MH-LViT: Multi-path Hybrid Lightweight ViT Models with Enhancement Training

Yating Li[1], Wenwu He[2], Shuli Xing[3], Hengliang Zhu[4]
School of Computer Science and Mathematics, Fujian University of Technology, Fuzhou 350118, China[1,2,3,4]
Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fuzhou 350118, China[2,3,4]

Abstract—**Vision Transformers (ViTs) have become increasingly popular in various vision tasks. However, it also becomes challenging to adapt them to applications where computation resources are very limited. To this end, we propose a novel multi-path hybrid architecture and develop a series of lightweight ViT (MH-LViT) models to balance well performance and complexity. Specifically, a triple-path architecture is exploited to facilitate feature representation learning that divides and shuffles image features in channels following a feature scale balancing strategy. In the first path ViTs are utilized to extract global features while in the second path CNNs are introduced to focus more on local features extraction. The third path completes the representation learning with a residual connection. Based on the developed lightweight models, a novel knowledge distillation framework IntPNKD (Normalized Knowledge Distillation with Intermediate Layer Prediction Alignment) is proposed to enhance their representation ability, and in the meanwhile, an additional Mixup regularization term is introduced to further improve their generalization ability. Experimental results on benchmark datasets show that, with the multi-path architecture, the developed lightweight models perform well by utilizing existing CNN and ViT components, and with the proposed model enhancement training methods, the resultant models outperform notably their competitors. For example, on dataset miniImageNet, our MH-LViT_M3 improves the top-1 accuracy by 4.43% and runs 4x faster on GPU, compared with EdgeViT-S; on dataset CIFA10, our MH-LViT_M1 improves the top-1 accuracy by 1.24% and the enhanced version MH-LViT_M1* by 2.28%, compared to the recent model EfficientViT_M1.**

Keywords*—*Multi-path hybrid; lightweight ViT; normalized knowledge distillation; Mixup regularization

I. Introduction

In recent years, Vision transformers (ViTs) [1] have received increasing attention in many visual tasks, achieving remarkable results in tasks such as image classification [1], [2], [3], target detection [4], [5], [6], [7] and semantic segmentation [8], [9], [10], [11], [12], [13]. However, the computational overhead of the self-attention mechanism makes ViTs less efficient than Convolutional Neural Networks (CNNs) [14] on memory and computationally constrained devices. The huge model size and computational cost make it challenging to adapt them to real-time applications. Therefore, researchers tend to build lightweight and efficient ViT models.

Some approaches aim to build lightweight versions of ViT models by reducing the number of feature channels or self-attentive heads. Nevertheless, such approaches usually lead to significant performance degradation. The author in [15] successfully improves the performance of existing tiny ViTs by introducing a plugin that groups and shuffles feature channels. Alternatively, some researchers have attempted to improve model performance through specific designs, such as combining computationally expensive self-attention with efficient convolutional operations to create hybrid efficient ViTs [16], [17], [18], [19], [20]. Among them, MobileViT [16] combines the image-specific inductive bias of CNNs and the global information processing capability of ViTs to encode both local and global information efficiently. EdgeViT [19] combines the attention mechanism and CNNs through the implementation of local-global-local (LGL) blocks. Nevertheless, most of these methods serially stack self-attention and convolutional layers, and the extraction of global features often compromises previously extracted local features, failing to take full advantage of global and local features. To overcome this problem, several studies have begun to explore the application of parallel structures in feature extraction. The parallel structure allows the self-attention and convolutional layers to work simultaneously and extracts global and local features independently, and then fuses them in some way. This structure can maintain the integrity of local features while combining global features to achieve a more comprehensive feature representation. For example, TransXNet [20] efficiently extracts and fuses global and local features by combining self-attention and convolution in parallel to achieve excellent performance.

Therefore, this paper adopts a parallel structure approach to extract both global and local features, fully exploiting the diversity of features to enhance the model performance. Unlike previous parallel structures, we have added an additional residual branch to enhance the learning ability of feature representation, without explicitly increasing parameters or computational overhead. Specifically, we exploit a feature scale balancing strategy to divide input features into three parts. The first one is fed into the Transformer branch to extract global features, the second one is fed into the CNN branch to extract local features, and the last one is directly used to form the Residual branch. Subsequently, the feature fusion module is employed to shuffle extracted features and balance their chances to be processed in different branches. Based on this, we conclude that the tiny model has strong features representation ability. To further release their representation potential, we propose a new knowledge distillation framework, IntPNKD (Normalized Knowledge Distillation with Intermediate Layer Prediction Alignment), to effectively improve the inference performance without additional inference cost. In addition to the standard knowledge distillation procedure of NKD (Normalized Knowledge Distillation) [21], IntPNKD aligns the predictions made respectively on intermediate feature maps of the teacher and the student. In the meanwhile, an image mixing regularizer is introduced to further enhance the generalization ability of the resultant model. The main contributions of this paper are summarized as follows:

- We propose a multi-path hybrid architecture to design lightweight ViT model (MH-LViT), where a triple-branch architecture (transformer, CNN, and residual) is employed to extract efficiently local and global features, and the features extracted by multiple branches are further shuffled and re-assigned with a feature fusion module.
- We propose a new knowledge distillation framework IntPNKD to improve the representation ability of MH-LViT, which is further enhanced by introducing an image mixing regularization term. The achieved performance improvement costs nothing in inference.
- We develop a series MH-LViT models with various sizes that are tested on multiple benchmark datasets. The experimental results show that our models balance well the efficiency and the accuracy.

II. Related Work

A. Efficient Vision Transformers

In order to reduce the number of parameters and the computational overhead, a series of works on efficient vision transformers have been proposed, which cover a wide range of interesting ideas such as lightweight module design, model compression, token compression, hybrid model design, and so on. Window-based Self-Attention ViT proposed in Swin [4] reduces the computation cost of each transformer block by dividing the feature map into windows and restricting the attention operation within a local window. In this way, the length of input sequence fed into Transformer is reduced, leading to improved efficiency. Wang et al. [22] proposed a spatial-reduction attention (SRA) to reduce the computation cost of self-attention, by downsampling the spatial resolution dimensions of the input Query and Key branches via a lightweight depthwise convolution. Token merging [23] achieves parameter compression by merging tokens with large semantic similarity. CVT [24] and LeViT [17] insert some convolutional layers into Transformer layers, to downsample feature maps and perform local information fusion, where the Transformer layers are used to capture the global information from deep features. Liu et al. [25] proposed a cascade group attention module that uses only three or four attention heads, to increase the diversity of features while reducing computational redundancy in multi-head attention.

B. Efficient Convolutional Neural Networks

Actually, before the arrival of ViT, models based on Convolutional Neural Networks (CNNs) faced the same challenge to deploy them to devices with limited computational resources. To fix this, a series of elegant work have been proposed. MobileNets [26], [27] leverage depthwise separable convolutions to reduce the computation complexity. ShuffleNet [28] utilizes pointwise group convolution and performs channel shuffle operations to reduce the computation cost. IDConv[20] reduces the computation overhead by utilizing dynamic deep convolutions and adaptive average pooling. Overall, efficient CNNs work well in a variety of scenarios which inspires us to develop a novel model that combines efficient CNNs and ViT in some way, to extract effectively both global and local features.

C. Knowledge Distillation

Knowledge distillation (KD) has attracted wide attention in the field of model compression, which typically transfers knowledge from a teacher (model) to a student (model) to improve the latter's performance. This framework, originally proposed by Hinton et al. [29], utilizes both the hard labels of ground truth and the soft ones provided by the teacher to guide the learning process of the student. Recently, there have emerged studies [30], [21] focused on KD in ViT. For instance, DeiT [30] introduces a novel distillation procedure which relies on a distillation token ensuring that the student learns from the teacher through attention, to achieve efficient transformers with competitive top-1 accuracy on ImageNet. MixSKD [31] is a self-knowledge distillation framework that enables the network to learn cross-image knowledge by modeling supervisory signals from mixup images. The most relevant work on knowledge distillation to ours is NKD [21], which uses cross-entropy to align respectively the probability distributions of target class and non-target classes. We build our KD framework on NKD and introduce Middle Feature Prediction Alignment into it for better knowledge transfer.

III. Preliminaries

A. EfficientViT

Unlike the classic ViT [1], EfficientViT [25] adopts a sandwich layout and a cascade group attention. Specifically, the cascade group self-attention layer Φ_i^A is sandwiched between two FFN (Feed Forward Network) layers each of them is denoted with Φ_i^F. The operation of the i-th block can be formulated as:

$$X_{i+1} = \Phi_i^{\mathrm{F}}(\Phi_i^{\mathrm{A}}(\Phi_i^{\mathrm{F}}(X_i))), \tag{1}$$

where X_i is the input feature of the i-th block. In the the cascade group self-attention layer Φ_i^A, the input feature X_i is divided into J parts which are fed into J attention heads, where the output of each head is added to the subsequent heads to enrich the feature information. Formally, the operation of the j-th attention head can be expressed as:

$$\mathring{X}_i^j = \begin{cases} \mathrm{Attn}(X_i^j), & if\ j = 0, \\ \mathrm{Attn}(X_i^j + \mathrm{Attn}(\mathring{X}_i^{j-1})), & if\ j \geq 1, \end{cases} \tag{2}$$

where, X_i^j denotes the j-th split of input X_i and $\mathring{X}_i^j$ is its corresponding attention output. Then J attention outputs are concatenated:

$$\hat{X}_i = \Phi_i^A(X_i) = \mathrm{concat}[\{\tilde{X}_i^j\}_{j=1}^J], \tag{3}$$

where, $\hat{X}_i$ is the output of the i-th cascade group attention layer.

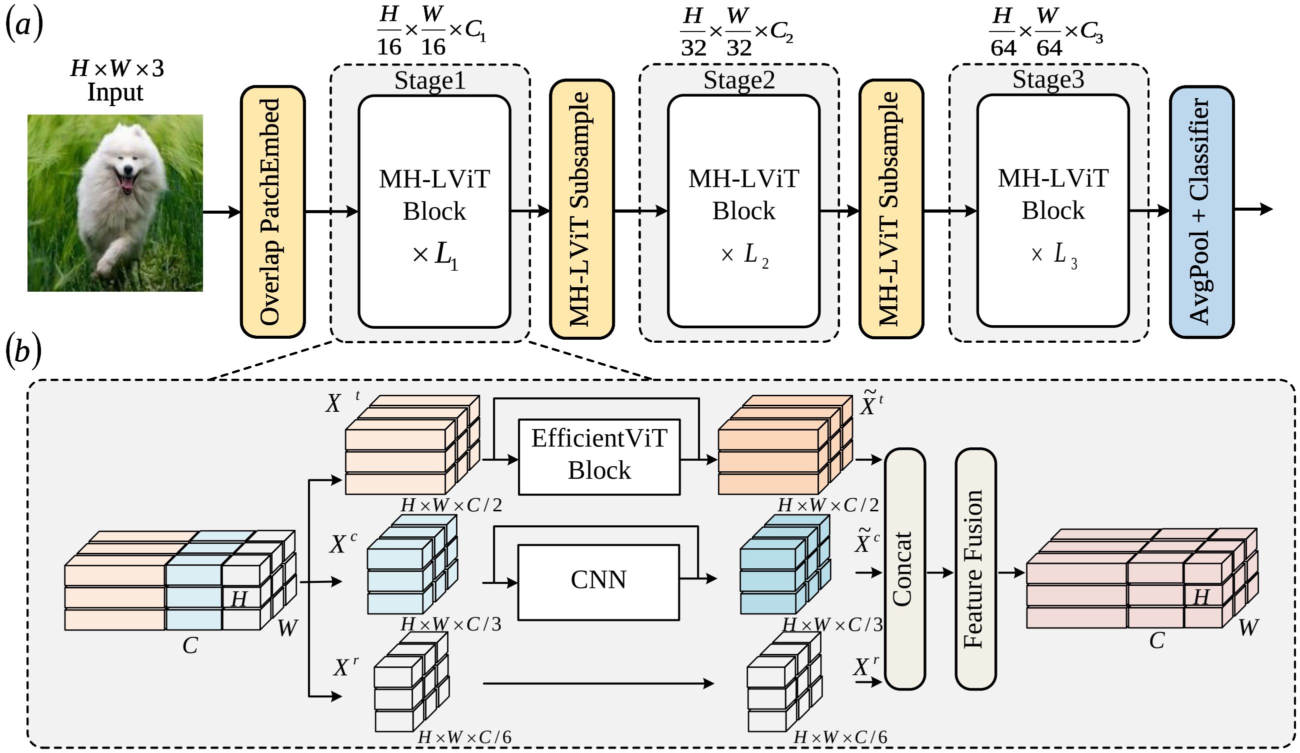

Fig. 1. Architecture of MH-LViT: (a) Overall architecture of MH-LViT; (b) Multi-Path branch block.

B. Normalized Knowledge Distillation

Normalized Knowledge Distillation (NKD) [21] is an improvement version of the decoupled knowledge distillation (DKD) [32] that normalizes the non-target logits and utilizes Cross-Entropy (CE) instead of Kullback-Leibler (KL) divergence to align the target and non-target class distributions respectively. In particular, the loss of NKD can be formulated as follows:

$$L_{NKD} = -T_t log(S_t) - \gamma \cdot \lambda^2 \cdot \sum_{k \neq t} \mathcal{N}(T_k^{\lambda}) log(\mathcal{N}(S_k^{\lambda})), \quad (4)$$

where, the index t denotes the target class, T_k (S_k) denotes the output probabilities of teacher (student) model corresponding to the k-th class, λ is the KD temperature [29], γ is the hyper-parameter to balance the two items in the loss, and $\mathcal{N}(*)$ denote the normalization operation.

C. Mixup

Mixup [33] generates a mixed image (x_{ab}, y_{ab}) by linearly combining a pair of original images $\{x_a, y_a; x_b, y_b\} \in D$, where D denote a data set, x_a (x_b) denotes the image a (b) and y_a (y_b) is its corresponding label. The mixed image and its corresponding label are formulated as follows:

$$\begin{aligned} x_{ab} &= \lambda x_a + (1 - \lambda) x_b, \\ y_{ab} &= \lambda y_a + (1 - \lambda) y_b, \end{aligned} \quad (5)$$

where, the combination is controlled by the mixing factor λ sampled from the beta distribution.

IV. Methods

A. Overview

We propose a new multi-path hybrid architecture to design lightweight ViT model MH-LViT. The overall architecture of MH-ViT is shown in Fig. 1(a). The model adopts a hierarchical architecture, that effectively reduces the resolution of feature maps during the forward propagation process, while gradually increases the number of channels for feature map. Specifically, MH-LViT contains three stages, each of which consists of L MH-LViT blocks. In order to reduce the amount of parameters and computation overhead, we divide the input features by channels in the ratio of $3:2:1$ via the feature scale balancing strategy, and then learn images' representation in parallel through three paths, i.e. Transformer, CNN and Residual. By introducing the multi-hybrid structure, the model can capture global information and local details at the same time, to understand well the images for latter computer vision (CV) tasks. In addition, a new type of knowledge distillation and a mixup regularization are exploited to further improve the inference performance without any additional inference cost.

B. Lightweight Model Design

1) Multi-Path branch: As shown in Fig. 1(b), the MH-LViT Block is mainly composed of a three-branch architecture and a feature fusion module. In particular, the three branches includes an efficient Transformer (EfficientViT Block), a lightweight CNN (IDConv) and a residual connection. Formally, the operations in MH-LViT Block can be written as follows:

$$X_i^t, X_i^c, X_i^r = \mathrm{Split}(X_i),\ X_i^t : X_i^c : X_i^r = 3:2:1; \tag{6}$$

$$\tilde{X}_i^t = \mathrm{EfficientViTBlock}(X_i^t); \tag{7}$$

$$\tilde{X}_i^c = \mathrm{IDConv}(X_i^c); \tag{8}$$

$$\tilde{X}_i = \mathrm{Concat}(\tilde{X}_i^t, \tilde{X}_i^c, X_i^r); \tag{9}$$

$$X_{i+1} = \mathrm{FF}(\tilde{X}_i). \tag{10}$$

Here, with a little notation abuse, X_i denotes the input feature of the i-th MH-LViT block and X_{i+1} the corresponding output. $Split(\cdot)$ denotes the operation to divide the input feature into three splits with the specified ratio and : denotes the ratio of the number of channels.

2) Transformer branch: It is particularly important to capture effective global information of images for CV tasks. To this end, we introduce the cutting-edge EfficientViT [25] to our model to extract global features. EfficientViT not only inherits powerful capabilities of the vanilla ViT model, but also significantly improves the computational efficiency via a sandwich layout and a cascade group attention. In addition, the depthwise convolution (DWConv) is used for information fusion before the final FFN layer of the block. The part of input features X_i^t is fed into the EfficientViT branch, to learn the representations of images base on the global information. The processing of X_i^t in an EfficientViT Block can be expressed as follows:

$$\tilde{X}_i^t = \mathrm{FFN}(\mathrm{DW}((\Phi_i^A(\mathrm{FFN}(\mathrm{DW}(X_i^t))))))), \tag{11}$$

where, $DW(\cdot)$ denotes the depthwise convolution and other notations denote the same operation as indicated before.

3) CNN branch: In order to inject inductive bias for local feature extraction, this paper introduces IDConv (Input-dependent Depthwise Convolution) [20] to extract image local information. IDConv can dynamically generate convolution kernels, which enhances the adaptability and characterization of CNN for different data features. Firstly, the spatial dimension of input features $X_i^c \in C/3{\times}H{\times}W$ is compressed to K^2 by using adaptive pooling, to aggregate the spatial context information. Subsequently, two consecutive $1{\times}1$ convolutional layers are utilized to generate the attention map $A' \in (G{\times}C/3){\times}K^2$, where G represents the number of attention groups. Then, A' is reshaped to $G{\times}C/3{\times}K^2$, where a softmax operation is applied in the G-dimension to generate the attention weights $A \in G{\times}C/3{\times}K^2$. Finally, the attention weight A is element-wise multiplied with a set of learnable parameters $P \in G{\times}C/3{\times}K^2$ and summed along the G-dimension to obtain the input-dependent deep convolution kernel $W \in C/3{\times}K^2$. This whole process can be expressed as follows:

$$A_i' = \mathrm{Conv}(\mathrm{Conv}(\mathrm{AdaptivePool}(X_i^c))); \tag{12}$$

$$A_i = \mathrm{Softmax}(\mathrm{Reshape}(A_i')); \tag{13}$$

$$W_i = \sum_{g=1}^{G} (P_i)_g (A_i)_g; \tag{14}$$

$$\tilde{X}_i^c = W_i X_i^c. \tag{15}$$

4) Residual branch: As we know, in CV models, the information of original features may be weakened or lost when a series of transformation operations have been performed on them. To further enrich the feature representation and to ensure that the model can make full use of the original information, we preserve part of original features, i.e. X_i^r, and do not perform any transforming or processing on these features. In this way, we include both processed and unprocessed raw features to enhance the learning ability of feature representation, without explicit increasing in parameters or computation overhead.

5) Fusion module: For the features extracted through multiple paths, we introduce a feature fusion module to shuffle them, to let features in each branch get the chance to be fed into other branches. The feature fusion module mainly implements channel shuffle, aiming to shuffle the limited interaction between branches by shuffling features from different branches. This mixes the features from different branches, thereby achieving information fusion. Specifically, the global features extracted through Transformer Branch, the local features through CNN Branch, and the features for residual connection are first concatenated and then input into the fusion module, where a channel shuffle operation is utilized to shuffle inputs. In particular, the input feature tensor $\tilde{X}_i$ is grouped into a certain number of groups by the channel-dimension, and then the channels within each group are rearranged to obtain the shuffled tensor $X_i^{shuffle}$. Finally, $X_i^{shuffle}$ is restored to the original shape to obtain the fused feature tensor X_{i+1}. This module effectively mixes the features extracted from different branches, to avoid features in one branch are locked within this branch. Formally, the operations in FF module can be written as follows:

$$\tilde{X}_i^{group} = \mathrm{Split}_{\mathrm{group}}(\tilde{X}_i); \tag{16}$$

$$\tilde{X}_i^{shuffle} = \mathrm{Shuffle}(\tilde{X}_i^{group}); \tag{17}$$

$$X_{i+1} = \mathrm{Reshape}(\tilde{X}_i^{shuffle}). \tag{18}$$

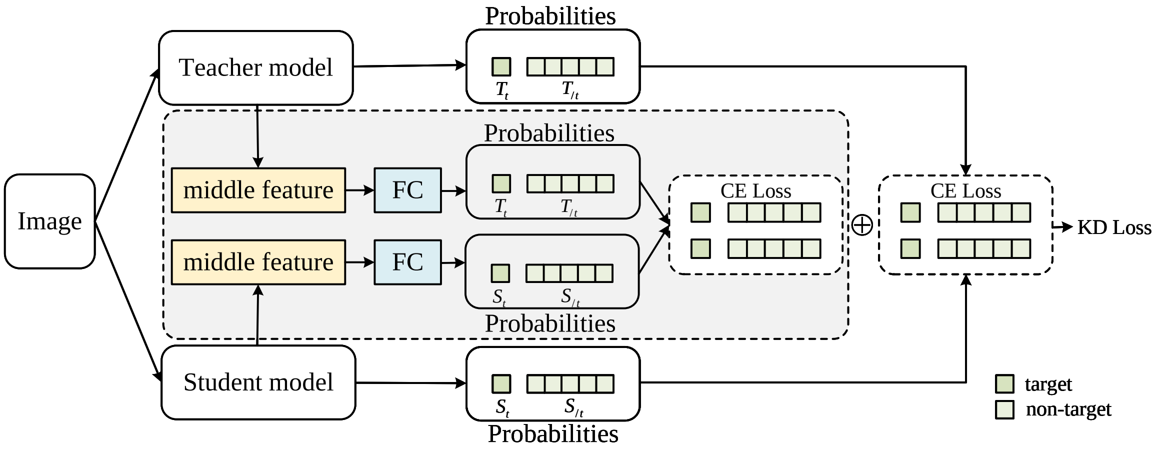

Fig. 2. Illustration of IntPNKD (Normalized knowledge distillation with intermediate layer prediction alignment).

C. Lightweight Model Enhancements

1) IntPNKD: Limited to its lightweight, the developed tiny model inevitably sacrifice its performance to some degree. To this end, we further propose a new distillation framework IntPNKD to improve its representation ability. Specifically, as shown in Fig. 2, IntPNKD transfers knowledge from the teacher (standard model) to the student (lightweight model), by aligning not only the output logits of two models (as vanilla NKD does) but also the predictions made on intermediate layers' feature maps. While the traditional knowledge distillation (NKD) focuses on aligning the final output logits, it may overlook important feature information encoded in the intermediate layers, which plays a critical role in the overall learning process. With the additional alignment, IntPNKD improves the effectiveness of knowledge distillation from teacher to student. Formally, let X^T_{mid} (X^S_{mid}) denote the features of the intermediate layer of teacher (student) model, $\text{FC}(\cdot)$ denote the fully connected layer and $\text{P}(\cdot)$ the softmax operation. Then the total distillation loss of IntPNKD can be written as:

$$L_{IntPNKD} = L_{NKD} + \text{KL}(\text{P}(\text{FC}(X^T_{mid})), \text{P}(\text{FC}(X^S_{mid}))). \tag{19}$$

2) Mixup regularization: Following the idea of image mixing in MixSKD [31], we introduce a regularizer L_{MR} to our case (the tiny student model), to enhance the generalization ability of the developed model. As shown in Fig. 3, we mix randomly two original images, e.g. x_a and x_b, to obtain a mixup one x_{ab}, and expect that the probability distribution output by the model on x_{ab} and the one given by mixing the logits of x_a and x_b are not too far from each other. In particular, the KL divergence is utilized to form the regularization term, to guide the model make relatively stable predictions on mixup image and original ones, leading to improved inference performance. The regularization term can be formulated as:

$$L_{MR} = \text{KL}(\text{P}^{\text{s}}(X_a, X_b), \text{P}^{\text{s}}(X_{ab})), \tag{20}$$

$$\text{P}^{\text{S}}(X_a, X_b) = \text{Softmax}(\lambda \text{S}(X_a) + (1-\lambda)\text{S}(X_b)), \tag{21}$$

$$\text{P}^{\text{S}}(X_{ab}) = \text{Softmax}(\text{S}(X_{ab})), \tag{22}$$

where, $\text{P}(\cdot)$ denotes the softmax operation, $S(\cdot)$ the logit (before softmax operation) output by the tiny student model, and λ is the mixing factor as in Eq. (5).

Combining everything together, the total loss function L used to train the lightweight model can be written as:

$$L = L_{CE} + L_{IntPNKD} + L_{MR}, \tag{23}$$

where, L_{CE} is the cross-entropy loss guided by the ground truth.

V. Experiments

In this section, we perform experiments on several benchmark datasets, to validate the effectiveness of proposed methods. We first elaborate on the implementation details of the experiments and then present the main experimental results of the proposed models and the relevant baseline models, which are discussed in depth. To further dissect the performance of proposed model, we also conduct a series of ablation studies to evaluate the practical effects of its key components.

A. Implementation Details

We conduct image classification experiments on three benchmark datasets, i.e. CIFAR10 [34], CIFAR100 [34] and miniImageNet [35]. In building the models, we used two tool libraries, i.e. PyTorch 1.11.0 [36] and Timm 0.5.4 [37]. The AdamW [38] optimizer and the cosine learning rate scheduler are used to train related models, each of which is trained 300 epochs from scratch on an Nvidia A100 GPU. For the input images, we resize and randomly crop them to 224 × 224 pixels.

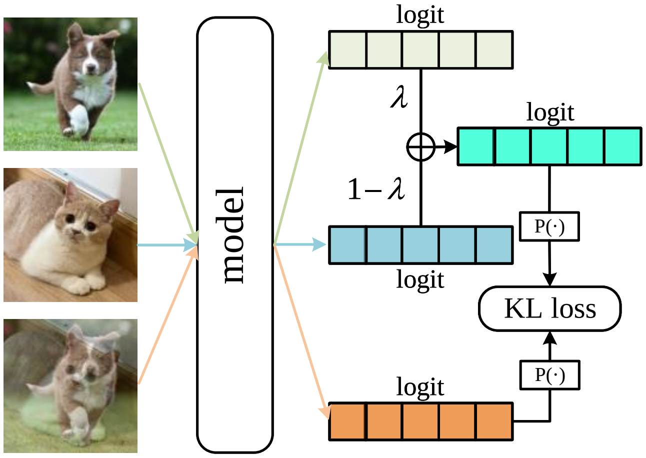

Fig. 3. Illustration of mixup regularization.

The batch size is fixed to 64, the initial learning rate is set to 1×10^{-3}, and the weight decay is 0.01. In addition, we use the same data augmentation strategies as in[30], including auto-augmentation [39] and random erasing [40]. We evaluate the model with the size, the top-1 accuracy, the throughput and Flops to get a full picture of its performance. As for KD, we utilize recently proposed CARTE-B [41] as the teacher model, which performs well with a medium model size by pre-training on ImageNet.

B. Main Results

We compare the developed MH-LViT with popular efficient models based on CNN or ViT and report the results. The results show that MH-LViT models with different sizes achieve the best accuracy and speed tradeoff across benchmark datasets in most cases.

1) Results on miniImageNet: Table I summarizes the results on the dataset miniImageNet achieved by the proposed lightweight models and multiple SOTA competitors. We first compare MH-LViT with tiny models whose size are close to it. As can be seen from the table, MH-LViT_M1 outperforms EfficientViT_M1 and EdgeViT-XXS by up to 5.57% and 6.07% respectively, with a comparable number of parameters. Compared with ShuffleNetV2 2.0x, MH-LViT_M2 improves the top-1 accuracy by 1.46% with fewer parameters and faster inference speed. Compared with EdgeViT-S, MH-LViT_M3 improves the top-1 accuracy by 1.74% and runs 4x faster on GPU. More interestingly, the enhanced version MH-LViT* outperforms all competitors in terms of accuracy. We can observe from Table I that, MH-LViT_M3* achieves 81.50% top-1accuracy on this dataset, which outperforms EdgeViT-S by up to 7.41% with a comparable number of parameters. Compared with EdgeViT-S, MH-LViT_M2* improves the top-1 accuracy by 4.19% with lower parameters and Flops. MH-ViT with various model sizes perform well on dataset miniImageNet and the results show that the multiple-path design is effective and model enhancement training strategy even goes an extra mile.

2) Results on CIFAR100: Table II shows in detail the performance of proposed models and its competitors on dataset CIFAR-100. For example, MH-LViT_M1 improves the top-1 accuracy by 2.93% compared to EfficientViT_M1. MH-LViT_M2 improves the top-1 accuracy by 2.32% and runs 3x faster on the GPU, compared to EdgeViT-XS that has a comparable number of parameters. Compared to ShuffleNetV2 2.0x that achieves the best top-1 accuracy on this dataset, MH-LViT_M2 provides a competitive top-1 accuracy but has a 1.8x throughput. MH-LViT_M2* even goes further and can beat ShuffleNetV2 2.0x with 4.01% improvement in top-1 accuracy. Compared to models with higher throughput such as EfficientViT_M3, MH-LViT_M2 improves its top-1 accuracy by 3.32% while maintaining a similar throughput. When model size goes large, MH-LViT_M3* outperforms EdgeViT-S in top-1 accuracy by 5.54% and runs 4.5x faster than it on GPU.

3) Results on CIFAR10: Table III summarizes experimental results of related models on dataset CIFA10. As what found in Tables I and II, from Table III one can find similar performance advantages of MH-LViT models over their competitors. For example, MH-LViT_M1 runs 5.2x faster than EdgeViT-XXS while improves the top-1 accuracy by 0.39%. The enhanced version MH-LViT_M1* achieves up to 96.28% top-1 accuracy on CIFA10 with 3.1M parameters. Compared with EdgeViT-XS model, MH-LViT_M2* runs 3.7x faster on the GPU with a comparable number of parameters and top-1 accuracy. MH-LViT_M3* achieves the highest top-1 accuracy of 96.75% with 12.5M parameters.

Overall, experimental results on bench mark datasets validate that the developed lightweight models with multiple-path design work well and can strike a good balance between the efficiency and the accuracy. The proposed model enhancement training strategy is effective and can provide further significant accuracy improvement without any additional inference cost.

C. Ablation Study

In this section, we validate the effectiveness of main components of MH-LViT/MH-LViT*, such as Multi-Path Branch, feature fusion module, IntPNKD, and mixup regularization, by

TABLE I. MH-LViT Image Classification Performance on MiniImageNet with Comparisons to State-of-the-Art Efficient CNN and ViT

Model	Params(M)↓	ACC_Top1 (%)↑	Throughput(images/s)↑	Flops(M)↓	Input	Epoch
ShuffleNetV2 0.5x	1.3	61.58	34012	44	224	300
MobileViT-XXS	1.3	45.66	4456	273	224	300
ShuffleNetV2 1.0x	2.3	73.19	5454	152	224	300
MobileViT-XS	2.3	55.12	3344	744	224	300
MobileNetV3-Small	2.5	69.82	9031	65	224	300
EfficientViT_M1	3.0	69.96	20093	167	224	300
MH-LViT_M1	3.1	75.53	19126	130	224	300
MH-LViT_M1*	3.1	**76.44**	19126	130	224	300
EdgeViT-XXS	4.1	69.46	3638	546	224	300
MobileNetV3-Large	5.4	74.01	7920	271	224	300
MobileViT-S	5.6	73.59	1939	1464	224	300
EdgeViT-XS	6.7	73.20	3852	1123	224	300
MH-LViT_M2	6.7	76.76	14325	377	224	300
MH-LViT_M2*	6.7	**78.28**	14325	377	224	300
EfficientViT_M3	6.9	68.87	16644	263	224	300
ShuffleNetV2 2.0x	7.4	75.30	7540	596	224	300
LeViT-128	9.2	65.42	10905	371	224	300
LeViT-192	10.9	67.65	8837	605	224	300
EdgeViT-S	11.1	74.09	2274	1897	224	300
EfficientViT_M5	12.4	70.98	10621	522	224	300
MH-LViT_M3	12.5	78.52	10347	452	224	300
MH-LViT_M3*	12.5	**81.50**	10347	452	224	300
LeViT-256	18.9	68.88	6494	1059	224	300
LeViT-384	39.1	69.45	3883	2250	224	300

TABLE II. MH-LViT Image Classification Performance on Cifar100 with Comparisons to State-of-the-Art Efficient CNN and ViT

Model	Params(M)↓	ACC_Top1 (%)↑	Throughput(images/s)↑	Flops(M)↓	Input	Epoch
ShuffleNetV2 0.5x	1.3	72.16	34012	44	224	300
MobileViT-XXS	1.3	64.38	4456	273	224	300
ShuffleNetV2 1.0x	2.3	76.19	5454	152	224	300
MobileViT-XS	2.3	75.21	3344	744	224	300
MobileNetV3-Small	2.5	71.21	9031	65	224	300
EfficientViT_M1	3.0	73.46	20093	167	224	300
MH-LViT_M1	3.1	76.39	19126	130	224	300
MH-LViT_M1*	3.1	**80.19**	19126	130	224	300
EdgeViT-XXS	4.1	66.79	3638	546	224	300
MobileNetV3-Large	5.4	71.74	7920	271	224	300
MobileViT-S	5.6	75.92	1939	1464	224	300
EdgeViT-XS	6.7	75.45	3852	1123	224	300
MH-LViT_M2	6.7	77.80	14325	377	224	300
MH-LViT_M2*	6.7	**81.78**	14325	377	224	300
EfficientViT_M3	6.9	74.48	16644	263	224	300
ShuffleNetV2 2.0x	7.4	77.77	7540	596	224	300
LeViT-128	9.2	69.17	10905	371	224	300
LeViT-192	10.9	71.08	8837	605	224	300
EdgeViT-S	11.1	76.61	2274	1897	224	300
EfficientViT_M5	12.4	74.24	10621	522	224	300
MH-LViT_M3	12.5	78.35	10347	452	224	300
MH-LViT_M3*	12.5	**82.15**	10347	452	224	300
LeViT-256	18.9	71.14	6494	1059	224	300
LeViT-384	39.1	72.17	3883	2250	224	300

performing an ablation study on the dataset CIFAR100. The experimental results are summarized in Tables IV to VI.

1) Multi-Path branch: As shown in Table IV, we remove respectively the CNN Branch, the Transformer Branch, and the Residual Branch but keep the same model size as the complete one, to verify their effectiveness. As can be seen from the table, when the CNN Branch is removed, top-1 accuracy of MH-LViT_M1 decreases by 1.56%, which shows that CNNs in MH-ViT is helpful to enhance the leaning representation ability. When the Transformer Branch is removed, the top-1 accuracy decreases by 4.43%, which validates well that Transformer blocks are crucial for our model. Meanwhile, the Residual Branch also plays an auxiliary role in improving the proposed model, and when it is removed, the model accuracy decreases by 0.55%.

2) Feature fusion module: The ablation experimental result of Feature Fusion Module is also reported in Table IV, where we can observe a decrease of 0.9% in top-1 accuracy when FF module is removed from MH-LViT_M1. This result reveals the important auxiliary role of FF module in the developed model.

TABLE III. MH-LViTImage Classification Performance on Cifar10 with Comparisons to State-of-the-Art Efficient CNN and ViT

Model	Params(M)↓	ACC_Top1 (%)↑	Throughput(images/s)↑	Flops(M)↓	Input	Epoch
ShuffleNetV2 0.5x	1.3	91.84	34012	44	224	300
MobileViT-XXS	1.3	94.57	4456	273	224	300
ShuffleNetV2 1.0x	2.3	94.50	5454	152	224	300
MobileViT-XS	2.3	88.58	3344	744	224	300
MobileNetV3-Small	2.5	94.66	9031	65	224	300
EfficientViT_M1	3.0	94.0	20093	167	224	300
MH-LViT_M1	3.1	95.24	19126	130	224	300
MH-LViT_M1*	3.1	**96.28**	19126	130	224	300
EdgeViT-XXS	4.1	94.85	3638	546	224	300
MobileNetV3-Large	5.4	95.56	7920	271	224	300
MobileViT-S	5.6	95.11	1939	1464	224	300
EdgeViT-XS	6.7	95.52	3852	1123	224	300
MH-LViT_M2	6.7	95.57	14325	377	224	300
MH-LViT_M2*	6.7	**96.67**	14325	377	224	300
EfficientViT_M3	6.9	94.51	16644	263	224	300
ShuffleNetV2 2.0x	7.4	95.15	7540	596	224	300
LeViT-128	9.2	94.0	10905	371	224	300
LeViT-192	10.9	94.33	8837	605	224	300
EdgeViT-S	11.1	95.58	2274	1897	224	300
EfficientViT_M5	12.4	94.61	10621	522	224	300
MH-LViT_M3	12.5	95.78	10347	452	224	300
MH-LViT_M3*	12.5	**96.75**	10347	452	224	300
LeViT-256	18.9	94.36	6494	1059	224	300
LeViT-384	39.1	94.59	3883	2250	224	300

Feature fusion can effectively integrate the features extracted from different branches to avoid features in one branch are locked within this branch and get a chance to be processed through other branches, enabling the model to understand input images well which in turn improves the classification accuracy. This indicates that feature fusion is an important factor in improving the performance of MH-LViT.

3) Feature scale balancing strategy: In the proposed models we divide the input features by channels in a ratio of $3 : 2 : 1$ via the feature scale balancing strategy, to learn image representation in parallel through multiple paths. In this section, we vary this ratio to investigate its effects on model performance. As shown in Table V, we split the features with several typical ratios, where the baseline model EfficientViT_M1 and a double-paths scenario with the ratio of $1 : 1$ are also included. From the table one can observe that, compare to the baseline, the model utilizing double-paths with ratio $1 : 1$ improves the top-1 accuracy significantly, while the model using the ratio of $1 : 1 : 1$ is 0.4% higher than that of $1 : 1$, showing that multiple paths are helpful to increase model accuracy. We can also find that $2 : 2 : 1$ outperforms $1 : 1 : 1$ and among all proposed ratios, $3 : 2 : 1$ performs best which is utilized to develop the tiny models. This study suggests that a reasonable splitting ratio can facilitate feature extraction to improve model performance.

4) IntPNKD: IntPNKD is proposed to train the developed lightweight models to further improve their representation ability. In order to investigate its effect we run MH-LViT_M1* without using the IntPNKD. As shown in Table VI, the top-1 accuracy decreases by 2.58% when we give up IntPNKD, which demonstrates well its effectiveness. We also test the version with NKD by removing KD based on intermediate features prediction and its top-1 accuracy is 79.17%, which is 1.02% lower than the complete version, showing that the proposed IntPNKD is significantly helpful. By leveraging KD, the lightweight models can obtain an extra significant improvement without any model modification.

TABLE IV. Ablation Study of MH-LViT_M1 Model Design on cifar100

Model	ACC_Top1(%)
MH-LViT_M1	76.39
MH-LViT_M1 w/o convolution path	74.83 (↓1.56)
MH-LViT_M1 w/o self-attention path	71.96 (↓4.43)
MH-LViT_M1 w/o residual path	75.84 (↓0.55)
MH-LViT_M1 w/o Fusion Module	75.49 (↓0.9)

5) Mixup regularization: As shown in Table VI, we can observe a decrease of 1.14% in top-1 accuracy by removing the mixup regularization term L_{MR} from MH-LViT_M1*. This validates well the effectiveness of introducing the regularizer based on image mixing. As like IntPNKD, Mixup Regularization can also be done without any model modification and both of them are inference-cost free inference performance improvers.

6) Visualization: To demonstrate more intuitively the advantages of proposed model, we use thermal maps to visualize the attention maps sampled from dataset miniImageNet, as shown in Fig. 4. From the figure, we can observe significant gaps among different models. Compared with its competitors, our model MH-LViT_M3 pay more attention to the key areas that include discriminative features for image classification.

In summary, the ablation experimental results fully demonstrate the effectiveness of model design elements and model enhancement training strategies, namely, the multi-path design, the feature scale balancing strategy, the feature fusion, the IntPNKD and the mixup regularization. With these components, the family models of MH-LViT achieve excellent performance on benchmark datasets.

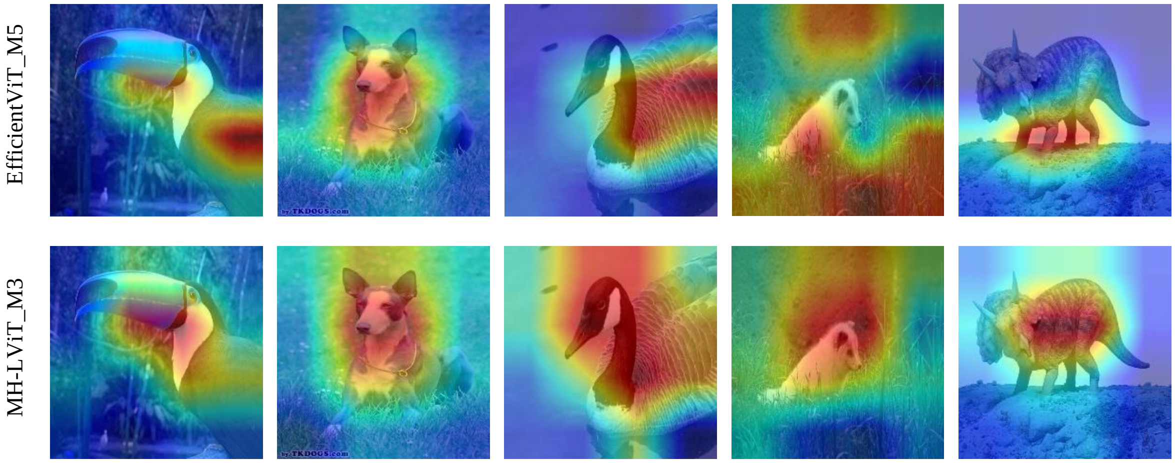

Fig. 4. Visualizations of attention maps on miniImageNet provided by MH-LViT_M3 and EfficientViT_M5.

TABLE V. Ablation Study of MH-LViT_M1 Feature Scale Balancing Strategy on Cifar100

Channel Split Ratio	ACC_Top1(%)
1 (EfficientViT_M1)	73.46
1:1	75.23 (↑1.77)
1:1:1	75.63 (↑2.17)
2:2:1	75.97 (↑2.51)
3:2:1	76.39 (↑2.93)

TABLE VI. Ablation Study of MH-LViT_M1 Model Enhancement on cifar100

Model	ACC_Top1(%)
MH_LViT_M1*	80.19
MH_LViT_M1* w/o IntPNKD	77.61 (↓2.58)
MH_LViT_M1* w/o IntPA	79.17 (↓1.02)
MH_LViT_M1* w/o L_{MR}	79.05 (↓1.14)

VI. Conclusion

In this paper, we propose a multi-path hybrid architecture for lightweight CV model to facilitate feature representation learning and develop a series of MH-LViT models. Within the multiple paths, the global features extraction ability is leveraged by ViT Branch and the local features extraction ability is enhanced by CNN branch. A residual connection branch is further introduce to complete the feature representation and a feature fusion module is utilized to shuffle extracted features and balance their chances to be processed in different branches. In order to exploit the representation potential of developed tiny models, we propose a novel knowledge distillation framework IntPNKD that introduces an extra intermediate layer prediction alignment in addition to the standard logit alignment. Finally, an mixup regularization term is utilized to further improve the generalization ability. Experimental results on benchmark datasets show that MH-LViT models balances well complexity and performance, providing an effective solution for visual tasks in resource-constrained applications. In the multi-path branches, we only utilize existing CNNs and ViTs and deliberately designed components will release fully the potential of proposed architecture. It is also interesting to evaluate MH-LViT models on more visual tasks.

Acknowledgment

This work is supported by the Natural Science Foundation of Fujian Province under Grant 2023J01348 and Grant 2020J01891, and partly by the National Natural Science Foundation of China under Grant 41971340.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *International Conference on Learning Representations*, 2020.

[2] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, “Scaling vision transformers,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 104–12 113.

[3] M. Ding, B. Xiao, N. Codella, P. Luo, J. Wang, and L. Yuan, “Davit: Dual attention vision transformers,” in *European conference on computer vision*. Springer, 2022, pp. 74–92.

[4] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[5] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, “Swin transformer v2: Scaling up capacity and resolution,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 009–12 019.

[6] Y. Li, H. Mao, R. Girshick, and K. He, “Exploring plain vision transformer backbones for object detection,” in *European Conference on Computer Vision*. Springer, 2022, pp. 280–296.

[7] J. N. Cheltha, C. Sharma, D. Prashar, A. A. Khan, and S. Kadry, “Enhanced human motion detection with hybrid rda-woa-based rnn and multiple hypothesis tracking for occlusion handling,” *Image and Vision Computing*, vol. 150, p. 105234, 2024.

[8] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, “Focal attention for long-range interactions in vision transformers,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 30 008–30 022, 2021.

[9] Z. Chen, Y. Duan, W. Wang, J. He, T. Lu, J. Dai, and Y. Qiao, “Vision transformer adapter for dense predictions,” in *The Eleventh International Conference on Learning Representations*, 2022.

[10] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao, “Eva: Exploring the limits of masked visual representation learning at scale,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 358–19 369.

[11] A. Alqarafi, A. A. Khan, R. K. Mahendran, M. Al-Sarem, and F. Albalwy, "Multi-scale gc-t2: Automated region of interest assisted skin cancer detection using multi-scale graph convolution and tri-movement based attention mechanism," *Biomedical Signal Processing and Control*, vol. 95, p. 106313, 2024.

[12] A. A. Khan, R. K. Mahendran, K. Perumal, and M. Faheem, "Dual-3dm 3-ad: mixed transformer based semantic segmentation and triplet pre-processing for early multi-class alzheimer's diagnosis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2024.

[13] A. Alhussen, M. A. Haq, A. A. Khan, R. K. Mahendran, and S. Kadry, "Xai-racapsnet: Relevance aware capsule network-based breast cancer detection using mammography images via explainability o-net roi segmentation," *Expert Systems with Applications*, p. 125461, 2024.

[14] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 116–131.

[15] X. Xu, S. Wang, Y. Chen, and J. Liu, "Plug n'play channel shuffle module for enhancing tiny vision transformers," in *2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2023, pp. 434–440.

[16] S. Mehta and M. Rastegari, "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," in *International Conference on Learning Representations*, 2021.

[17] B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jégou, and M. Douze, "Levit: a vision transformer in convnet's clothing for faster inference," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 259–12 269.

[18] S. d'Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, "Convit: Improving vision transformers with soft convolutional inductive biases," in *International conference on machine learning*. PMLR, 2021, pp. 2286–2296.

[19] J. Pan, A. Bulat, F. Tan, X. Zhu, L. Dudziak, H. Li, G. Tzimiropoulos, and B. Martinez, "Edgevits: Competing light-weight cnns on mobile devices with vision transformers," in *European Conference on Computer Vision*. Springer, 2022, pp. 294–311.

[20] M. Lou, H.-Y. Zhou, S. Yang, and Y. Yu, "Transxnet: Learning both global and local dynamics with a dual dynamic token mixer for visual recognition," *arXiv preprint arXiv:2310.19380*, 2023.

[21] Z. Yang, A. Zeng, Z. Li, T. Zhang, C. Yuan, and Y. Li, "From knowledge distillation to self-knowledge distillation: A unified approach with normalized loss and customized soft labels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 185–17 194.

[22] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[23] D. Bolya, C.-Y. Fu, X. Dai, P. Zhang, C. Feichtenhofer, and J. Hoffman, "Token merging: Your vit but faster," in *The Eleventh International Conference on Learning Representations*, 2022.

[24] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 22–31.

[25] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, and Y. Yuan, "Efficientvit: Memory efficient vision transformer with cascaded group attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 420–14 430.

[26] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[28] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.

[29] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *Computer Science*, vol. 14, no. 7, pp. 38–39, 2015.

[30] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*. PMLR, 2021, pp. 10 347–10 357.

[31] C. Yang, Z. An, H. Zhou, L. Cai, X. Zhi, J. Wu, Y. Xu, and Q. Zhang, "Mixskd: Self-knowledge distillation from mixup for image recognition," in *European Conference on Computer Vision*. Springer, 2022, pp. 534–551.

[32] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 11 953–11 962.

[33] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[34] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Handbook of Systemic Autoimmune Diseases*, vol. 1, no. 4, 2009.

[35] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, 2016.

[36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[37] R. Wightman *et al.*, "Pytorch image models," 2019.

[38] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[39] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 113–123.

[40] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 13 001–13 008.

[41] Y. Yu, S. Buchanan, D. Pai, T. Chu, Z. Wu, S. Tong, B. Haeffele, and Y. Ma, "White-box transformers via sparse rate reduction," *Advances in Neural Information Processing Systems*, vol. 36, 2024.