Transformers Meet Small Datasets

RAN SHAO1,2 AND XIAO-JUN BI3
1College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China
2College of Electronics and Information Engineering, Harbin Vocational & Technical College, Harbin 150001, China
3Department of Information Engineering, Minzu University of China, Beijing 100081, China
Corresponding author: Xiao-Jun Bi (bixiaojun@hrbeu.edu.cn)
This work was supported in part by the Major Projects of the National Social Science Foundation of China under Grant 20&ZD279.

ABSTRACT The research and application areas of transformers have been extensively enlarged due to the success of vision transformers (ViTs). However, due to the lack of local content acquisition capabilities, the pure transformer architectures cannot be trained directly on small datasets. In this work, we first propose a new hybrid model by combining the transformer and convolution neural network (CNN). The proposed model improves the classification ability on small datasets. This is accomplished by introducing more convolution operations in the transformer’s two core sections: 1) Instead of the original multi-head attention mechanism, we design a convolutional parameter sharing multi-head attention (CPSA) block that incorporates the convolutional parameter sharing projection in the attention mechanism; 2) the feed-forward network in each transformer encoder block is replaced with a local feed-forward network (LFFN) block that introduces a sandglass block with more depth-wise convolutions to provide more locality to the transformers. We achieve state-of-the-art results when training from scratch on 4 small datasets as compared with the transformers and CNNs without extensive computing resources and auxiliary training. The proposed strategy opens up new paths for the application of transformers on small datasets.

INDEX TERMS Convolutional neural networks, small datasets, transformer, vision transformer.

I. INTRODUCTION Transformers [1], [2] have become a de-facto standard in deep learning and have been widely adopted in various fields. These models have been widely adopted in modern deep learning, such as natural language processing (NLP) [3], [4], [5], computer vision (CV) [6], [7], [8], [9], and speech processing [10], [11], [12], due to their ability to model long-range dependencies.

The transformers were initially proposed as sequence-to-sequence models [13] for machine translation. More recently, the vision transformer (ViT) [14] is proposed, which is the first pure transformer architecture applied in computer vision. It obtains competitive results as compared to many state-of-the-art convolution neural networks (CNNs) [15], [16], [17], [18]. However, despite the success of ViT on large datasets, its performance still falls behind as compared to the similar-sized CNNs [19] (e.g., ResNet [15]), when trained on small datasets. This may be because the ViT lacks the locality incorporated in the design of CNNs, thus making them very suitable for solving the vision problems [14]. Please note that the images have a strong 2D local structure with spatially neighboring pixels generally being highly correlated. Therefore, the CNNs that employ more locality are able to capture this local information. However, the CNN models are not good at capturing long-distance dependency information in image data. Consequently, the long-distance dependency features are lost. Additionally, due to the over-flexibility of ViT in terms of extraction features, it is prone to overfitting, when trained using small datasets, consequently leading to a poor predictive performance on small datasets [20], e.g., image classification on CIFAR.

However, for some specialized fields, obtaining large datasets is a challenging task. Please note that due to the small sample size and poor diversity, a dataset of suitable size cannot be obtained even by using generative adversarial networks. For example, in the medical field, most positive samples of diseases are small [21], and we cannot obtain the datasets as big as JFT-300 [22]. Additionally, for a novel or
rare disease, it is even more difficult to obtain a valid positive sample.

In order to work effectively with small datasets, most transformers models employ a pre-training approach [1], [19]. However, the pre-training of transformers on large datasets followed by a downstream finetuning on small datasets might become onerous or ineffective at times. For instance, we may need to train ten separate models for the same task and deploy them on various hardware platforms, but pre-training ten models using a large dataset is problematic. Furthermore, the problems in some domains have nothing in common with the domains of pre-trained models. As a result, the pre-training has a little or no influence on the performance in a new domain, especially when the domains are sufficiently distinct [23], [24].

In order to overcome these limitations of pre-training, several hybrid architectures have been proposed recently [25], [26], [27], [28]. The idea behind these works is to combine the convolutional layers with the attention layers to induce a local inductive bias in ViT. The advantages of both paradigms are combined in these architectures, i.e., attention layers model long-range dependencies, while the convolutional layers capture the local contents of an image. The empirical results in most of these works suggest that the hybrid architectures can be trained on middle-size datasets, such as ImageNet, and outperform various large CNN models, such as ResNet. However, these methods still fall behind the similar-sized CNNs on small datasets like CIFAR-10, when trained from scratch. However, most of the vision applications do not have the liberty of datasets having a scale comparable to ImageNet from an application standpoint.

In this work, we hypothesize that the transformers can be trained on small datasets from scratch by reasonably introducing more locality. In order to verify the proposed hypothesis, we propose transformers meet small datasets (TSD). The TSD is a new hybrid architecture that introduces more convolution operations in the transformer’s two core sections and is intrinsically efficient, both in terms of floating-point operations (FLOPs) and number of parameters.

As shown in Fig. 1, we divide the transformer into multiple stages for designing a novel transformer framework. The input images first go through the convolution stem for feature extraction as presented in [10] and [27]. By using the convolutional stem instead of the original patch embedding block in the ViT model, we can obtain the low-dimensional features from the images more flexibly and efficiently. This enables us to avoid the problem of patch edge information loss in the ViT model. The resulting data without position embedding is then fed in the proposed convolutional parameter sharing multi-head attention (CPSA) block for attention representation learning. In the CPSA block, inspired by [29] and [30], the linear projection before each self-attention block in the transformer model is replaced by the convolutional parameter sharing projection. This replacement enables the utilization of $3 \times 3$ depth-wise separable convolution operations [31] on a 2D-reshaped token map and shares the parameters of key matrices and value matrices of the original transformer model. As a result, the block captures more local spatial dependencies and removes the local content ambiguity from the attention process. At the same time, it also reduces the computational complexity and over-flexibility of the transformer model by subsampling and sharing the parameters of matrices, resulting in an increase in the efficiency with minimal performance degradation. Furthermore, inspired by [16], [25], and [27], we replace the feed-forward network (FFN) in the transformer model with the proposed local feed-forward network (LFFN) block, which introduces a sandglass block with more depth-wise convolutions in the feed-forward network for providing more locality to the transformers. Due to a well-designed structure, such depth-wise convolutions precisely provide the mechanism for local content aggregation that is not available in the feed-forward network of ViT. Please note that both in terms of parameters and computing complexity, the depth-wise convolution is efficient. Finally, the global average pooling is used to replace the class token in the transformer model for better classification results.

The main contributions of this work are as follows.

1) We proposed a novel visual transformer architecture namely TSD. It obtains all the advantages of both CNNs and transformers by reasonably introducing two novel blocks, i.e., CPSA and LFFN, and designing the network structure, thus enabling the transformer to capture both the local and long-range information from images.

2) The experimental results show that when the proposed model TSD is trained using small datasets from scratch, it achieves state-of-the-art performance while remaining lightweight and efficient. Furthermore, the proposed TSD outperforms CNN-based models and previous transformer-based models, achieving a top-1 accuracy of 96.25% on CIFAR-10, while having fewer FLOPs and number of parameters.

3) Based on the proposed approach, we can eliminate the token positional embedding without sacrificing the model performance. This streamlines the architecture design while allowing us to change the input image resolutions, which is crucial for many vision-based tasks.

The rest of this paper is organized as follows.

We conclude a brief overview of vision transformers, depth-wise convolution, and inverted residuals in Section II. Section III presents the details of the proposed TSD. Section IV describes the training details and comprehensive experimental comparisons. Finally, Section V concludes this work.

II. RELATED WORKS

A. VISION TRANSFORMERS

The vision transformer (ViT) [14] is the first network to incorporate the transformer block for computer vision problems. This model achieves the SOTA results when the dataset
is large enough, e.g., on ImageNet-22k and JFT-300M, etc. Subsequently, various parallel models have been proposed to effectively fit the field of computer vision. For instance, transformer-in-transformer (TNT) [32] utilizes the external embedded transform block for patch embedding and the internal transform block for pixel embedding to achieve patch-level and pixel-level synchronization. Tokens-to-token (T2T) [33] uses a sliding window for converting multiple tokens originally located in the ViT into a single token. This operation improves the performance of the model; however, multiple tokens greatly increase the computational complexity. Recently, several data efficient ViTs have been proposed to reduce their reliance on large datasets. For instance, ConViT [34] uses gated positional self-attention (GPSA) to simulate the properties of a convolution layer, so that the network has convolution-like local feature extraction capability. The convolution enhanced image transformer (CeiT) [27] uses convolution to first extract the features from an image, and then passes these features into the transformer backbone as tokens. It proposes an enhanced feed-forward network that further combines the convolution and transformer for improving the performance of the network. CvT [29] modified both the linear projection and multi-layer perceptron with convolution operation. Swin transformer [35] is a hierarchical transformer that utilizes the attention computed in non-overlapping local windows to gradually reduce the number of tokens through patch merging. These improvements have been made to eliminate the need of pre-training the transformers on large datasets, such as JFT-300M, and achieve high accuracy by training the models from scratch by using the middle-sized datasets, like ImageNet. In this work, we aim to accomplish efficient training for the transformer model from scratch by using small datasets, like CIFAR-10.

However, it is noteworthy that very few attempts have been made on small datasets to the best of our knowledge. The work presented in [36] is the only work that considers the training of transformer model by using small dataset. The authors in this work propose an efficient training of ViT on small datasets by introducing a self-supervised task that extracts additional information from the images and encourages the ViT to learn more local contents. However, this self-supervised task results in a more complicated design, and the performance still falls behind the similar sized CNN counterparts, when trained on small datasets like CIFAR-10. The key contrast between the proposed work and previous research is that we redesign a streamlined transformer network by reasonably incorporating the convolutions in its core sections, so that it can be trained effectively and efficiently on small datasets to outperform the state-of-the-art transformers and CNNs.

B. DEPTH-WISE CONVOLUTION AND INVERTED RESIDUALS

As compared with normal convolution, the computations in depth-wise convolution are only handled channel-wise. In other words, only one input feature map is used for convolution to obtain a channel of the output feature map. As a result, the depth-wise convolution is effective in terms of computation and number of parameters. The MobileNet based on depth-wise separable convolutions is initially presented by Howard et al. [37]. This network is very mobile device friendly and computationally efficient. Since then, various other works have employed depth-wise convolutions for
creating efficient networks. This encourages us to consider the introduction of more depth-wise convolutions in the proposed model for improving the locality of the network and ensure its effectiveness.

MobileNetV2 [16] introduced the first inverted residual blocks, which are based on depth-wise convolution. A series of \(1 \times 1 - \text{depth-wise} - 1 \times 1\) convolutions make up the inverted residual blocks. Between the two \(1 \times 1\) convolutions, a hidden dimension is expanded. Subsequently, in order to eliminate the risk of lower dimensions information loss in the inverted residual blocks, a new inverted residual block called sandglass block is proposed in MobileNeXt [38]. We observe that the feed-forward network of vision transformers and the extension of the hidden dimension in the inverted residuals are fairly similar. This motivated us to come up with a new design for the novel block instead of FFN, so that the transformer is able to capture more locality.

III. THE PROPOSED METHOD
A. OVERALL ARCHITECTURE
In this work, we aim to design a new hybrid transformer architecture that introduces more convolution operations in the transformer’s core sections so that it is able to capture more locality and improves the accuracy and efficiency of classification on small datasets. An overview of the proposed TSD is depicted in Fig. 1. A multi-stage architecture is adopted, where all the stages share a similar architecture in this work, which consists of a CPSA block and LFFN block.

First, instead of employing a poor tokenization like ViT that divides a given image into non-overlapping patches of equal size, and linearly projects each patch on the visual tokens. Inspired by the works presented in [10], [27], and [29], we utilize a convolution token embedding (CTE) block consisting of \(3 \times 3\) convolutions, followed by ReLU activation and a max-pooling layer for effectively extracting the local information. The block increases the ability of the model to capture both the low-dimensional local information and avoid the loss of patch edge information. Afterwards, the tokens without the position embedding are passed through a new transformer encoder comprising CPSA and LFFN blocks. The CPSA and LFFN blocks not only reduce the computational complexity and the over-flexibility of the transformer model, but also capture both the local contents and long-range dependencies. A global average pooling layer, a fully connected layer, and a classification layer with SoftMax make up the model’s final layers.

Due to a strong ability of the proposed model to capture the local information and fewer parameters, it is more suitable for solving image classification tasks using small datasets. In addition, this model does not require positional embedding to aid the training process.

Now, we first show the implementation of novel projection method using the CPSA block. Then, we elaborate the LFFN block and its efficient design for improving the performance of the network.

B. CONVOLUTIONAL PARAMETERS SHARING MULTI-HEAD ATTENTION (CPSA)
Since the proposed TSD needs to achieve more locality and be efficient as well, we propose a CPSA block to replace the traditional multi-head attention layer (MHA). Fig. 2(a) presents the original MHA that employs the linear projection and Fig. 2(b) presents the proposed CPSA block that employs the convolutional parameters sharing projection.

Unlike MHA, in order to reduce the model complexity and achieve efficiency, we enable sharing between the key and value matrices \(S\). In other words, the proposed CPSA takes a query \(Q\), sharing parameters matrices \(S\) instead of a query \(Q\), a key \(K\) and a value \(V\) to yield a refined token feature. First, the proposed CPSA reduces the spatial scales of \(S\) based on depth-wise convolutions before performing the attention operation, as presented in Fig. 2. We reshape the tokens from 1D to 2D feature maps. Then, a convolutional projection is implemented by using a depth-wise separable convolution operation with kernel size \(3\) and a step size of \(2\). This operation increases both the local information acquisition ability and model efficiency simultaneously. Afterwards, the tokens are compressed into 1D sequence features of subsequent processing. This is mathematically formulated as follows:

\[
S = \text{Flatten}(\text{DWConv2d}((\text{Reshape2D}(x_i, s, t)))) \tag{1}
\]

where, \(x_i \in \mathbb{R}^{N \times C}\) denotes the input tokens before convolution projection. \(N\) represents the number of token features and \(C\) represents the feature dimension of the token. \(S \in \mathbb{R}^{N/2 \times C}\) denotes the parameters shared between matrices used for calculating the subsequent attention factor, \(\text{DWConv2d}\) denotes depth-wise separable convolution operation, \(\text{Reshape2D}\) denotes the transformation of 1D features into 2D features, \(s\) represents the size of the convolution kernel, and \(t\) represents the size of convolution stride.

Similar to the work presented in [30] for guaranteeing the consistent dimensionality of the output tokens, \(Q \in \mathbb{R}^{N \times C}\) always maintains its original dimension and does not participate in the convolution operation. Now, the proposed attention operation is mathematically expressed as follows:

\[
\text{Attention}(Q, S) = \text{Soft max}\left(\frac{QS^T}{\sqrt{d_{\text{head}}}}\right)S \tag{2}
\]

where, \(\text{attention()}\) denotes the output token features after attention calculation and \(d_{\text{head}}\) denotes the dimension of each attention head.

Three important factors in the CPSA block guarantee the efficiency in the proposed method. First, the use of depthwise convolution. We only introduce additional \(s^2C\) parameters and \(O(s^2CT)\) FLOPs as compared to the linear projection, which is negligible as compared to the total number of parameters and FLOPs in the models. Second, the process of matrix sharing \(S\). With this improvement, the number of parameters of key and value are reduced by half. In this way, the learning ability of the network is reduced. However, in case of small datasets, this helps to prevent overfitting. Third, the
convolution projection which enables us to utilize a convolution kernel with a stride of 2. Each convolution operation is comparable to a down sampling operation of matrices that halves the data. Consequently, the number of S parameters decreases by a factor of 2, and the computational cost of the later corresponding multiheaded attention mechanism is lowered by a factor of 4.

C. LOCAL FEED-FORWARD NETWORK (LFFN)

In order to further improve the ability of transformers to obtain locality in higher and lower dimensions at the same time, we propose LFFN block which introduces a sandglass block with more depth-wise convolutions in the original FFN of transformers. The structure of the LFFN block is presented in Fig. 3.

The LFFN block performs the following procedures. First, given tokens $x_i^h \in \mathbb{R}^{N \times C}$ generated from the CPSA block, where $N$ represents the number of token features and $C$ represents the feature dimension of the token, the embedding tokens are converted from 1D to 2D features, i.e., $x_i^j \in \mathbb{R}^{\sqrt{N} \times \sqrt{N} \times C}$. Second, in order to capture the local information in lower dimensions, we process the 2D tokens by using a depth-wise convolution with a kernel size of 3, generating $x_i^{d(1)} \in \mathbb{R}^{\sqrt{N} \times \sqrt{N} \times (e \times C)}$. The feature dimension is then increased by performing convolution with a kernel size of 1 to a higher dimension of $x_i^{l(1)} \in \mathbb{R}^{\sqrt{N} \times \sqrt{N} \times (e \times C)}$. After expanding the dimensionality of the tokens, we further improve the local dependency in a higher dimension with the neighboring tokens by performing a depth-wise convolution with a kernel size of 3 on these tokens, thus obtaining $x_i^{d(2)} \in \mathbb{R}^{\sqrt{N} \times \sqrt{N} \times (e \times C)}$. Third, $x_i^{l(2)} \in \mathbb{R}^{\sqrt{N} \times \sqrt{N} \times C}$ is obtained by performing a convolution with a kernel size of 1 for dimensionality reduction. We finally utilize a depth-wise convolution with a kernel of size 3 for obtaining much more locality in a lower dimension, yielding $x_i^{d(3)} \in \mathbb{R}^{\sqrt{N} \times \sqrt{N} \times C}$, and then convert the tokens from 2D to 1D features, i.e., $x_i^{h+1} \in \mathbb{R}^{N \times C}$. Additionally, following each depth-wise convolution, h-swish and squeeze-and-excitation (SE) modules are added. These processes are mathematically described as follows:

\[
\begin{align*}
x_i^S &= \text{Reshape2D}(x_i^h) \quad (3) \\
x_i^{d(1)} &= \text{DConv2d}(x_i^S) \quad (4) \\
x_i^{l(1)} &= \text{Conv}(x_i^{d(1)}) \quad (5) \\
x_i^{d(2)} &= \text{DConv2d}(x_i^{l(1)}) \quad (6) \\
x_i^{l(2)} &= \text{Conv}(x_i^{d(2)}) \quad (7) \\
x_i^{d(3)} &= \text{DConv2d}(x_i^{l(2)}) \quad (8) \\
x_i^{h+1} &= \text{Flatten}(x_i^{d(3)}) \quad (9)
\end{align*}
\]

Moreover, in the LFFN block, the major extra computation cost is introduced by the depth-wise convolutions, which is negligible as compared to the total number of parameters and FLOPs in the whole model.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we evaluate the proposed TSD model by using small image classification datasets. Moreover, we also conduct the ablation investigations to confirm the design of the proposed architecture.
TABLE 1. Variants of our TSD architecture. And $k \times s \times 1$ means a convolution/pooling with kernel size of $k$ and stride of $s$.

| Model | CTE | encoder | embedding | CPSA | LFFN | Params | FLOPs |
|-------|-----|---------|-----------|------|------|--------|-------|
|       | conv | maxpool | channels  | conv | heads| e      | k     | (M)   | (G)   |
| TSD-T | $k_3$ | $k_3$ | 64        | 2    | 128  | 2      | 3    | 1.2   | 0.11  |
| TSD-S | $k_3$ | $k_3$ | 64        | 4    | 128  | 2      | 3    | 2.5   | 0.26  |
| TSD-B | $k_3$ | $k_3$ | 64        | 6    | 128  | 2      | 3    | 3.8   | 0.39  |

TABLE 2. The size of the datasets used in our empirical analysis.

| Datasets        | Train size | Test size | Classes |
|-----------------|------------|-----------|---------|
| CIFAR-10        | 50,000     | 10,000    | 10      |
| CIFAR-100       | 50,000     | 10,000    | 100     |
| MNIST           | 55,000     | 10,000    | 10      |
| Fashion-MNIST   | 60,000     | 10,000    | 10      |

A. EXPERIMENTAL SETTINGS

1) NETWORK ARCHITECTURES

We design the proposed TSD architectures by following the basic settings of ViTs. The corresponding details are presented in Table 1. The CTE block consists of a convolutional layer with a kernel size of 3 and a stride of 1. It generates 64 enriched channels. For stable training, a BatchNorm layer is also introduced. Then, a max-pooling layer is used with a kernel size of 3 and a stride of 2. The depth of the encoder blocks is set to 2, 4, and 6 separately. The size of the convolution kernel in the CPSA block, which is used for convolution projection, is set to 3, and the number of heads is set to 2. For the LFFN block, we set the expand ratio $e$ to be 2. The kernel size for performing depth-wise convolutions is $3 \times 3$.

2) DATASETS

Instead of using a large training dataset, such as JFT-300 or ImageNet, we adopt 4 well-known and commonly used small datasets, including CIFAR-10 [39], CIFAR-100 [39], MNIST [40], and Fashion-MNIST [41]. Table 2 presents the number of samples for each of these 4 datasets. In order to further validate the classification performance of the proposed model presented in Table 1, we perform an extended experiment by using the EMDS-6 [42]. EMDS-6 is an environmental microorganism dataset containing 840 distinct environmental microorganism images of 21 categories. As the EMDS-6 is a very small dataset, 37.5% of the dataset is selected as the training set, 25% as the validation set, and 37.5% as the test set as discussed in [43]. For performing data augmentation, three different GAN enhancements presented in [44] are adopted to enhance the EMDS-6 dataset, namely EF-DCGAN, EF-WGAN, and EF-WGAN-GP. After data augmentation, the size of the dataset increases 6 times than that of the original dataset.

3) IMPLEMENTATION DETAILS

We perform experiments on NVIDIA GTX1080Ti GPU and adopt the same training strategy used in CvT [29]. The initial learning rate is set to 0.02. The weight decay factor in Adam optimizer is set to 0.05. The label smoothing with a probability of 0.1 is used. The batch size is set to 64 by default. All the models are warmed up for 10 epochs. Unless mentioned otherwise, we train the model for 300 epochs. Lastly, we adopt the same data augmentation and regularization method as CvT. In the extended experiment, the learning rate is set to 0.002 and the batch size is set to 32. The model is trained for 100 epochs. For the proposed TSD model, we use the images and the corresponding labels as the input and output for performing end-to-end training. The cross-entropy loss ensures that the network training proceeds properly.

B. COMPARISON TO STATE OF THE ART

We report the results on small datasets, such as CIFAR-10, CIFAR-100 etc. in Table 3. For comparison, we select the state-of-the-art classification models, including transformer-based models (ViT [14], Swin [35], T2T [33], CvT [29], and CvT+$L_{	ext{decode}}$ [36]) and representative CNN based models (ResNet18 [15], ResNet50 [15], MobileNetV2 [16], ResNet164-v2 [17], and ResNet1001-v2 [17]) to evaluate the effectiveness of the proposed TSD models.

1) COMPARISON WITH THE TRANSFORMER-BASED MODELS

We first compare the proposed TSD model with the transformer-based models using all the datasets presented in Table 3. The TSD-T achieves an accuracy of 93.27%, which is higher than ViT-Base, Swin-T, and T2T-ViT-14 by 15.79%, 13.7%, and 7.15%, respectively on CIFAR-10. Similarly, it achieves 76.02%, 96.56%, and 99.75% accuracies and outperforms the aforementioned transformer-based models on CIFAR-100, Fashion-MNIST, and MNIST, respectively. This is a surprising result since the size of TSD-T is one-seventeenth the size of the smallest model, i.e., T2T-ViT-14. However, this result is a consequence of the improvements in the training strategy and the modifications made in the architecture of the model.

In order to further demonstrate the improvements achieved by the proposed structure, we compare the performances of TSD-T with CvT-13. TSD-T models follow the same training strategy as presented in Section IV.A. The TSD-T outperforms CvT-13 by a margin of 1.25% on CIFAR-10 and 0.7% on CIFAR-100. In addition, it outperforms CvT-13 on the remaining three datasets as well. The number of parameters of TSD-T is one-sixteenth of CvT-13, and the number of
TABLE 3. Top-1 validation accuracy trained from scratch on different datasets (300 epochs).

| Model                  | CIFAR-10 | CIFAR-100 | Fashion-MNIST | MNIST    | #Params | FLOPs |
|------------------------|----------|-----------|---------------|----------|---------|-------|
| Convolutional Networks |          |           |               |          |         |       |
| ResNet18               | 90.56%   | 65.86%    | 94.20%        | 99.20%   | 11.7M   | 1.8G  |
| ResNet50               | 91.02%   | 64.52%    | 93.67%        | 99.21%   | 25.6M   | 4.1G  |
| MobileNetV2/1.4        | 92.81%   | 67.57%    | 94.57%        | 99.41%   | 6.9M    | 1.2G  |
| ResNet164-v2[56]       | 94.54%   | 75.67%    | -             | -        | 1.7M    | 0.26G |
| ResNet1001-v2[56]      | 95.38%   | -         | -             | -        | 10.2M   | 1.55G |
| Vision Transformers    |          |           |               |          |         |       |
| ViT-Base               | 77.48%   | 47.65%    | 84.65%        | 83.88%   | 86M     | 18.7G |
| Swin-T                 | 79.57%   | 63.88%    | 89.00%        | 90.11%   | 29M     | 4.5G  |
| T2T-ViT-14             | 86.12%   | 70.04%    | 92.52%        | 98.41%   | 21.5M   | 5.2G  |
| CvT-13                 | 92.02%   | 75.32%    | 94.65%        | 99.02%   | 20M     | 4.5G  |
| CvT-13+Ldrloc          | 94.30%   | 77.21%    | 95.26%        | 99.32%   | 20M     | 4.5G  |
| Ours                   |          |           |               |          |         |       |
| TSD-T                  | 93.27%   | 76.02%    | 95.12%        | 99.02%   | 1.2M    | 0.11G |
| TSD-S                  | 95.80%   | 77.91%    | 96.20%        | 99.61%   | 2.5M    | 0.26G |
| TSD-B                  | 96.25%   | 78.84%    | 96.56%        | 99.75%   | 3.8M    | 0.39G |

FLOPs is one-fortieth of CvT-13. We also compare TSD with CvT-13+Ldrloc that is specifically designed for small datasets. Please note that the TSD-S with much fewer FLOPs and parameters achieves higher accuracy as compared to CvT-13+Ldrloc on all small datasets presented in Table 3. In order to further validate the scalability of the proposed models, we evaluate the performance of TSD-B by training it from scratch based on small datasets. To the best of our knowledge, TSD-B is the best state-of-the-art model. It achieves the highest accuracy, i.e., 96.25% on CIFAR-10 and 78.84% on CIFAR-100. The experimental results show that the transformer can be trained from scratch with high accuracy based on small datasets by incorporating more locality. This is consistent with the hypothesis considered in this work.

2) COMPARISON WITH THE CNN-BASED MODELS

We compare the proposed TSD model with CNN-based models based on the datasets presented in Table 3. The TSD-T not only achieves a top-1 accuracy of 0.46% higher as compared to the MobileNetV2/1.4 on CIFAR-10, but also surpasses the MobileNetV2/1.4 on three other small datasets as well. However, the TSD-T consists of one-tenth FLOPs and one-sixth parameters as compared to MobileNetV2/1.4. For a TSD-S of similar size as ResNet164-v2, its performance is 95.80% on CIFAR-10, which is +1.26% higher as compared to ResNet164-v2 (94.54%), which is designed especially for CIFAR. This model also outperforms larger CNN models, such as ResNet1001-v2, on all the datasets presented in Table 3, while having fewer parameters and FLOPs. When the TSD-B model is used, the classification accuracy is better as compared to all the CNN-based models. This shows that the local information acquisition capability of the model has been successfully enhanced and surpasses the state-of-the-art CNN-based models.

C. EXTENDED EXPERIMENT

1) CLASSIFICATION PERFORMANCE ON EMDS-6

To further validate the classification performance of the proposed model based on a small dataset, we applied the proposed TSD to a small microbial dataset, i.e., EMDS-6. As shown in Table 4, we compared the proposed TSD-B model with other models by using the test set of EMDS-6. As compared to the CNN-based models, TSD-B achieves an accuracy of 39.57%, which is higher than MobileNetV2/1.4, ResNet18, and ResNet50 by 4.28%, 5.24%, and 5.24%, respectively. As compared to the transformer-based models, the proposed model achieves higher accuracy as compared to T2T-ViT-14 and ViT-Base, i.e., 4.29% and 6.83% higher, respectively. The classification accuracy of TSD-B is slightly lower than that of Xception. However, please note that the size of TSD-B is one-sixth the size of the Xception model. In addition, in terms of inference time, the model proposed in this work outperforms all other models and consumes 3.70 seconds only. The experimental results further illustrate the rationality of the model architecture presented in this paper.

2) CLASSIFICATION PERFORMANCE ON EMDS-6 AFTER DATA AUGMENTATION WITH GANS

In order to further verify the performance of the proposed model in this work, we first utilize three EF-GAN models [44] for data enhancement of the EMDS-6 dataset. The corresponding enhancement configuration is presented in section IV. As shown in Table 5, we find that the performance of all models improves after data augmentation. The classification performance of the TSD-B model significantly outperforms the T2T-ViT-14 model under three different data augmentation cases. After data enhancement using the EF-WGAN-GP model, the highest classification accuracy
TABLE 4. Comparison of classification results of different models on the EMDS-6. Time(s) denotes inference time.

| Model          | EMDS-6  | #Params | Time(s) |
|----------------|---------|---------|---------|
| ViT-Base       | 31.74%  | 86M     | 3.77    |
| ResNet50       | 33.33%  | 25.6M   | 6.23    |
| ResNet18       | 33.33%  | 11.7M   | 4.92    |
| T2T-ViT-14     | 34.28%  | 21.5M   | 4.44    |
| MobileNetV2/1.4| 34.29%  | 6.9M    | 5.13    |
| Xception       | 40.32%  | 23M     | 5.63    |
| TSD-B          | 38.37%  | 3.8M    | 3.70    |

TABLE 5. Comparison of classification results of different models on the EMDS-6 after data augmentation.

| Model          | EF-DCGAN | EF-WGAN     | EF-WGAN-GP |
|----------------|----------|-------------|-------------|
| T2T-ViT-14     | 36.28%   | 37.13%      | 39.53%      |
| Xception       | 44.32%   | 45.22%      | 45.71%      |
| TSD-B          | 43.74%   | 45.02%      | 46.37%      |

TABLE 6. Ablations on CPSA and LFFN. We use CPSA and LFFN in all our models.

| Model          | None | CPSA | LFFN | CIFAR-10 |
|----------------|------|------|------|----------|
| TSD-T          | ✓    | ✓    | ✓    | 85.84%   |
| TSD-S          | ✓    | ✓    | ✓    | 90.35%   |
| TSD-B          | ✓    | ✓    | ✓    | 93.27%   |

TABLE 7. Ablations on position embedding.

| Model          | PE | CIFAR-10 | CIFAR-100 |
|----------------|----|----------|-----------|
| Transformer-based model |    |          |           |
| T2T-ViT-14     | ✓  | 86.12%   | 70.04%    |
| Our models     |    |          |           |
| TSD-T          | ✓  | 93.16%   | 75.59%    |
| TSD-S          | ✓  | 95.41%   | 77.02%    |
| TSD-B          | ✓  | 95.80%   | 77.91%    |

of 46.37% is obtained by outperforming the best Xception model. This further validates the feature extraction capability of the model proposed in this work.

D. ABBLATION STUDY

We conduct different ablation experiments to verify the effectiveness of the proposed blocks. First, we observe how the presence or absence of the CPSA and LFFN blocks affect the TSD. Second, we show that the position embeddings can be removed from the model by introducing the proposed CPSA and LFFN. Third, we investigate the effect of the convolutional parameter sharing (CPS) projection of the CPSA block on the performance of the model. Lastly, we study the effectiveness of different types of LFFN blocks.

1) EFFECTIVENESS OF CPSA AND LFFN

The proposed TSD model differs from the transformer in a number of ways. The most notable difference is that the proposed model incorporates CPSA and LFFN in place of the original MHA and FFN available in the transformer models, respectively. Now, we study the influence of CPSA and LFFN on the performance of the proposed method by mutating every block towards a transformer block, while keeping the other parameters unchanged. We obtain the classification accuracy on CIFAR-10 dataset. The model is trained for 300 epochs. The effects of each modification on the proposed TSD models are presented in Table 6. We observe that the CPSA sub-block is the most significant feature and having an LFFN also makes the model more effective as compared to the original transformer containing FFN. These findings show that the addition of CPSA and LFFN in the transformer assists in increasing the classification precision on small datasets.

2) REMOVAL OF POSITION EMBEDDING

We investigate whether the position embedding is required in the proposed TSD given that the convolutions have been incorporated in the model. The results in Table 7 reveal that the performance of the proposed model is not negatively impacted by removing the position embedding. As a result, we remove the position embeddings from the TSD by default. Comparatively, since the T2T-ViT-14 does not describe more spatial relations within the images, removing the position embedding of T2T-ViT-14 results in performance loss of 3.47% on the CIFAR-10 dataset and a loss of 4.07% on the CIFAR-100 dataset. This demonstrates the potency of the proposed models. The position embedding is frequently implemented by using the learnable vectors of fixed length, thus restricting the ability of the trained model to adapt the input of variable length. However, a variety of vision applications use images with different resolutions.

The CPVT [45] attempts to represent the position information on-the-fly by replacing the explicit position embedding of ViT with a conditional position encoding module. The positional embedding can be entirely removed by using TSD, thus creating a possibility of simplifying the adaptation process for new vision tasks without necessitating a redesign of the embedding.
3) EFFECTIVENESS OF CPS PROJECTION
We study how the proposed CPS projection block affects the performance of the proposed TSD model by mutating a CPS projection in the CPSA block towards a linear projection in the MHA block. The corresponding results are presented in Table 8. We observe that replacing the linear projection with the convolutional projection improves the top-1 accuracy on CIFAR-10. When we employ the CPS projection, we obtain the highest accuracy on CIFAR-10. Additionally, as more stages employ the design, the performance keeps improving. Therefore, the proposed hypothesis is proved, i.e., employing the CPS projection effectively improves the model performance on small datasets.

4) DIFFERENT TYPES OF LFFN BLOCK
In the LFFN block, the size of the kernel determines the size of the captured local information. Therefore, we test the kernel sizes of $1 \times 1$, $3 \times 3$, and $5 \times 5$ as presented in Table 9. The $1 \times 1$ type shows poor performance as compared to the baselines that do not employ depth-wise convolution as an extension layer. This demonstrates that increasing the number of layers in the transformer will not necessarily result in improvement. When the kernel size is increased, the block captures more local content. We find both types of $3 \times 3$ and $5 \times 5$ gains. However, we select the kernel size of $3 \times 3$ as default based on the trade-off between the number of parameters and accuracy of the model.

5) EFFECTIVENESS OF DEPTH-WISE CONVOLUTION IN LFFN BLOCK
In the LFFN block, we use depth-wise separable convolution to extract the specific elements efficiently. Therefore, in this section, we investigate the effect of the depth-wise separable convolution by replacing it with standard $3 \times 3$ convolution as presented in Table 10. We find that using standard convolution in the LFFN module not only results in low classification accuracy but also the number of parameters and FLOPs considerably exceeds that of depth-wise separable convolution. This is due to the strong feature capture capability of transformer-based models, using a deep-wise separable convolution.
convolution instead of standard convolution can better prevent model overfitting and improve the generalization ability of the model, thus performing better in classification tasks on small datasets. Therefore, we choose the depth-wise separable convolution as the convolution layer of LFFN.

### E. QUALITATIVE RESULT

In order to further evaluate the property of the proposed TSD, we apply Grad-CAM [46] on the output in the last stage to qualitatively evaluate the TSD. The attention scores on the CIFAR-10 dataset are presented in Fig. 4. It is evident that the shape of the object in the background can be captured more accurately by using the proposed model. In particular, the proposed TSD accurately covers the targets in the images and pays less attention to the background as compared to the baseline ViT and CvT13+Lblocl. As a result, we conclude that the proposed approach motivates ViT to perform better on small datasets by effectively introducing more locality.

### V. CONCLUSION

In this work, we first present a hybrid architecture comprising CNN and transformer model, which is suitable for image classification tasks based on small datasets. The extensive experiments have shown that the introduction of CPSA and LFFN blocks enables the proposed network architecture to achieve the state-of-art performance, while maintaining computational efficiency. Furthermore, due to an increased ability of the network to capture more local contents, the proposed model no longer requires a position embedding. By introducing more locality, we provide a new perspective for applying transforms on small datasets.

### REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 5998–6008.

[2] R. Castro, I. Pineda, W. Lim, and M. E. Morocho-Cayamcela, “Deep learning approaches based on transformer architectures for image captioning tasks,” IEEE Access, vol. 10, pp. 33679–33694, 2022, doi: 10.1109/ACCESS.2022.3162685.

[3] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A lite BERT for self-supervised learning of language representations,” 2019, arXiv:1909.11942.

[4] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, “XLNet: Generalized autoregressive pretraining for language understanding,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 5753–5763.

[5] D. W. Otter, J. R. Medina, and J. K. Kalita, “A survey of the usages of deep learning for natural language processing,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 2, pp. 604–624, Feb. 2021, doi: 10.1109/TNNLS.2020.2979670.

[6] D. Karimi, H. Dou, and A. Gholipour, “Medical image segmentation using transformer networks,” IEEE Access, vol. 10, pp. 29322–29332, 2022, doi: 10.1109/ACCESS.2022.3156894.

[7] J. Duan, Y. He, B. Du, R. M. R. Ghandour, W. Wu, and H. Zhang, “Intelligent localization of transformer internal degradations combining deep convolutional neural networks and image segmentation,” IEEE Access, vol. 7, pp. 62705–62720, 2019, doi: 10.1109/ACCESS.2019.2916461.

[8] C. Yan, G. Shi, and Z. Wu, “SMIR: A transformer-based model for MRI super-resolution reconstruction,” in Proc. IEEE Int. Conf. Med. Imag. Phys. Eng. (ICMIPE), Nov. 2021, pp. 1–6.

[9] C. Liu, R. Zhao, and Z. Shi, “Remote-sensing image captioning based on multilayer aggregated transformer,” IEEE Geosci. Remote Sens. Lett., vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2022.3150957.

[10] R. Shao, X.-J. Bi: Transformers Meet Small Datasets.

[11] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented transformer for speech recognition,” 2020, arXiv:2005.08100.

[12] J. Kim, M. El-Khamy, and J. Lee, “T-GSA: Transformer with Gaussian-weighted self-attention for speech enhancement,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2020, pp. 6669–6673.

[13] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst., Montreal, QC, Canada, vol. 27, Dec. 2014, pp. 3104–3112.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, M. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Hormann, and M. Möller, “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020, arXiv:2010.11929.

[15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4510–4520.

[17] K. He, “Identity mappings in deep residual networks,” in Proc. Eur. Conf. Comput. Vis. Amsterdam, The Netherlands, 2016, pp. 630–645.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE CVPR, Jul. 2017, pp. 4700–4708.

[19] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 843–852.

[20] N. Park and S. Kim, “How do vision transformers work?” 2022, arXiv:2202.06709.

[21] A. Ke, W. Ellsworth, O. Banerjee, A. Y. Ng, and R. Rajpurkar, “CheXtransfer: Performance and parameter efficiency of ImageNet models for chest X-ray interpretation,” in Proc. Conf. Health, Inference, Learn., Apr. 2021, pp. 116–124.

[22] A. Steinr, A. Kolesnikov, X. Zhai, R. Wightman, I. Uszkoreit, and L. Beyer, “How to train your ViT? Data augmentation, and regularization in vision transformers,” 2021, arXiv:2106.10270.

[23] Z. Dai, G. Lai, Y. Yang, and Q. V. Le, “Funnel-transformer: Filtering out sequential redundancy for efficient language processing,” 2020, arXiv:2006.03236.

[24] S. Mehta, M. Ghazvininejad, S. Iyer, L. Zettlemoyer, and H. Hajishirzi, “Effectiveness of data in deep learning era,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, no. 2, pp. 604–624, Feb. 2021, doi: 10.1109/ACCESS.2020.3026364.

[25] Z. Dai, G. Lai, Y. Yang, and Q. V. Le, “Delight: Deep and light-weight transformer,” 2020, arXiv:2008.06823.

[26] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, “LocalViT: Bringing locality to vision transformers,” 2021, arXiv:2104.05707.

[27] Z. Dai, H. Liu, Q. V. Le, and M. Tan, “CoAtNet: Marrying convolution and attention for all data sizes,” in Proc. Adv. Neural Inf. Process. Syst., vol. 34. Red Hook, NY, USA: Curran Associates, 2021, pp. 3965–3977.

[28] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu, “Incorporating convolution designs into visual transformers,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 579–588.
[28] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, “Bottleneck transformers for visual recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 16519–16529.

[29] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, “CV: Introducing convolutions to vision transformers,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 22–31.

[30] 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 Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 568–578.

[31] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in Proc. Adv. Neural Inf. Process. Syst., vol. 34, 2021, pp. 15908–15919.

[32] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. H. Tay, J. Feng, and S. Yan, “Tokens-to-token ViT: Training vision transformers from scratch on ImageNet,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 558–567.

[33] S. d’Ascoli et al., “Convit: Improving vision transformers with soft convolutional inductive biases,” in Proc. PMLR, 2021, pp. 2286–2296.

[34] 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 Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 10012–10022.

[35] Y. Liu, “Efficient training of visual transformers with small-size datasets,” in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 1–17.

[36] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNetv2: Efficient convolutional neural networks for mobile vision applications,” 2017, arXiv:1704.04861.

[37] D. Zhou, Q. Hou, Y. Chen, J. Feng, and S. Yan, “Rethinking bottleneck structure for efficient mobile network design,” in Proc. Eur. Conf. Comput. Vis. Glasgow, U.K., Aug. 2020, pp. 680–697.

[38] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” Dept. CS, Univ. Toronto, Toronto, ON, Canada, Tech. Rep. learning-features-2009-TR, Apr. 2009.

[39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[40] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms,” 2017, arXiv:1708.07747.

[41] P. Zhao, C. Li, M. M. Rahaman, H. Xu, P. Ma, H. Yang, H. Sun, T. Jiang, N. Xu, and M. Grzegorzek, “EMDS-6: Environmental microorganism image dataset sixth version for image denoising, segmentation, feature extraction, classification, and detection method evaluation,” Frontiers Microbiol., vol. 13, p. 1334, Apr. 2022, doi: 10.3389/fmicb.2022.829027.