Improving Vision Transformers to Learn Small-Size Dataset From Scratch

SEUNGHOON LEE, SEUNGHYUN LEE, AND BYUNG CHEOL SONG, (Senior Member, IEEE)
Department of Electronic Engineering, Inha University, Incheon 22212, South Korea
Corresponding author: Byung Cheol Song (bcsong@inha.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant by the Korea Government [Ministry of Science and ICT (MSIT)] (Artificial Intelligence Convergence Innovation Human Resources Development (Inha University)) under Grant RS-2022-00155915, and in part by the National Research Foundation of Korea (NRF) Grant Funded by the Korea Government (MSIT) under Grant 2022R1A2C2010095 and Grant 2022R1A4A1033549.

ABSTRACT This paper proposes various techniques that help Vision Transformer (ViT) to learn small-size datasets from scratch successfully. ViT, which applied the transformer structure to the image classification task, has outperformed convolutional neural networks, recently. However, the high performance of ViT results from pre-training using large-size datasets, and its dependence on large datasets comes from low locality inductive bias. And conventional ViT cannot effectively attend the target class due to redundant attention caused by a rather high constant temperature factor. In order to improve the locality inductive bias of ViT, this paper proposes novel tokenization (Shifted Patch Tokenization: SPT) using shifted patches and a position encoding (CoordConv Position Encoding: CPE) using $1 \times 1$ CoordConv. Also, to improve poor attention, we propose a new self-attention mechanism (Locality Self-Attention: LSA) based on learnable temperature and self-relation masking. SPT, CPE, and LSA are intuitive techniques, but they successfully improve the performance of ViT even on small-size datasets. We qualitatively show that each technique attends a more important area and contributes to having a flatter loss landscape. Moreover, the proposed techniques are generic add-on modules applicable to various ViT backbones. Our experiments show, when learning Tiny-ImageNet from scratch, the proposed scheme based on SPT, CPE, and LSA increases the accuracy of ViT backbones by +3.66 on average and up to +5.7. Also, the performance improvement of ViT backbones in ImageNet-1K classification, learning on COCO from scratch, and transfer learning on classification datasets verify that the generalization ability of the proposed method is excellent.

INDEX TERMS Vision transformer, attention mechanism, data efficient learning.

I. INTRODUCTION

Recently, lots researches have been reported that Vision Transformer (ViT) [1] showed better performance than Convolutional Neural Networks (CNNs), such as ResNet [2] and EfficientNet [3]. This made Transformer receive a lot of attention in the computer vision field. ViTs rarely use convolutional filters, i.e., the core of CNNs. Actually, ViTs employ them only for tokenization. Thus, ViTs structurally lack locality inductive bias than CNNs. So they require a huge amount of training data to obtain acceptable visual representation [4]. For example, to learn a small-size dataset, ViT had to precede pre-training on a large-size dataset such as JFT-300M [5]. However, building a large dataset is very expensive, so it is a difficult problem in the real world.

In order to alleviate the burden of pre-training, several ViTs which can learn a mid-size dataset such as ImageNet from scratch have been proposed. Such data-efficient ViTs tried to increase the locality inductive bias in terms of network architecture. For instance, some adopted a hierarchical structure like CNNs to leverage various receptive fields [6], [7], [8], and the others tried to modify the self-attention mechanism itself [6], [8], [9], [10], [11]. However, learning from scratch on mid-size datasets still requires significant costs. Moreover, if the domain gap between the pre-training dataset and the target dataset is large, transfer learning may not be an effective
strategy for learning a small-size dataset [12]. So we need to study ViT that can learn small-size datasets from scratch.

We suffer from two problems that limit the performance of ViT on small-size datasets. The first problem is the structure of the ViT, which is designed to have very little locality inductive bias. ViT divides a given image into non-overlapping patches of same size, and linearly projects each patch to a visual token. So, the tokenization of ViT has the permutation invariant property, which enables a good embedding of relations between patches [13]. However, these simple linear projections over non-overlapping patches do not sufficiently embed spatial relationships between adjacent pixels into each visual token. Also, there is no locality inductive bias in the transformer block that captures the relationship between the visual tokens. Also, conventional ViT never injects position information except when generating initial input tokens. This lack of position information makes it difficult for ViT to analyze high-dimensional positional relations between visual tokens and to learn effectively with a small amount of data [1], [15].

The second problem is the poor attention mechanism. The feature dimension of image data is usually much larger than that of signals in other domains such as NLP. In other words, image data with a large feature dimension has a much larger number of tokens than other domains. So, if the attention mechanism used in NLP is directly applied to such a large number of tokens as it is, the distribution of attention scores will be over-smoothed. Therefore, we face the problem that ViTs cannot attend locally to important visual tokens.

So in this paper, we propose several novel solutions to make ViT effectively learn small-size datasets from scratch. The first solution is Shifted Patch Tokenization (SPT) to further utilize spatial relations between neighboring pixels in the tokenization process. The idea of SPT was derived from Temporal Shift Module (TSM) [16]. TSM is effective temporal modeling which shifts some temporal channels of features. Inspired by this, we propose effective spatial tokenization that tokenizes patches together that are spatially shifted slightly but have the same semantic meaning. This has the effect of increasing the locality inductive bias by embedding more spatial information in each visual token. Second, we propose CoordConv Position Encoding (CPE) that injects position information inside the transformer block. CoordConv, that is an extension of convolutions that allows convolutional filters to consider spatial coordinates of pixels, can integrate a localization prior into the network architecture [17], [18]. Motivated from this property, we propose a method to replace the linear projections of a transformer block with $1 \times 1$ CoordConv [19]. Thereby, position information about tokens can be continuously encoded even inside the transformer blocks. Finally, we propose Locality Self-Attention (LSA), which allows ViT to attend locally. LSA mitigates the smoothing phenomenon of attention score distribution by excluding self-tokens and applying learnable temperature to the softmax function. LSA induces attention to work locally by forcing each token to focus more on tokens with large relation to itself.

Our proposed method can be very easily applied as add-on modules to various ViT backbones and helps ViT to learn strong representations. In Fig. 1, it is shown that the proposed method dramatically improves the performance of various ViT backbones when learning a small-size dataset from scratch. In particular, when SPT, CPE, and LSA are applied together, the accuracy of CoAtNet-0 [20] reaches 84.82%, i.e., increases by +1.41 in case of CIFAR-100, which is superior to the powerful CNN RegNetY-400MF [21]. Also, the proposed method improves the performance of ViT backbones in ImageNet-1K classification [22], learning on COCO [23] from scratch, and transfer learning on classification datasets because it has good generalization ability (see Sec. V-B and Sec. V-C). For example, In the case of ImageNet-1K, the proposed method increases the accuracy of
DeiT-Ti [24] and Swin-T [6] by +3.2 and +1.1, respectively. The main contribution points of this paper are as follows: noitemsep

- To solve problems that prevent ViTs from learning small datasets from scratch, that is, lack of locality inductive bias and over-smoothed attention, SPT, CPE, and LSA are presented by flattening loss landscapes. They can be easily applied to various ViT backbones and bring high-performance improvement for cost.
- We experimentally verify the generalization ability of the proposed method in various datasets, learning strategies, and tasks, and confirm that it is superior to CNNs compared to cost.

This paper is organized as follows: Sec. II reviews various approaches to improve ViT. And Sec. III describes the related preliminaries and then describes the formulations and structure of the proposed method in detail. In Sec. IV, the datasets, backbones, and training regime used in the experiments are explained. Sec. V shows quantitatively and qualitatively that the proposed method works well in various datasets and ViT backbones. Finally, Sec. VI concludes this study.

II. RELATED WORKS

Recently, several methods have been proposed to alleviate the dependence of ViT on large-size datasets. The previous ViTs could learn mid-size datasets from scratch. For example, DeiT [24] improved the efficiency of ViTs by employing data augmentations and regularizations, and then realized knowledge distillation by introducing the distillation token concept. T2T [14] proposed a method to tokenize overlapping patches by applying a transformer block. This made it possible to learn local structure information around a token. PiT [7] produced various receptive fields through spatial dimension reduction based on the pooling structure of a convolutional layer. CvT [8] replaced both linear projection and multi-layer perceptron with convolutional layers. Also, like PiT, CvT generated various receptive fields with a convolutional layer. Swin Transformer [6] presented an efficient hierarchical transformer that gradually reduces the number of tokens through patch merging while using attention of local windows. CaiT [9] employed LayerScale, which converges well even in training ViTs with a large depth. In addition, the transformer layer of CaiT is divided into a patch-attention layer and a class-attention layer, which is effective for class embedding. CoAtNet [20] is a hybrid model that properly combines only the strengths of CNN and Transformer. It unified depth-wise convolution and self-attention through relative attention, and effectively improved generalization by stacking convolutional blocks in front of transformer blocks.

Position encoding is important for transformer to capture sequence ordering of input tokens. Wu et al. [15] proposed a new position encoding called image Relative Position Encoding (iRPE). Since iRPE considers interactions between queries and relative position within the self-attention mechanism, it accomplished effective position encoding.

Park and Kim [25] demonstrated that strong representations can be learned by smoothing the loss landscape of ViT. For example, it was shown that increasing the locality inductive bias of ViT through local attention flattened the loss landscape and improved generalization ability.

As such, various efforts have been made to resolve ViT’s data dependence. However, learning small-size datasets from scratch is still challenging. Therefore, this paper presents several solutions which allow various ViT backbones to effectively learn small-size datasets from scratch by flattening loss landscapes.

III. METHOD

This section describes three key ideas helping ViT to learn small-size datasets: SPT, CPE and LSA. Also, the optimal structure of each technique is explored. Fig. 2(a) depicts the concept of SPT. First, SPT spatially shifts an input image in several directions and concatenates them with the input image. This figure is an example of shifting in four diagonal directions. Next, patch partitioning is applied like standard ViT. Then, for embedding into visual tokens, three processes are sequentially performed: patch flattening, layer normalization [26], and linear projection. Through this, patches with the same semantic meaning but slightly shifted spatially can be embedded together in visual tokens. As a result, SPT can utilize more spatial information and increase the locality inductive bias of ViT. Fig. 2(b) explains the second idea, i.e., CPE. CPE first flattens additional feature maps with coordinate information, e.g., x- and y-coordinates, and concatenates the input visual token and reshaped feature maps, and then applies linear projection. Since CPE linearly projects the coordinate information together, it can realize effective 2D position encoding for visual tokens. Fig. 2(c) describes LSA. In general, a softmax function can control the smoothness of attention scores distribution through temperature scaling [27]. LSA primarily sharpens the distribution of attention scores by learning the temperature parameters of the softmax function. Additionally, the self-token relation is removed by applying the so-called diagonal masking, which forcibly suppresses the diagonal components of the similarity matrix computed by Query and Key. This masking relatively increases the attention scores between different tokens, making the distribution of attention scores sharper. As a result, LSA improve ViT by inducing locally-focused attention.

A. PRELIMINARY

Prior to a detailed description of SPT, CPE, and LSA, this section briefly reviews the tokenization, CoordConv, and formulation of the self-attention mechanism of standard ViT.

1) TOKENIZATION

Let \( \mathbf{x} \in \mathbb{R}^{H \times W \times C} \) be an input image. Here, \( H, W, \) and \( C \) indicate the height, width, and channel of the image, respectively. First, ViT divides the input image into non-overlapping patches and flatten the patches to obtain a sequence of
FIGURE 2. Architectures of the proposed techniques. (a) SPT where $N_T$ and $d$ are the number of visual tokens and the token dimension. (b) CPE where $H$ and $W$ are the height and width of feature map, i.e., $N_T = H \times W$. (c) LSA where the graph at each stage presents the attention score distribution when using softmax. Here, blue blocks are newly added parts in this paper.

vectors. This process is formulated as Eq. (1)

$$P(x) = [x_1^p; x_2^p; \ldots; x_N^p]$$

where $x_i^p \in \mathbb{R}^{P \times C}$ represents the $i$-th flattened vector. $P$ and $N (=HW/P^2)$ stand for the patch size and the number of patches, respectively.

Next, ViT perform patch embedding by linearly projecting each vector into the space of the hidden dimension of the transformer encoder. Each patch embedding corresponds to a visual token that is input to the transformer encoder, so this series of processes is called tokenization ($T$). This is defined by

$$T(x) = P(x)E_t$$

where $E_t \in \mathbb{R}^{(P^2 \cdot C) \times d}$ is the learnable linear projection for tokens, and $d$ is the hidden dimension of the transformer encoder. This operation is exactly the same as for a non-overlapping convolutional layer, and this simple method does not have sufficient locality inductive bias. In Sec. III-B, we propose SPT to leverage rich spatial information by increasing the receptive field of tokenization.

2) CoordConv

CoordConv [19] injected additional coordinate information into the convolutional layer. Coordinate information is defined as the x- and y-coordinates of each pixel in the feature map, i.e., $c_x \in \mathbb{R}^{H \times W \times 1}$ and $c_y \in \mathbb{R}^{H \times W \times 1}$ and is scaled to have a range of [-1, 1]. And, the coordinate channels are channel-wise concatenated with the feature map $x \in \mathbb{R}^{H \times W \times C}$ and then input to the convolutional layer. The position information of each pixel is explicitly encoded in the feature map, providing a localization prior to the neural network. Since the increased computational cost is only an operation for two additional channels, CoordConv is very computationally efficient. Sec. III-C proposes CPE replacing ViT’s linear projections with $1 \times 1$ CoordConv. CPE effectively improves the performance of ViT backbones by increasing the locality inductive bias inside the transformer blocks.

3) ATTENTION MECHANISM

Meanwhile, the self-attention mechanism of standard ViT operates as follows. First, learnable linear projections are applied to each token to obtain Query, Key, and Value. Next, calculate the similarity matrix, that is, $R \in \mathbb{R}^{(N+1) \times (N+1)}$, indicating the semantic relation between tokens through the dot product operation of Query and Key. The diagonal components of $R$ represent self-token relations, and the off-diagonal components represent inter-token relations:

$$R(x) = xE_q(xE_k)^T$$

Here, $E_q \in \mathbb{R}^{d_q \times d}$, $E_k \in \mathbb{R}^{d \times d_k}$ indicate learnable linear projections for Query and Key, respectively. And, $d_q$ and $d_k$ are the dimensions of Query and Key, respectively. Next, $R$ is divided by the square root of the Key dimension, and then the softmax function is applied to obtain the attention score matrix. Finally, calculate the self-attention, defined by the dot product:

$$V(x) = \sum_{i=1}^{N+1} x_i^p \cdot \text{softmax}(R(x))_{i}$$

where $x_i^p \in \mathbb{R}^{P \times C}$ is the $i$-th flattened token.
product of the attention score matrix and Value, as in Eq. (4):

$$ S_{\text{A}}(x) = \text{softmax}(R/\sqrt{d_k}) x E_v $$

where $E_v \in \mathbb{R}^{d_x \times d_k}$ is a learnable linear projection of Value, and $d_v$ is the Value dimension.

Eq. (4) was designed so that the attentions of tokens with large relations get large. However, due to the following two causes, attentions of standard ViT tend to be similar to each other regardless of relations. The first cause is that since $x E_q$ and $x E_k$ are linearly projected from the same input tokens, token vectors belonging to Query and Key tend to have similar values. Eq. (3) shows that $R$ is the dot product of Query and Key. So, self-token relations which are dot products of similar vectors are usually larger than inter-token relations. Therefore, the softmax function of Eq. (4) gives relatively high scores to self-token relations and small scores to inter-token relations. The second cause is that dividing $R$ by $\sqrt{d_k}$ in Eq. 4 acts as a high temperature scaling for softmax. Table 1 shows that the attention scores smoothed by high temperature degrade the performance of ViT. Here, we can achieve the best performance when the softmax temperature is less than $\sqrt{d_k}$. In Sec. III-D, we propose the LSA for improving the performance of ViT by solving the smoothing problem of the attention scores distribution.

**TABLE 1.** Top-1 accuracy according to temperature. Here, ViT was trained from scratch on CIFAR100 and Tiny-ImageNet. Detailed learning related matters are described in Sec. IV. Gray highlight means the baseline model.

| Temperature       | 0.25$\sqrt{d_k}$ | 0.5$\sqrt{d_k}$ | $\sqrt{d_k}$ | 2$\sqrt{d_k}$ | 4$\sqrt{d_k}$ |
|-------------------|------------------|------------------|--------------|--------------|--------------|
| CIFAR100          | 73.70            | 74.54            | 73.81        | 72.77        | 71.55        |
| Tiny-ImageNet     | 57.62            | 57.65            | 57.07        | 56.98        | 56.43        |

**B. SHIFTED PATCH TOKENIZATION**

1) **FORMULATION**

First, an input image is spatially shifted by half the patch size in four diagonal directions, that is, left-up, right-up, left-down, and right-down. In this paper, this shifting strategy is named $S$ for convenience, and the SPT of all experiments follows $S$. Of course, various shifting strategies other than $S$ are available, and they are dealt with in the Fig. 3 and Table 2-3. Next, the shifted features are cropped to the same size as the input image and then concatenated with the input. Then, the concatenated features are divided into non-overlapping patches and the patches are flattened as in Eq. (1). Next, visual tokens are obtained through layer normalization (LN) and linear projection. The whole process is formulated as Eq. (5):

$$ S(x) = \text{LN}(P([x s_1 s_2 \ldots s_{N_S}])) E_S $$

where $s_i \in \mathbb{R}^{H \times W \times C}$ represents the $i$-th shifted image according to $S$ and $E_S \in \mathbb{R}^{(P^2 - C(N_s+1) \times d_S)}$ indicates a learnable linear projection. Also, $d_S$ represents the hidden dimension of the transformer encoder, and $N_S$ represents the number of images shifted by $S$. In Eq. (5), an additional tokenization scheme [14], [20] can be applied after the concatenation process of shifted features.

2) **PATCH EMBEDDING LAYER**

This section describes how to use SPT as a patch embedding layer. We concatenate a class token to visual tokens and then add position encoding. Here the class token is the token with representation information of the entire image, and the position encoding gives position information to the visual tokens. If a class token is not used, only position encoding is added to the output of SPT. How to apply the SPT to the patch embedding layer is formulated as follows:

$$ S_{\text{pe}}(x) = \begin{cases} [x_{\text{cls}}, S(x)] + E_{\text{pos}} & \text{if } x_{\text{cls}} \text{ exist} \\ S(x) + E_{\text{pos}} & \text{otherwise} \end{cases} $$

where $x_{\text{cls}} \in \mathbb{R}^{d_S}$ is a class token and $E_{\text{pos}} \in \mathbb{R}^{(N+1) \times d_S}$ is the learnable position encoding. Also, $N$ is the number of embedded tokens in Eq. (5).

3) **POOLING LAYER**

Tokenization is the process of embedding 3D-tensor features into 2D-matrix features. For example, it embeds $x \in \mathbb{R}^{H \times W \times C}$ into $y = T(x) \in \mathbb{R}^{N \times d}$. Since $N = HW/P^2$, the spatial size of the 3D feature is reduced by $P^2$ through the tokenization process. So, if tokenization is used as a pooling layer, the number of visual tokens can be reduced. Therefore, we propose to use SPT as a pooling layer as follows: First, class tokens and visual tokens are separated, and visual tokens in the form of 2D-matrix are reshaped into 3D-tensor with spatial structure, i.e., $R : \mathbb{R}^{N \times d} \rightarrow \mathbb{R}^{(H/P) \times (W/P) \times d}$. Then, if the SPT of Eq. (5) is applied, new visual tokens with a reduced number of tokens are embedded. Finally, the linearly projected class token is connected with the embedded visual tokens. If there is no class token, only $R$ is applied before the output of SPT. The whole process is formulated as Eq. (7):

$$ S_{\text{pool}}(y) = \begin{cases} [x_{\text{cls}}, E_{\text{cls}}; S(R(y))] & \text{if } x_{\text{cls}} \text{ exist} \\ S(R(y)) & \text{otherwise} \end{cases} $$

where $E_{\text{cls}} \in \mathbb{R}^{d_x \times d_S}$ is a learnable linear projection. In addition, $d'_{S}$ is the hidden dimension of the next stage. As a result, SPT embeds rich spatial information into visual tokens by increasing the receptive field of tokenization as much as spatially shifted.
TABLE 2. Top-1 accuracy (%) of various shift directions. ViT was trained from scratch on CIFAR10 (C-10), CIFAR100 (C-100), SVHN, and Tiny-ImageNet (TINY).

| Directions   | C-10   | C-100  | SVHN   | TINY   |
|--------------|--------|--------|--------|--------|
| 4 Cardinal   | 94.44  | 76.29  | 97.87  | 60.35  |
| 4 Diagonal   | 94.33  | 76.29  | 97.86  | 60.67  |
| 8 Cardinal   | 94.41  | 76.40  | 97.81  | 60.57  |

4) CHOICE OF SHIFT STRATEGY
SPT can adopt a wide variety of spatial shift strategies. However, in this paper, only the spatial shift that can be easily defined by the shift direction and the shift ratio is considered. We examined the following three shift directions. The first is the 4 cardinal directions consisting of up, down, left, and right directions (Fig. 3(a)). The second is 4 diagonal directions including up-left, up-right, down-left, and down-right (Fig. 3(b)). The last is the 8 cardinal directions including all the preceding directions (Fig. 3(c)).

Table 2 shows top-1 accuracy in small-size datasets such as CIFAR10, CIFAR100, SVHN, and Tiny-ImageNet for each shift direction. This experiment applied SPT to standard ViT. 4 cardinal directions showed the best performance in CIFAR10 and SVHN. On the other hand, 4 diagonal directions and 8 cardinal directions provided the best performance in CIFAR100 and Tiny-ImageNet, respectively. This shows that the shift direction is somewhat dependent on the dataset. For example, in CIFAR10 or CIFAR100, the target class tends to be in the center of the image, whereas datasets do not. The location of the target class has some degree of correlation with the shift direction, and the correlation can affect the performance. However, since the performance difference was not significant, in this paper, the shift direction in the experiment was fixed to 4 diagonal directions.

Next, we look at various shift ratios. The degree of image shifting in SPT is defined as follows: \( \text{SHIFT} = P \times r_{shift} \), where \( P \) represents the patch size, and \( r_{shift} \) represents the shift ratio. Table 3 shows the performance according to shift ratio for CIFAR100, Tiny-ImageNet, and ImageNet-1K. In this experiment, a model with SPT applied to standard ViT was used, and 4 diagonal directions were adopted. In CIFAR100 and ImageNet-1K, a ratio of 0.5 was the best, and in Tiny-ImageNet, a ratio of 0.25 was the best. This experimental result shows that the optimal shift ratio also depends on the datasets. Since the relatively most reasonable shift ratio is 0.5 according to our experiment, all the experiments in this paper fixed the shift ratio to 0.5.

Note that more various shifting strategies will be available in addition to the methods considered here. The exploration of optimal shifting strategy according to datasets remains future work.

5) THE POOLING LAYER ALSO NEEDS SHIFTED PATCHES
Patch embedding is the process of converting a large image into a small number of visual tokens, very similar to pooling. However, in this paper, only the spatial shift that can be easily defined by the shift direction and the shift ratio is considered. We examined the following three shift directions. The first is the 4 cardinal directions consisting of up, down, left, and right directions (Fig. 3(a)). The second is 4 diagonal directions including up-left, up-right, down-left, and down-right (Fig. 3(b)). The last is the 8 cardinal directions including all the preceding directions (Fig. 3(c)).

Table 2 shows top-1 accuracy in small-size datasets such as CIFAR10, CIFAR100, SVHN, and Tiny-ImageNet for each shift direction. This experiment applied SPT to standard ViT. 4 cardinal directions showed the best performance in CIFAR10 and SVHN. On the other hand, 4 diagonal directions and 8 cardinal directions provided the best performance in CIFAR100 and Tiny-ImageNet, respectively. This shows that the shift direction is somewhat dependent on the characteristics of datasets. For example, in CIFAR10 or CIFAR100, the target class tends to be in the center of the image, whereas datasets do not. The location of the target class has some degree of correlation with the shift direction, and the correlation can affect the performance. However, since the performance difference was not significant, in this paper, the shift direction in the experiment was fixed to 4 diagonal directions.

Next, we look at various shift ratios. The degree of image shifting in SPT is defined as follows: \( \text{SHIFT} = P \times r_{shift} \), where \( P \) represents the patch size, and \( r_{shift} \) represents the shift ratio. Table 3 shows the performance according to shift ratio for CIFAR100, Tiny-ImageNet, and ImageNet-1K. In this experiment, a model with SPT applied to standard ViT was used, and 4 diagonal directions were adopted. In CIFAR100 and ImageNet-1K, a ratio of 0.5 was the best, and in Tiny-ImageNet, a ratio of 0.25 was the best. This experimental result shows that the optimal shift ratio also depends on the datasets. Since the relatively most reasonable shift ratio is 0.5 according to our experiment, all the experiments in this paper fixed the shift ratio to 0.5.

Note that more various shifting strategies will be available in addition to the methods considered here. The exploration of optimal shifting strategy according to datasets remains future work.

So, SPT for improving tokenization can also be applied to the pooling layer. We experimented to verify that SPT was effective in both patch embedding and pooling layer, and Swin-T of a hierarchical structure was used for this experiment. Table 4 shows that SPT improves performance not only in the patch embedding layer but also in the pooling layer. Especially, it achieved a significant accuracy increase of +4.91 in TINY. Thus, we apply SPT to all pooling layers of ViT backbones with a hierarchical structure.

C. CoordConv POSITION ENCODING

1) FORMULATION
First, we flatten \( \mathbf{c}_x \) and \( \mathbf{c}_y \) so that they have the same shape as visual tokens \( \mathbf{x} \in \mathbb{R}^{HW \times d'} \). Then, we concatenate \( \mathbf{x}, \mathbf{c}_x \) and \( \mathbf{c}_y \), and then apply linear projection \( \mathbf{E}_c \in \mathbb{R}^{d \times d'} \). Here, \( d' \) is the dimension of output tokens. As in Eq. (7), the class-token is linearly projected separately and then concatenated with the output visual tokens.

\[
\mathbf{C}(\mathbf{x}) = \begin{cases} 
[\mathbf{x}_{cls} \mathbf{E}_{cls} \mid \mathbf{x} \mathbf{F}(\mathbf{c}_x) \mathbf{F}(\mathbf{c}_y)] \mathbf{E}_c & \text{if } \mathbf{x}_{cls} \text{ exist} \\
[\mathbf{x} \mathbf{F}(\mathbf{c}_x) \mathbf{F}(\mathbf{c}_y)] \mathbf{E}_c & \text{otherwise}
\end{cases}
\]  

2) WHERE TO APPLY
Linear projection in a transformer block is utilized in Query, Key, and Value projections of the multi-head self-attention layer, as well as in output projections that merge multi-head attention outputs. And the FNN’s channel expansion and compression also uses linear projection. CPE can be applied to all of these locations and improve the locality inductive bias of the transformer block through additional coordinate information embedding. In this paper, CPE is applied to both multi-head self-attention and FFN according to the results of the Table 5.

3) CPE CAN IMPROVE BOTH MSA AND FFN
As a layer to replace the linear projection, CPE can be applied to the linear projections inside the Multi-head Self-Attention layer (MSA) and the linear projections of the FFN. Table 5
than attending to its own tokens. The proposed diagonal makes ViT’s attentions more focused on other tokens rather than the softmax operation. Specifically, diagonal masking forces inter-token relations by fundamentally excluding self-token maskings from the diagonal components of R. This allows ViT to determine the softmax temperature by itself during the learning process. Based on Eq. (4), the LSA with both diagonal masking and learnable temperature scaling applied is defined by:

\[ R_{i,j}^M(x) = \begin{cases} R_{i,j}(x) & (i \neq j) \\ -\infty & (i = j) \end{cases} \]  

(9)

where \( R_{i,j}^M \) indicates each component of the masked similarity matrix.

2) LEARNABLE TEMPERATURE SCALING

The second step for LSA is the learnable temperature scaling, which allows ViT to determine the softmax temperature by itself during the learning process. Based on Eq. (4), the LSA with both diagonal masking and learnable temperature scaling applied is defined by:

\[ L(x) = \text{softmax}(R_{i,j}^M(x)/\tau)xE_v \]  

(10)

where \( \tau \) is the learnable temperature. LSA improves the performance of ViT by allowing it to focus more on tokens that are related to it through diagonal masking and learnable temperature.

3) SHARING TEMPERATURE ACROSS HEADS

In general, the Transformer block uses a MSA that uses multiple self-attentions [28]. Therefore, all heads can share the same temperature or have them individually. Table 7 compares the performance of ViT when the learnable temperature of each attention head is shared and when it is not shared. When the temperature was individually learned in each head, there was a performance improvement of +0.26 and +1.33 for CIFAR-100 and Tiny-ImageNet, respectively. Therefore, we can find that the attention head focusing on different features has meaning only when it learns the appropriate temperature for each.

4) LEARNING LOW TEMPERATURE

As seen in Table 1, the well-learned temperature should be less than \( \sqrt{d_k} \). Fig. 4 shows the average learned temperature according to depth. Note that the average learned temperature is lower than the constant temperature of standard ViT. In general, the low temperature of softmax sharpens the score distribution. Therefore, the learnable temperature scaling sharpens the distribution of attention scores.

5) RESOLVING SMOOTHING PROBLEM

LSA solves the smoothing problem of the attention score distribution. Fig. 5 shows the depth-wise averages of total Kullback-Leibler divergence (\( D_{KL}^{\text{total}} \)) for all heads. Here, T and M mean that only learnable temperature scaling and
diagonal masking is applied to ViTs, respectively, and L indicates that the entire LSA is applied to ViTs. The lower the average of $D_{\text{total}}^{\text{KL}}$, the flatter the attention score distribution. We can find that when LSA is fully applied, the average of $D_{\text{total}}^{\text{KL}}$ is larger by about 0.5 than standard ViT, so LSA attenuates the smoothing phenomenon of the attention score distribution.

6) BOTH MASKING AND LEARNABLE TEMPERATURE ARE NEEDED
Let’s look at the effect of learnable temperature scaling (T) and diagonal masking (M), two key elements of LSA (L), on overall performance. Table 8 shows that learnable temperature scaling and diagonal masking effectively resolves the smoothing phenomenon of attention score distribution (see Fig. 5). For example, learnable temperature scaling and diagonal masking in TINY improved performance by +0.88 and +1.22, respectively. Considering that the LSA applied with both techniques shows a performance improvement of +1.43, we can claim that the contribution of each is sufficiently large and the two techniques produce a synergy.

E. WHY DOES THE PROPOSED METHOD WORK?
SPT further facilitates the analysis of spatial correlation by tokenizing shifted patches of the same semantic meaning together. This indicates that locality inductive bias is effectively improved. And CPE embeds spatial coordinates, one of the critical image locality information, into each token. This also improves the locality inductive bias. The improved locality inductive bias through SPT and CPE allows ViT to learn a strong representation despite little data [25]. In addition, attention acts more locally in transformers that have learned strong representation. Through learnable temperature scaling and diagonal masking, LSA’s attention focuses more on meaningful regions of the feature map (see fig. 6). In other words, it makes it learn strong representation to improve the performance of ViTs.

IV. EXPERIMENT SETUP
This section describes the setup of experiments to verify the performance of the proposed method. First, the datasets used in this study are described, and then the detailed configuration of ViT backbones is given. Finally, the detailed regime in which each model was trained is described.

A. DATASET
For small-size dataset experiments, CIFAR-10, and CIFAR-100 (C-10 and C-100) [29], SVHN [30], and TinyImageNet (TINY) [31] were employed. For the mid-size dataset experiment, ImageNet-1K [22] was employed. In addition, experiments were conducted on COCO 2017 [23] to verify the extendibility of the proposed method to other tasks. Table 9 represents the train set size, test set size, and the number of classes of the datasets used in this paper.

B. BACKBONE
In order to achieve high performance on a small-size dataset, we have to limit the capacity of ViT by adjusting model hyperparameters. Since small-size datasets generally have small resolution, the number of tokens in the backbones designed for ImageNet should also be adjusted. So, we set the number of visual tokens in the transformer block to $8 \times 8$. If a pooling layer is used in the middle of the transformer, the number of tokens in the intermediate stage is $8 \times 8$. Here, C-10, C-100, and SVHN all have a resolution of $32 \times 32$,

| Model       | ViT   | T-ViT | M-ViT | L-ViT |
|-------------|-------|-------|-------|-------|
| CIFAR100    | 73.81 | 74.35 | 74.34 | 74.87 |
| Tiny-ImageNet| 57.07 | 57.95 | 58.29 | 58.50 |

TABLE 9. Detailed specifications of the datasets.
and TINY has a resolution of $64 \times 64$. So, in the case of TINY, we double the patch size compared to other datasets. Alternatively, we control the number of tokens by adding a pooling layer. On the other hand, mid-size datasets such as ImageNet-1K and COCO were trained according to popular configurations. The detailed configuration of each model for the small-size dataset is as follows.

1) ViT
In a standard ViT [1] with a high risk of overfitting, the depth was set to 9, the hidden dimension was set to 192, and the number of heads was set to 12. The patch size in TINY is 8, and the patch size in the other datasets is 4.

2) PiT
Based on the PiT-XS [7] architecture, the patch size was set to 4 in TINY and was set to 2 in the remaining datasets.

3) T2T
We designed based on T2T-14 [14] using Tokens-to-Token modules with a kernel size of 3, the stride of 2, and padding of 1. In TINY, the depth of the Tokens-to-Token module was 3, and it was 2 in the rest of the datasets. And to apply SPT to T2T using transformer-based tokenization, we concatenate shifted images before the Tokens-to-Token module of the first stage.

4) CaiT
Referring to CaiT-XS24 [9] and CaiT-XXS24 [9], CaiT set the patch size to 4 in TINY and 2 in the other datasets.

5) SWIN TRANSFORMER
Swin Transformer uses Swin-T [6] and Swin-S [6] as backbones. For all datasets, the patch size was set to 2 and the window size was set to 4. Note that conventional Swin-T and Swin-S go through 3 pooling layers because they are composed of 4 stages. So, since the image with a resolution of $32 \times 32$ has $2 \times 2$ resolution in the last stage, a window attention size of 4 cannot be applied. Therefore, we modified the models to have three stages in C-10, C-100, and SVHN. That is, Swin-T consisted of 3 stages of 2, 6, and 4 depth, and Swin-S consisted of 3 stages of 3, 6, 12 depth.

6) CoAtNet
CoAtNet used CoAtNet-0 [20] as a backbone. In TINY, the pooling layer was used from the second stage, and the pooling layer was used from the third stage in the rest of the datasets.

C. TRAINING REGIME
In the small-size dataset training, Basically, the following learning strategies are followed. We applied data augmentations such as CutMix [32], Mixup [33], Auto Augment [34], Repeated Augment [35] to all models. In addition, regularization techniques such as label smoothing [36], stochastic depth [37], and random erasing [38] were employed. And, AdamW [39] was used as the optimizer. Weight decays were set to 0.05, batch size to 128, and warm-up to 10. All models were trained for 100 epochs, and cosine learning rate decay was used. The initial learning rate was set to 0.003 in ViT and CNN, and 0.001 in the rest of the models. ImageNet was trained at $224 \times 224$ resolution, and a batch size of 256 was used. The rest of the training regimes followed [6] as it is. To learn COCO from scratch, we referred to Direct(P1x)-1x pipeline of [40] where a detection dataset was trained from scratch.

V. MAIN RESULTS
A. ABALATION STUDY
This section deals with ablation studies for SPT, CPE, and LSA. First, we show that the redundant attention of each technique is improved through the visualization of class token attention. Next, we demonstrate that each technique flattens the loss landscapes through 3D loss landscapes visualization. And we quantitatively shows that the performance of ViT has improved thanks to the proposed techniques. For this experiments, two small-size datasets, i.e., C-100 and TINY are used, and the backbone are ViT [1] and Swin-T [6]. The gray highlight in each table indicates the baseline model.

1) ATTENTION ONLY TO IMPORTANT LOCATIONS
Fig. 6 visualizes the attention scores of the final class token when SPT, CPE, and LSA were applied to ViT. First, when the LSA is applied, the attentions scattered in the background are more concentrated on the target class. Next, when either SPT or CPE is applied, the attention can better capture the shape of target class. Finally, when all of SPT, CPE, and LSA are applied, the shape of the class is maintained and ViT is focused on the more important region. Looking at the first example of the figure (top row), the dog is more accurately discriminated because the facial region that is crucial for distinguishing the dog is attended the most. Therefore, we can find that the proposed method effectively increases the locality inductive bias and induces the attention of ViTs to improve.

2) THE EFFECT OF FLATTENING THE LOSS LANDSCAPE
Fig. 7 visualizes the two-dimensional loss contour according to the application of SPT, CPE, and LSA. Here, the visualization method of Li et al. [45] was adopted. We can observe that SPT and CPE significantly flatten the ViT’s loss landscape. This proves that in ViT, which structurally lacks locality information of tokens, spatial shift modeling of visual tokens and position encoding improves locality inductive bias very effectively. Also, Table 10 quantitatively shows that SPT and CPE have a huge effect on improving the performance of ViT. On the other hand, LSA has an insignificant effect of flattening the loss landscape of ViT compared to CPE and SPT. Table 10 supports that LSA has the smallest influence on the performance improvement of ViT in both CIFAR100 and Tiny-ImageNet. Note that the loss landscape is most...
FIGURE 6. Average attention score of class tokens of all attention heads in ViT’s last translator block for validation set of Tiny-ImageNet. The upper-left box shows the predicted class and confidence score of each model. Gray and white boxes indicate that the prediction is incorrect and correct, respectively.

FIGURE 7. Change of loss contour of ViT according to the application of the proposed method. CIFAR-100 was used for this experiment. We can see that SPT and CPE, which have a great influence on performance improvement, flatten the loss landscape of ViT more clearly. Also, flattening is most pronounced when SPT, CPE, and LSA are applied.

effectively flattened when SPT, CPE, and LSA are all applied. Actually, Table 10 proves this fact. Thus, Fig. 7 and Table 10 show that the more flattened the loss landscape, the better the performance of ViT. In other words, the proposed method effectively flattens the loss landscape, which in turn helps ViT learn strong representations.

3) SPT, CPE, AND LSA HAVE SYNERGY EFFECT
Table 10 verifies the synergy effect of SPT, CPE and LSA. The best synergy, i.e., the highest performance improvement, was observed when both were utilized simultaneously. That is, when all techniques were applied, the performance improvement of ViT reached a whopping +4.69 and +5.08 for C-100 and TINY, and that of Swin-T reached +3.62 and +5.7 for C-100 and TINY, respectively. Therefore, we argue that SPT, CPE, and LSA designed to overcome the drawbacks of ViTs are effective add-on modules.

B. IMAGE CLASSIFICATION
This section shows that the proposed techniques effectively improve the performance of ViT backbones even in various datasets and tasks. First, we prove that the performance of ViTs is dramatically improved when data is insufficient. This shows that the proposed method improves ViT comparable to CNNs even if there is not much data. Second, we demonstrate that the proposed methods is also effective in transfer learning and a mid-size dataset, i.e., ImageNet-1K. This supports the generalization of the proposed method. Finally, we show that the proposed method improves the performance of ViT backbone in object detection and instance segmentation tasks. This indicates that the proposed method works well not only for image classification but also for other vision tasks.

1) LEARNING SMALL-SIZE DATASETS FROM SCRATCH
Table 11 shows the performance improvement when the proposed method was applied to various ViT backbones. In all
TABLE 10. Ablation study of the application of data augmentations, SPT, CPE, and LSA. Top-1 accuracy is measured in CIFAR100 and Tiny-ImageNet.

| Model Configuration | Data Aug. | SPT | CPE | LSA | CIFAR100 | Tiny-ImageNet |
|---------------------|-----------|-----|-----|-----|----------|--------------|
| ViT                 | ✓         | ✓   | ✓   | ✓   | 60.01    | 73.81        |
|                     |           |     |     |     | 73.61    | 76.00        |
|                     |           |     |     |     | 74.87    | 78.14        |
|                     |           |     |     |     | 78.50    |              |
| Swin-T              | ✓         | ✓   | ✓   | ✓   | 65.17    | 77.11        |
|                     |           |     |     |     | 80.26    | 77.75        |
|                     |           |     |     |     | 77.48    | 80.45        |
|                     |           |     |     |     | 80.73    |              |

TABLE 11. Top-1 accuracy when learning small-size datasets from scratch. Tiny-ImageNet was used here, and throughput was measured on V100 GPU.

| Method | Throughput | FLOPs | Params. | C-10 | C-100 | SVHN | TINY |
|--------|------------|-------|---------|------|-------|------|------|
| CNN    |            |       |         |      |       |      |      |
| ResNet110 [2] | 2479   | 1.02G | 1.7M    | 96.37 | 79.86 | 97.85 | 62.96 |
| EfficientNet-B0 [41] | 5979   | 0.12G | 3.7M    | 94.66 | 76.04 | 97.22 | 66.79 |
| EfficientNetV2-S [42] | 3139   | 0.95G | 20.7M   | 96.41 | 79.89 | 97.83 | 65.31 |
| RegNetY-200MF [21] | 2144   | 0.91G | 2.4M    | 97.02 | 82.40 | 97.95 | 71.25 |
| RegNetY-400MF [21] | 1359   | 1.89G | 5.8M    | 97.48 | 83.51 | 98.10 | 72.89 |
| ViT    |            |       |         |      |       |      |      |
| ViT [1] | 8992     | 0.19G | 2.8M    | 93.58 | 73.81 | 97.82 | 57.07 |
| T2T-14 [14] | 3823   | 0.65G | 6.7M    | 95.30 | 77.00 | 97.90 | 60.57 |
| CaiT-XXS24 [9] | 3830   | 0.60G | 9.1M    | 94.91 | 76.89 | 98.13 | 64.37 |
| CaiT-XXS24 [9] | 3188   | 0.87G | 13.6M   | 93.99 | 76.59 | 98.16 | 65.56 |
| PiT-XS [7] | 8361     | 0.28G | 7.1M    | 94.24 | 74.99 | 97.83 | 60.25 |
| Swin-T [6] | 2641     | 0.98G | 19.1M   | 94.46 | 77.11 | 97.72 | 64.27 |
| Swin-S [6] | 1630     | 1.90G | 33.3M   | 95.78 | 78.69 | 98.05 | 66.20 |
| CoAtNet-0 [20] | 4329   | 1.51G | 10.4M   | 97.40 | 83.41 | 97.93 | 68.78 |
| Ours    |            |       |         |      |       |      |      |
| SCL-ViT | 6887     | 0.20G | 5.4M    | 95.41 | 78.50 | 97.87 | 62.15 |
| SCL-T2T-14 | 2802   | 0.68G | 13.0M   | 96.35 | 78.95 | 97.99 | 64.31 |
| SCL-CaiT-XXS24 | 3237   | 0.61G | 9.3M    | 96.20 | 81.34 | 98.20 | 67.27 |
| SCL-CaiT-XXS24 | 2575   | 0.89G | 13.9M   | 96.44 | 81.62 | 98.24 | 67.95 |
| SCL-PiT-XS | 4999     | 0.33G | 14.8M   | 96.47 | 80.21 | 97.93 | 63.92 |
| SCL-Swin-T | 1909    | 1.22G | 25.3M   | 96.29 | 80.73 | 98.00 | 69.97 |
| SCL-Swin-S | 1234    | 2.14G | 39.6M   | 96.52 | 81.68 | 98.07 | 70.30 |
| SCL-CoAtNet-0 | 2779 | 1.72G | 15.0M   | 97.50 | 84.82 | 98.03 | 70.05 |

TABLE 12. Results of learning COCO from scratch. Mask-R CNN [43] was trained with Direct(P1x)-1x pipeline of [40] using only COCO. FPS was measured on V100 GPU.

| Pre-training | Backbone | FPS | Params. | AP_{box}^S | AP_{box}^L | AP_{box}^M | AP_{mask}^S | AP_{mask}^L | AP_{mask}^M |
|--------------|----------|-----|---------|-------------|------------|------------|-------------|------------|------------|
| ImageNet-1K  | X101-64 [44] | 8.8 | 102M    | 43.3        | 64.3       | 47.4       | 38.8        | 61.3       | 41.7       |
| Direct(P1x)  | Swin-T   | 13.1 | 49M     | 45.5        | 66.3       | 50.2       | 41.0        | 63.6       | 44.5       |
|              | Swin-S   | 10.2 | 70M     | 46.9        | 67.2       | 51.5       | 42.0        | 64.7       | 45.7       |
|              | SCL-Swin-T | 10.6 | 55M     | 46.7        | 67.1       | 51.6       | 41.9        | 64.6       | 45.5       |
|              | SCL-Swin-S | 8.0  | 77M     | 47.5        | 67.6       | 52.3       | 42.5        | 65.3       | 46.1       |

In cases, the proposed method effectively improved the performance of ViT backbones, especially in C-100 and TINY. For example, SCL improved PiT-XS by +5.25 in C-100, and improved Swin-T up to +5.7 in TINY. Also, this table shows that the proposed method effectively reduces the gap between ViT and CNN on small-size datasets. For example, the hybrid model SCL-CoAtNet-0 achieves the best performance on C-10 and C-100 beyond CNNs. And, SCL-CaiT-XXS24 overwhelmed CNNs for SVHN. In addition, the proposed method improves the performance of ViT more efficiently than methods of increasing model scales such as depth and hidden dimension. For example, SCL-CaiT-XXS24 has much better performance on all datasets, especially, +4.75 on the C-100, with a lower cost than CaiT-XXS24.

2) LEARNING ImageNet-1K FROM SCRATCH

Table 13 shows the performance when training ImageNet-1K from scratch. The proposed method improved the accuracy...
of DeiT-Ti by +3.2 and the accuracy of Swin-T by +1.1. Note that SCL-Swin-T outperforms RegNetY-8GF [21]. This result supports that the proposed method improves ViT backbones even on mid-size datasets.

3) TRANSFER LEARNING

Table 14 shows that the proposed method works well in transfer learning. In this experiment, transfer learning of a pre-trained network with ImageNet was performed to C-100 and TINY. There was performance improvement in both datasets. For example, the accuracy of DeiT-Ti on TINY increased by +3.59. Therefore, we argue that the proposed method has sufficient generalization ability.

C. COCO 2017

The training set of the COCO 2017 consists of 118K images. This capacity is very small compared to 1.2M of ImageNet, which is widely used for pretraining purposes. By learning the object detection and instance segmentation tasks of COCO from scratch, we tested whether the proposed method works in other vision tasks. Table 12 shows that the proposed method improves the Swin-T backbone by +1.2 AP\text{box}, +0.9 AP\text{mask} in object detection and instance segmentation, respectively. Also, the performance on COCO is improved by +0.6 AP\text{box} and +0.5 AP\text{mask}, even on a larger scale Swin-S backbone. Therefore, the proposed method is effective even in object detection and instance segmentation where locality information is important.

VI. CONCLUSION

To train ViT from scratch on a small-size dataset, this paper presents novel techniques to increase the locality inductive bias of ViT and to solve the smoothing distribution problem of attention scores. First, SPT embeds rich spatial information into visual tokens through specific transformations. Second, CPE explicitly uses the coordinate information of the visual tokens to position encoding. Finally, LSA induces ViT to attend locally through learnable parameters and diagonal masking within softmax. SPT, CPE, and LSA can achieve significant performance improvement independently, and they are applicable to any ViTs. The proposed method effectively generalizes the performance of ViT backbones by flattening the loss landscape. Therefore, this study shows that ViTs can learn small-size datasets from scratch and provides an opportunity for ViTs to develop further.

REFERENCES

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learn. Represent., 2021, pp. 1–22.
[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[3] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” 2019, arXiv:1905.11946.
[4] B. Neysabur, “Towards learning convolutions from scratch,” 2020, arXiv:2007.13657.
[5] 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.
[6] 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.
[7] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, “Rethinking spatial dimensions of vision transformers,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 11936–11945.
[8] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, “CvT: Introducing convolutions to vision transformers,” in Proc. IEEE Int. Conf. Comput. Vision. (ICCV), Oct. 2021, pp. 22–31.
[9] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jegou, “Going deeper with image transformers,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 32–42.
[10] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, “LocalViT: Bringing locality to vision transformers,” 2021, arXiv:2104.05707.
[11] W. Xu, Y. Xu, T. Chang, and Z. Tu, “Co-scale conv-attentional image transformers,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 9981–9990.
[12] Z. Wang, Z. Dai, B. Poczos, and J. Carbonell, “Characterizing and avoiding negative transferr,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 11285–11294.
[13] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, and V. Zambaldi, “Relational inductive biases, deep learning, and graph networks,” 2018, arXiv:1806.01261.
[14] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. 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.
[15] K. Wu, H. Peng, M. Chen, J. Fu, and H. Chao, “Rethinking and improving relative position encoding for vision transformer,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 10033–10041.
[16] J. Lin, C. Gan, and S. Han, “TSM: Temporal shift module for efficient video understanding.” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 7082–7092.
[17] R. El Jundi, T. Dargent, C. Petitjean, P.HONEINE, and F. Abdallah, “Investigating CoordConv for fully and weakly supervised medical image segmentation,” in Proc. 10th Int. Conf. Image Process. Theory, Tools Appl. (IPTA), Nov. 2020, pp. 1–5.
[18] R. El Jundi, C. Petitjean, P.HONEINE, and F. Abdallah, “CoordConv-UNET: Investigating CoordConv for organ segmentation,” IRBM, vol. 42, no. 6, pp. 415–423, Dec. 2021.
[19] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski, “An intriguing failing of convolutional neural networks and the coordconv solution.” in Proc. Adv. Neural Inf. Process. Syst., vol. 31, 2018, pp. 1–12.
[20] Z. Dai, H. Liu, Q. V. Le, and M. Tan, “CoAtNet: Marrying convolution and attention for all data sizes,” 2021, arXiv:2106.04803.
[21] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollar, “Designing
network design spaces,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10428–10436.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), vol. 25, Dec. 2012, pp. 1097–1105.

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2014, pp. 740–755.

[24] J. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, “Training data-efficient image transformers & distillation through attention,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 10347–10357.

[25] N. Park and S. Kim, “How do vision transformers work?” in Proc. Int. Conf. Learn. Represent., 2022, pp. 1–26. [Online]. Available: https://openreview.net/forum?id=D78Go4hVcxO

[26] J. Lei Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” 2016, arXiv:1607.06450.

[27] Y.-L. He, X.-L. Zhang, W. Ao, and J. Z. Huang, “Determining the optimal temperature parameter for Softmax function in reinforcement learning,” Appl. Soft Comput., vol. 70, pp. 80–85, Sep. 2018.

[28] 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. 1–11.

[29] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” Citeseer, Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

[30] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, “Reading digits in natural images with unsupervised feature learning,” in Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn., 2011, pp. 1–9.

[31] Y. Le and X. Yang, “Tiny ImageNet visual recognition challenge,” CS 231N, vol. 7, no. 7, p. 3, 2015.

[32] Y. Sun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, “CutMix: Regularization strategy to train strong classifiers with localizable features,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 6023–6032.

[33] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “Mixup: Beyond empirical risk minimization,” in Proc. Int. Conf. Learn. Represent., 2018, pp. 1–13. [Online]. Available: https://openreview.net/forum?id=1Ddp1-R6

[34] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, “AutoAugment: Learning augmentation strategies from data,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 113–123.

[35] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “RandAugment: Practical automated data augmentation with a reduced search space,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 3008–3017.

[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). Jun. 2016, pp. 2818–2826.

[37] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, “Deep networks with stochastic depth,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 646–661.

[38] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random erasing data augmentation,” in Proc. AAAI Conf. Artif. Intell., 2020, vol. 34, no. 7, pp. 13001–13008.

[39] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2015, arXiv:1412.6980.

[40] Y. Li, H. Zhang, and Y. Zhang, “Rethinking training from scratch for object detection,” 2021, arXiv:2106.03112.

[41] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 6105–6114.

[42] M. Tan and Q. Le, “EfficientNetV2: Smaller models and faster training,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 10096–10106.

[43] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2017, pp. 2961–2969.

[44] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1492–1500.

[45] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, “Visualizing the loss landscape of neural nets,” in Proc. Adv. Neural Inf. Process. Syst.vol. 31, 2018, pp. 1–11.

SEUNGHOON LEE received the B.S. degree and the master’s degree in electronic engineering from Inha University, Incheon, South Korea, in 2017, where he is currently pursuing the Ph.D. degree in electronic engineering. His research interests include computer vision and machine learning.

SEUNGHYUN LEE received the B.S. degree in electronic engineering from Inha University, Incheon, South Korea, in 2017, where he is currently pursuing the Ph.D. degree in electronic engineering. His research interests include computer vision and machine learning.

BYUNG CHEOL SONG (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1994, 1996, and 2001, respectively. From 2001 to 2008, he was a Senior Engineer at the Digital Media Research and Development Center, Samsung Electronics Company Ltd., Suwon, South Korea. In 2008, he joined the Department of Electronic Engineering, Inha University, Incheon, South Korea, where he is currently a Professor. His research interests include the general areas of image processing and computer vision.

* * *

---