. This provides more abundant feature map information for the inference of subsequent network layers.

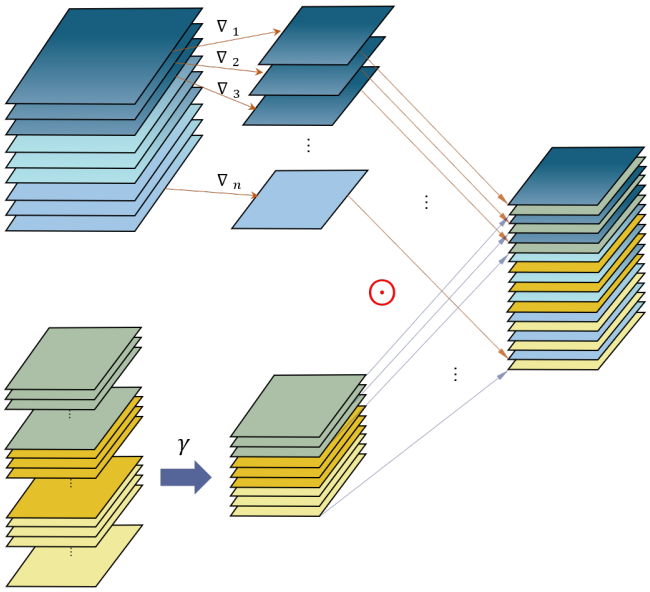


Figure 2. Explains cross concatenation in the proposed unit.

A new connection strategy that addresses the aim of achieving more effective information communication and fusion among groups is introduced by using the φunit module. This improvement enhances network performance and inference accuracy. The relevant verification will be detailed and clarified in Section 4.1. There, experimental results and quantitative analysis are used to verify and assess the benefits of the φunit module in information communication and fusion, and its impact on network performance. Such verification will further prove the φunit module's effectiveness in addressing the group connection and information fusion problem.

3.3. φNet

φNet is a novel network architecture, which is inspired by classic networks such as U-Net [41], GhostNet [40] and MobileNetV2 [17]. In φNet, the up-convolution operation from U-Net [41] is adopted in this work, and we introduce φunit to enable cross-level feature fusion. φunit is one of the basic units of φNet, which fuses the high-dimensional feature information and low-dimensional feature information extracted by the bottleneck group. The fused feature information comprises both abstract information from high-dimensional convolution and superficial information from low-dimensional convolution. Specifically, we fuse the two feature maps by cross-stitching them following the channel order. By this way, each pixel position in the fused feature map possesses feature representations from both the current and the previous levels. Furthermore, this study also takes into account the phenomenon of excessive redundant feature information in the extracted feature images [40]. By meticulously designing the network structure and feature selection strategy, φNet can generate low-cost redundant features with lower cost and improve the overall performance. Additionally, the bottleneck structure from MobileNetV2 is also employed in this work to further increase the representation power and convergence speed of φNet. A network called φNet, which integrates the strengths of U-Net, GhostNet and MobileNetV2, is built in this study. By using key techniques such as up-convolution, feature fusion and residual structure, φNet can achieve excellent performance in image processing tasks. The structure and organization of the related network is displayed in Table 1.

This paper uses the bottleneck module in Mobinetv2 to extract basic feature information, and adjusts the feature depth and size according to the dimension change rule and network actual characteristics of Mobinetv2. The network construction strategy is as

follows: first, use one convolution layer and three bottleneck layers to obtain the shallowest feature of the original image; second, use multiple bottleneck structures to extract high-dimensional abstract features from low-dimensional features; then, use the φunit proposed in this paper to fuse the extracted features twice; finally, convert all fused image features into final classification information.

Table 1. The overall architecture of φNet.

Input	Operator	t	c	n	s
$224^2 \times 3$	Conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	4	2
$28^2 \times 32$	φunit	-	48	1	-
$28^2 \times 48$	bottleneck	6	48	5	2
$14^2 \times 48$	φunit	-	64	1	-
$14^2 \times 64$	bottleneck	6	96	4	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	avgpool	-	-	1	-
$1^2 \times 160$	linear	-	k	-	-

t denotes expansion factor, c denotes output dimension, n denotes repetition count, s denotes the stride value used in the first block.

This study proposes a novel network φNet, which can fully capture the feature space and logical feature continuity. The overview of the network structure is shown in Figure 3. The network structure used in this study constructs the low-dimensional and high-dimensional representations of the basic features through the downsampling layer and the feature extraction layer, respectively, after performing the basic feature extraction on the original image. Next, the high-dimensional features obtained are upsampled to narrow the gap with the low-dimensional features, and a downsampling operation is also carried out, which serves as one of the input contents for the next feature fusion. The function of φunit is to fuse the processed high-dimensional and low-dimensional features and pass them to the next layer, so as to achieve the goal of retaining the high-dimensional abstract features while alleviating the problem of low-dimensional feature loss caused by the network depth increase. This process is executed twice in total, and the detailed proof is given in Section 4.2. Finally, in order to integrate the global features and prepare for the prediction task, the final fused features are subjected to a final feature extraction, and then input to the classification module to complete the classification task. The effectiveness of the network is verified by a series of comparative experiments in Section 4.3. It is worth noting that the framework design proposed in this paper is not the only feasible method, and there is still much potential for improvement for the φNet network. For example, by applying cutting-edge techniques such as neural architecture search, the network structure can be optimized, thereby improving the network performance. Therefore, this paper hopes that future researchers can further explore and improve the φNet network to achieve better results.

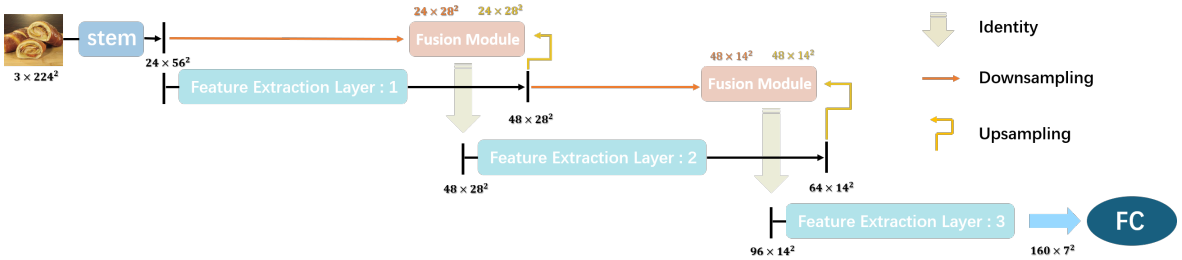


Figure 3. φNet Construction.

3.4. Datasets

This study uses two public datasets (food101 and Dog vs. Cat), and divides them into three different types of datasets for different purposes of the experiment, to verify and test whether the proposed φunit and φNet have the expected performance advantages. All experimental datasets are divided into training sets and validation sets according to the ratio of 8:2, and experiments are conducted on these datasets. This study uses a series of measures to increase the training difficulty in the preprocessing stage of the training set data samples, in order to enhance the generalization ability of the model on the validation set, including but not limited to randomly cropping the training pictures to 224×224 after enlarging them to 256×256, randomly flipping the pictures, regularization and other basic operations. For the validation set data, this study simply enlarges the images to 256×256 and crops them from the center to 224×224, and then regularizes them to improve validation efficiency. The training strategy of ghostv2 [44] is adopted in this paper, with AdamW [53] as the optimizer, and batchsize, lr and wd set to 128, 5×10^{-4} and 5×10^{-2}, respectively. All experiments are done on Nvidia4090 with 100 epochs of training and validation operations.

In Section 4.1 of this paper, the module effectiveness verification experiment uses the Dog VS. Cat dataset to verify the effectiveness of the φunit proposed in this paper. The Dog VS. Cat dataset is used as a testbed, which comprises 12,461 images of dogs and 12,470 images of cats, for a binary classification problem. The reasons for selecting this dataset are as follows: First, a smaller number of categories leads to a high baseline accuracy for the model. This demands a strong feature extraction ability from the model, which highlights the advantage of the proposed module in this work. Second, the dataset contains a large and balanced amount of samples. This helps us to train the model nodes sufficiently, and reduce the effect of external factors (such as sample imbalance, inadequate node training, etc.) on the outcomes. Therefore, the accuracy difference resulting from model variations is perceptible to us. Thus, using the Dog VS. Cat dataset allows the model to be tested in a neat and controlled environment, and obtain an intuition of its performance in this work.

This paper explores the module location and number in Section 4.2, and adjusts the food101 dataset adopted in the experiments appropriately, to investigate how φunit location and number variations affect φNet performance. The food101 dataset was chosen as the basic dataset for this experiment. The dataset contains 101 different food categories, each with about 1000 images, totaling about 100,000 images. The reason for choosing this dataset is that it is closer to the actual application scenario of the model in the future, and it contains rich scene noise, which has a greater impact on the key feature information in the feature maps extracted by the network. Therefore, by using this dataset and combining the test results, this experiment can reflect the robustness of the model in information extraction and classification in complex scenes to a certain extent. To shorten the model training time, this experiment randomly extracted image data of 50 classes from the basic dataset as the dataset for this experiment, and explored the impact of φunit module location and number on model performance through a series of comparative experiments and validation experiments.

Section 4.3 of this paper conducts model comparison validation experiments, and in order to better demonstrate the advantages of the proposed φNet over many widely validated lightweight networks, this experiment uses the complete food101 dataset. The relevant experimental results and model parameters can help us have a comprehensive understanding of the model, and also provide a comprehensive evaluation of its performance.

This paper will demonstrate the experimental operations on the datasets mentioned above in detail in the following sections, and offer corresponding data support. These experimental results will assist in evaluating and comparing the performance of various models on the multi-classification task.

4. Experiments and Result

This section divides the whole experiment into three parts, namely the effectiveness verification of the φunit module, the selection of the position and number of the φunit module, and the comparison of the actual measurement results of the model. Through these experiments, this section aims to comprehensively demonstrate the effectiveness of the proposed φunit module and the excellent performance of the φNet model.

4.1. Module Effectiveness Verification

To explore whether φunit has the expected efficiency and portability in future application scenarios, this experiment selects the more operable VGG16 [16] as the baseline model for this experiment. For the position selection of adding or replacing the verification module, this experiment selects between different groups where both dimension and feature size change, and the whole experimental process is shown in Figure 4. To evaluate whether φunit has good performance at any position in the model, this experiment adds a combination of deconvolution and φunit at three positions in the baseline model, respectively, for testing; to verify the efficiency of the proposed φunit compared with the conventional information fusion module, this experiment adds two extra control experiments between the dimension change from 128 to 256 in the baseline model. The relevant experimental results are shown in Table 2.

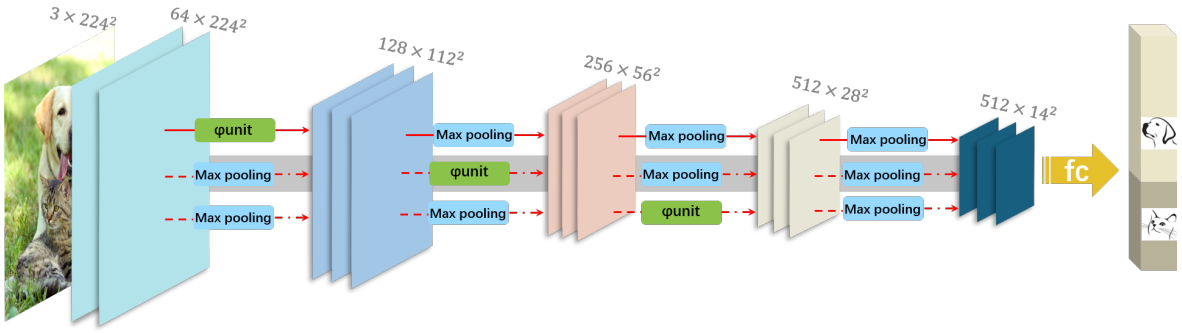


Figure 4. This figure illustrates the network for evaluating the efficiency and portability of the φunit design based on VGG16. The original image information is fed into the network after being improved by VGG16 to extract feature information, and then the feature information is passed to the FC layer to obtain the classification information obtained by the network after inference. Each red horizontal connection in the diagram represents a validation experiment scheme.

Table 2. Different connectivity handling operations are used at different dimensional variations in the VGG16 network.

Channel	Operation	Acc. (%)
-	-	97.33
64–128	TransConv + φunit	97.07
128–256	TransConv + φunit	**97.77**
128–256	TransConv + Cat	97.03
128–256	TransConv + Conv	97.53
256–512	TransConv + φunit	**97.51**

"Channel" denotes the channel variations in the connections.

The experimental results in Table 2 are based on the Dog VS. Cat binary classification dataset. Since the dataset has fewer classification categories, VGG16 can achieve a baseline accuracy of 97.3% on the dataset without any modification. This also indicates that, to further improve the accuracy on such a high accuracy basis, the extra replaced and added modules need to have strong feature extraction and information integration capabilities,

and have good feature integration in the whole model. Only in this way can the added and replaced modules make a meaningful contribution to the further performance improvement of the model. In all the improved models with added φunit modules, except for the model whose overall performance declined when the feature dimension changed from 64 to 128, the performance of the improved models at other locations increased. The reason for this phenomenon is that when φunit is located at a too marginal position in the network, the whole network will focus too much on acquiring the edge feature information, and the subsequent feature extraction layers will have difficulty finding the required information through too much edge information, which leads to the decline of the overall accuracy of the model. This also indirectly verifies the efficient information fusion ability of φunit module. To better demonstrate the advantages of the proposed φunit over the traditional information fusion modules, this experiment added different information fusion modules at the position where the feature dimension changed from 128 to 256 for comparative experiments. The experimental results show that the proposed φunit performs better than the traditional information fusion methods. This also proves that the module has significant advantages in both structure and feature information fusion mode. This further confirms the effectiveness and superiority of the module in improving the model performance.

4.2. Module Position and Number Exploration

As the arguments in the previous section demonstrated, the feature fusion module proposed in this paper had different performance effects in different positions of the model. Hence, in this section, this paper will carry out a series of investigations on the appropriate positions and numbers of the module to be added.

Based on the architecture of MobinetV2, this paper determines the basic feature dimension information change of "φNet" and inserts φunit modules at the locations where the dimensions change. Next, the relevant modules are gradually reduced and the accuracy comparison analysis is performed until a model that can achieve a high effect is found, namely φNet. Since the proposed model mainly relies on MobinetV2, we use MobinetV2 as a reference object for adjusting the model process. Figure 5 illustrates the trend of changes in the number of training iterations and accuracy resulting from different module locations in the experimental process, while Table 3 provides a detailed comparison of the experimental results.

Table 3. Comparison of accuracy after removing different positional φunit with Mobinetv2.

Erase	Acc. (%)	Compare Model (Mobinetv2)
-	72.31	
First unit	72.43	**72.76**
First and Last unit	**75.31**	

According to the experimental results of the relationship between φunit distance and network endpoint distance in Section 4.1, when φunit distance is too close to the network endpoint, the network tends to focus on shallow or deep feature information and ignore global feature information, thus reducing the final prediction accuracy of the model; Table 3 shows the results of the first and second experiments compared with the baseline model, which provide higher credibility for this theory. The orange and green curves in Figure 5 represent the results of the first and second experiments in Table 3, respectively. The blue curve shows the baseline model's corresponding experiments. The trend of the accuracy of the model during the training process indicates that: At the initial stage, the model focuses on extracting and analyzing shallow feature information, while the φunit test model added near the network endpoint has a slight advantage over the baseline model in terms of accuracy. Generally, as the training number increases, the model will gradually focus more and more on the comprehensive analysis of the overall feature information. However, in the later stage of training, if the model pays too much attention to shallow or deep feature information and neglects the analysis of the overall feature information, it will eventually

lead to the model failing to maintain the growth trend after reaching a certain accuracy threshold. Figure 5 shows that the test model's accuracy in the later stage is lower than that of the baseline model. After removing the φunit at the beginning and end of the model, this chapter further tested the improved model. Surprisingly, the test model significantly outperformed the MobinetV2 baseline model's accuracy performance, and whether it was in the training stage before or after training, the model could always surpass the baseline model in accuracy and maintain its leading position. The above results also indirectly confirmed two points: (1) the proposed φunit was not effective at all positions, only at the intermediate position that balanced shallow and abstract features; (2) the experiment also verified again that the proposed φunit had high information integration efficiency (still achieving superior experimental results under much fewer parameters than the baseline model). In conclusion, the model with the highest accuracy in the experiment was determined as φNet.

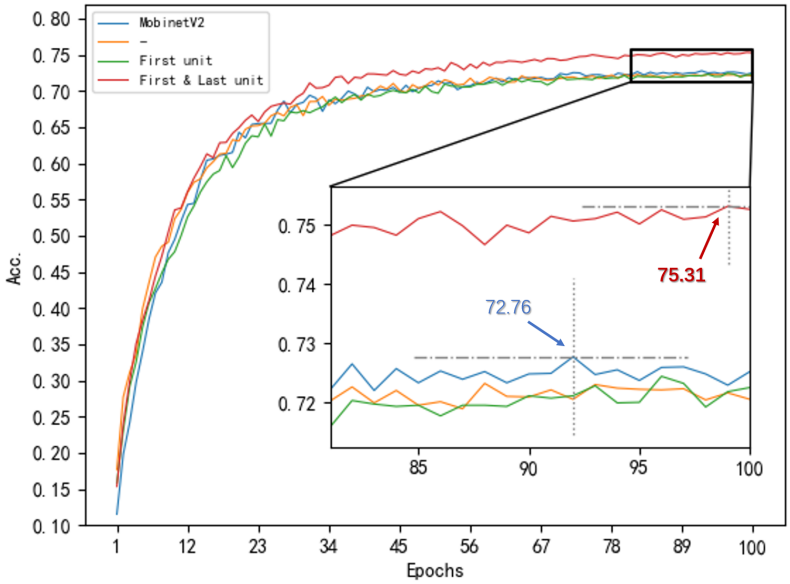


Figure 5. The accuracy of the model at various positions versus adding φunit.

This paper focuses on the placement of the connection modules in the proposed model, without delving into the details of the number of repeated modules per layer. Nevertheless, it is confident in this work that further research on optimizing the structure and quantity of this model will reveal its enormous scalability potential.

4.3. Comparative Validation of Model Performance

This paper uses the classic models in lightweight models as experimental subjects, and presents the comparison results in Table 4. The table data indicates that the model proposed in this paper achieves a comparable accuracy level with the optimal model with almost the lowest parameter amount. This benefit mainly stems from the novel information fusion module φunit proposed in this paper, which can fuse the preceding and following information more effectively, thereby improving the model performance. Meanwhile, this paper incorporates the design ideas of various outstanding models, while considering the requirements of lightweight models, and ultimately attains a good overall effect.

The model proposed in this paper has nearly the lowest parameter amount, which offers multiple benefits. On the one hand, the low parameter amount enables faster model training, because the model can converge to the optimal solution more quickly. On the other hand, the low parameter amount also lowers the memory usage of the model, which is crucial for resource-limited mobile devices. Moreover, the low parameter amount also boosts the inference speed of the model, which can perform predictions and inferences more rapidly in practical applications. These benefits are of great importance for lightweight

models on mobile devices. Mobile devices are often limited by computing resources and storage space, so the need for lightweight models is growing. This paper proposes an effective approach for developing lightweight models, which can keep high accuracy and satisfy the requirements of resource-limited environments. This paper validates the superiority of the model proposed in this paper through experiments and comparisons, and demonstrates its application potential in the field of lightweight models. This paper trusts that this research has a positive influence on advancing machine learning applications on mobile devices, and offers valuable guidance for future research and development [54].

Table 4. Comparative analysis of classical lightweight models on food101 in terms of classification accuracy, parameter count, and computational complexity.

Model	Params. (M)	FLOPs. (M)	Acc. (%)
Mobinetv3_small	2.54	**59.81**	60.50
Squeezenet	**1.23**	355.69	60.63
Mobilenetv3_large	5.48	227.71	66.57
Ghost	5.18	150.68	67.95
GhostV2	6.15	178.59	68.00
shufflenetv2_x1.5	3.50	303.67	69.65
Ours	1.24	225.91	71.64
Mobinetv2	2.22	318.96	**71.91**

5. Evaluation and Discussion

This paper investigates the inter-group connection problem in lightweight models, which has been neglected. Several classic models are analyzed and an efficient information fusion module φunit is proposed in this work, which can fuse the feature information from the current layer and the upper layer effectively, to alleviate the performance degradation problem caused by the discontinuity of feature information before and after the feature information fusion operation. The method achieves significant performance improvement with a small cost compared with traditional information fusion methods. In Section 4.1, this paper proves the stronger feature fusion and portability of φunit compared with the traditional feature information fusion modules from the test results through a series of ablation experiments and comparative experiments; in Section 4.2, through the position exploration experiment of φunit, it analyzes the different influences of the position of φunit in the model on the model performance, and then determines the position relationship of φunit in the model in the future practical application scenarios. Based on this, this experiment constructs φNet on the basis of verifying the effectiveness of the proposed module and summarizing the previous work. Based on φunit, we construct φNet, which reduces the parameter size and computation cost by nearly half and 30%, respectively, compared with the reference network. φNet can maintain high performance even when the number of classification categories increases, only 0.3% lower than the best model. This paper has room for further research on the relevant details of the model, such as the analysis of the influence degree of the number of repeated modules within each group on the model performance. This paper only explores one feasible scheme, and does not fully exploit the potential of φNet. Subsequent research can find the optimal configuration through grid search or automated search algorithm, in order to significantly improve the performance of the improved model. In addition, parameter tuning and network structure search also need to be further explored. By applying parameter optimization algorithm and network structure search algorithm, the performance and generalization ability of the model can be further enhanced in this work. These studies will contribute to a deeper understanding of the influence of model structure and parameter configuration, and provide more optimization insights and guidance for advancing lightweight models.

Author Contributions: Conceptualization, Z.L. and R.Z.; Designing experimental protocols, Z.L. and R.Z.; data analyses, Y.L., P.R. and X.Y. writing—original draft preparation Z.L. and S.C.; supervision R.Z. writing—review and editing, Z.L., R.Z., Y.L. and P.R.; revision of the article Y.L., P.R., X.Y. and S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key R&D Plan under Grant No. 2021YFF0601200 and 2021YFF0601204.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
SE	Squeeze-and-Excitation
CBAM	Convolutional Block Attention Module
ECA	Efficient Channel Attention
DFC	decoupled fully connected
LR	Learning Rate
WD	weight decay
MSA	Multi-Head Self-Attention
FC	Fully Connected

References

1. Zhang, X.; Li, S.; Li, X.; Huang, P.; Shan, J.; Chen, T. DeSTSeg: Segmentation Guided Denoising Student-Teacher for Anomaly Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 3914–3923.
2. Liu, Z.; Zhou, Y.; Xu, Y.; Wang, Z. Simplenet: A simple network for image anomaly detection and localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 20402–20411.
3. Jiang, H.; Dang, Z.; Wei, Z.; Xie, J.; Yang, J.; Salzmann, M. Robust Outlier Rejection for 3D Registration with Variational Bayes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 1148–1157.
4. Zhang, J.; Liu, R.; Shi, H.; Yang, K.; Reiß, S.; Peng, K.; Fu, H.; Wang, K.; Stiefelhagen, R. Delivering Arbitrary-Modal Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 1136–1147.
5. Ru, L.; Zheng, H.; Zhan, Y.; Du, B. Token contrast for weakly-supervised semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 3093–3102.
6. Jiang, J.; Zheng, N. MixPHM: Redundancy-Aware Parameter-Efficient Tuning for Low-Resource Visual Question Answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 24203–24213.
7. Yue, H.; Wang, K.; Zhang, G.; Feng, H.; Han, J.; Ding, E.; Wang, J. Cyclically disentangled feature translation for face anti-spoofing. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 3358–3366.
8. Liu, C.; Yu, X.; Tsai, Y.H.; Faraki, M.; Moslemi, R.; Chandraker, M.; Fu, Y. Learning to learn across diverse data biases in deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 4072–4082.
9. Wang, K.; Wang, S.; Zhang, P.; Zhou, Z.; Zhu, Z.; Wang, X.; Peng, X.; Sun, B.; Li, H.; You, Y. An efficient training approach for very large scale face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 4083–4092.
10. Tang, Z.; Qiu, Z.; Hao, Y.; Hong, R.; Yao, T. 3D Human Pose Estimation with Spatio-Temporal Criss-Cross Attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 4790–4799.

11. Wang, Z.; Nie, X.; Qu, X.; Chen, Y.; Liu, S. Distribution-aware single-stage models for multi-person 3D pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 13096–13105.
12. Weng, C.Y.; Curless, B.; Srinivasan, P.P.; Barron, J.T.; Kemelmacher-Shlizerman, I. Humannerf: Free-viewpoint rendering of moving people from monocular video. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 16210–16220.
13. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
14. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
15. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 818–833.
16. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**. arXiv:1409.1556.
17. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
18. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
19. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
20. Zagoruyko, S.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146.
21. Han, D.; Kim, J.; Kim, J. Deep pyramidal residual networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5927–5935.
22. Mahendran, N. Squeeze aggregated excitation network. *arXiv* **2023**, arXiv:2308.13343
23. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
24. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
25. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
26. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
27. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
28. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11534–11542.
29. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv* **2020**. arXiv:2010.11929.
30. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv* **2015**. arXiv:1510.00149.
31. Fang, G.; Ma, X.; Song, M.; Mi, M.B.; Wang, X. Depgraph: Towards any structural pruning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 16091–16101.
32. Lin, M.; Ji, R.; Wang, Y.; Zhang, Y.; Zhang, B.; Tian, Y.; Shao, L. Hrank: Filter pruning using high-rank feature map. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1529–1538.
33. Meng, F.; Cheng, H.; Li, K.; Luo, H.; Guo, X.; Lu, G.; Sun, X. Pruning filter in filter. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17629–17640.
34. Zhao, B.; Cui, Q.; Song, R.; Qiu, Y.; Liang, J. Decoupled knowledge distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 11953–11962.
35. Lin, S.; Xie, H.; Wang, B.; Yu, K.; Chang, X.; Liang, X.; Wang, G. Knowledge distillation via the target-aware transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 10915–10924.
36. Dong, P.; Li, L.; Wei, Z. Diswot: Student architecture search for distillation without training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 11898–11908.
37. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style ConvNets Great again. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 13728–13737.

38. Choi, K.; Lee, H.Y.; Hong, D.; Yu, J.; Park, N.; Kim, Y.; Lee, J. It's All In the Teacher: Zero-Shot Quantization Brought Closer to the Teacher. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 8311–8321.
39. Yamamoto, K. Learnable companding quantization for accurate low-bit neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 5029–5038.
40. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1580–1589.
41. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
42. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6848–6856.
43. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, South Korea, 27 October–2 November 2019; pp. 1314–1324.
44. Tang, Y.; Han, K.; Guo, J.; Xu, C.; Xu, C.; Wang, Y. GhostNetv2: Enhance cheap operation with long-range attention. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 9969–9982.
45. Touvron, H.; Bojanowski, P.; Caron, M.; Cord, M.; El-Nouby, A.; Grave, E.; Izacard, G.; Joulin, A.; Synnaeve, G.; Verbeek, J.; et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 5314–5321. [CrossRef] [PubMed]
46. Park, N.; Kim, S. How do vision transformers work? *arXiv* **2022**, arXiv:2202.06709.
47. Zeiler, M.D.; Krishnan, D.; Taylor, G.W.; Fergus, R. Deconvolutional networks. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2528–2535.
48. Dumoulin, V.; Visin, F. A guide to convolution arithmetic for deep learning. *arXiv* **2016**, arXiv:1603.07285.
49. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307. [CrossRef] [PubMed]
50. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
51. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
52. Wu, B.; Wan, A.; Yue, X.; Jin, P.; Zhao, S.; Golmant, N.; Gholaminejad, A.; Gonzalez, J.; Keutzer, K. Shift: A zero flop, zero parameter alternative to spatial convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 9127–9135.
53. Loshchilov, I.; Hutter, F. Fixing Weight Decay Regularization in Adam. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26 April–1 May 2020.
54. Kumar, N.; Hashmi, A.; Gupta, M.; Kundu, A. Automatic diagnosis of Covid-19 related pneumonia from CXR and CT-Scan images. *Eng. Technol. Appl. Sci. Res.* **2022**, *12*, 7993–7997. [CrossRef]

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.