Neural Processing Letters
https://doi.org/10.1007/s11063-022-11004-3

Learning Transferable Feature Representation with Swin Transformer for Object Recognition

Jian-Xin Ren[1] · Yu-Jie Xiong[1] · Xi-Jiong Xie[2] · Yu-Fan Dai[1]

Accepted: 8 August 2022

Abstract
Recent, substantial advancements in deep learning technologies have driven the flourishing of computer vision. However, the heavy dependence on the scale of training data limits deep learning applications because it is generally hard to obtain such a large number of data in many practical scenarios. And, deep learning seems to offer no significant advantage compared with traditional machine methods in a lack of sufficient training data. The proposed approach in this paper overcomes the problem of insufficient training data by taking Swin Transformer as the backbone for feature extraction and performing the fine-tuning strategies on the target dataset for learning transferable feature representation. Our experimental results demonstrate that the proposed method has a good performance for object recognition on small-scale datasets.

Keywords Transfer learning · Swin transformer · Object recognition

1 Introduction

Object recognition originated in the 1960s and is the basic task of computer vision. The recent rise of deep learning has allowed for the evolution of image recognition [1]. As shown in Fig. 1, object recognition usually contains three steps: (1) data preprocessing, (2) feature extraction, and (3) category prediction. In traditional object recognition, the processes of feature extraction algorithms binding classifier achieved a good performance in some simple recognition tasks. These algorithms, including SIFT [2], HOG, and SURF [3] extracted image features by artificial means. Therefore, its recognition accuracy depends heavily on the capacity of feature extraction. Conventional classification algorithms include KNN, SVM,

✉ Yu-Jie Xiong
xiong@sues.edu.cn

Jian-Xin Ren
751139811@qq.com

Xi-Jiong Xie
xiexijiong@nbu.edu.cn

[1] School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

[2] School of Information Science and Engineering, Ningbo University, Zhejiang 315211, China

Published online: 27 August 2022

Fig. 1 The process of object recognition

and Bayesian classifier. Among them, the approach of SVM as a classifier and the HOG as a feature extractor has gained great success in pedestrian detection [4]. However, these approaches are unsuitable for large-scale complex image recognition tasks owing to several drawbacks, including inefficiency, high cost, and lack of generalization ability. Concerning the deficiency of traditional recognition methods, Hinton et al. [5] proposed the concept of deep learning for the first time in 2006. After nearly ten years of development, the first deep convolution neural network (CNN) Alexnet [6] was proposed and won the championship in Beyond ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [7]. CNN has become central to computer vision and other AI applications due to its excellent performance. Since then, many notable CNN models like VGG [8], ResNet [9], and DenseNet [10] has been progressing to replace conventional recognition algorithm.

On the other hand, self-attention-based architectures, especially the most popular transformer [11] structure, have garnered much interest in natural language processing (NLP) today. The attention mechanism refers to the ability to selectively focus on specific relevant information and ignore irrelevant information when inputting images or sentences, which is inspired by the human visual system. The transformer is composed of an encoder-decoder structure, which uses a self-attention mechanism to catch the long-range dependencies in the data. Compared with a conventional method like LSTM [12], the transformer is easier to train and more efficient. Generally, the training process of transformer structures consists of two phases: pre-training on an extensive database and fine-tuning the model for downstream tasks. Due to the efficiency and high transferability of the transformer, it has become the popular choice in NLP.

With the tremendous success of the transformer in the field of NLP, researchers began trying to combine the attention mechanism with CNN or replace some components of the convolution network and get decent results. However, these methods are all variants of CNN in nature [13]. Recently, a great deal of works began to transfer transformer structures to visual tasks and achieved satisfactory results, such as ViT [14], Swin [15], DeiT [16], CoaT [17], CaiTCoaT [18], and so on. Unlike a previous study, these networks are pure Transformer structures employing image patches as input. Presently, as a new approach for image processing, the transformer has revolutionized the field of computer vision.

Inspired by the success of the transformer, we attempt to apply Swin Transformer for small-scale object recognition. The current study is rarely conducted for small-scale object recognition due to massive publicly available datasets like ImageNet [19] for model training. However, it is not easy to acquire such data on a large scale in practical [20]. As a result, research on object recognition suffered from severe overfitting due to insufficient training data.

With the deepening of the research, it has been found that transfer learning can effectively overcome the dilemma of insufficient data. The goal of transfer learning is to improve the performance of target learning tasks by transferring adequate knowledge from one or more source tasks. By sharing the low-level features across the source tasks and target task and fine-tuning the high-level features in the target task, transfer learning effectively solves various problems caused by insufficient samples. This paper first evaluates the transfer learning capabilities of several typical CNN and transformer architectures according to the number

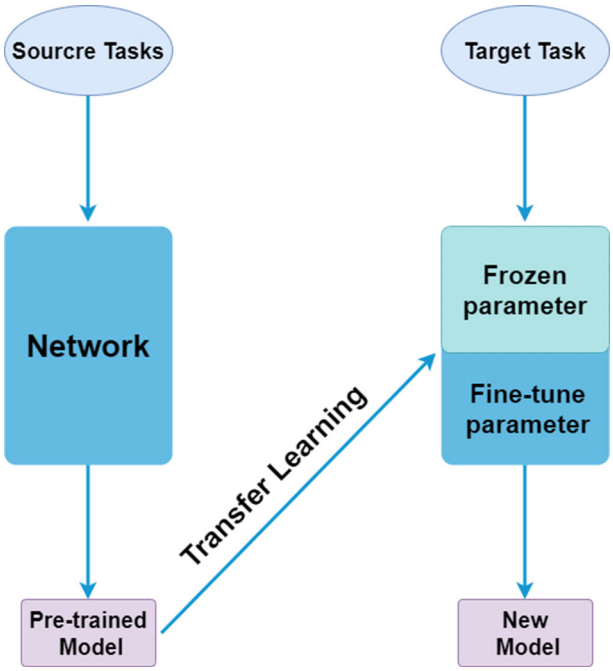

Fig. 2 The process of fine-tuning

of trainable parameters and the accuracy of CIFAR-10 and CIFAR-100. Then ImageNet pre-trained model are used to initialize the Swin Transformer architectures and are fine-tuned using the CIFAR-10 and CIFAR-100. In this way, a precise fine-tuning strategy is determined that is suitable for Swin Transformer. Moreover, the performance of the Swin Transformer on several small-scale datasets is evaluated by the fine-tuning strategy. The final experimental results show that our approach effectively helps improve the model's performance on a small-scale dataset.

2 Related Work

2.1 Transfer Learning

Transfer learning refers to transferring model parameters learned in source tasks to a new domain to improve the performance in a target task. Today, an increasing amount of work with the transfer learning has been performed. Fine-tuning is a highly efficient way of transfer learning. Figure 2 shows the detailed process of fine-tuning. The specific operation process of the fine-tuning is divided into two steps: (1) train the models into source tasks and get the pre-trained model; (2) transfer the parameters of the pre-trained model, freeze the low-level and fine-tune the high-level parameters. This paper significantly speeds up the model's convergence by freezing some parameters. Moreover, the experiment analyses the relationship between performance changes and the number of frozen parameters. The results demonstrated that our method dramatically reduces training time while preserving satisfactory accuracy compared to training the models from scratch.

2.2 Transformer

The transformer was proposed by Aswani et al. [11] in 2017 and soon became the mainstream framework in the field of NLP. Compared with the traditional LSTM, the parallelization design of the transformer significantly reduced the time needed to train the network and overcome computational performance and convergence issues caused by the problem of long-term dependencies. Because of its unusual properties and high transferability, many researchers have recently introduced a transformer into the task of computer vision. For example, Dosovitskiy et al. [14] first proposed a pure transformer with an encoder-decoder structure and showed a robust performance on many public datasets. ViT successfully resolved sharp increases in computational complexity with the resolution size. The primary approach is to split an image into fixed-size patches and send the segmented patch into a sequence of offset 2D patches through patch embedding. Then the output is sent to the transformer encoder, and the connection between each small patch is calculated through the multi-head self-attention (MSA) mechanism. Finally, the MLP layers classify the corresponding original image. Although ViT is quite expensive in computing, it is undoubtedly pushing the application of transformer in computer vision to a new height. Liu et al. [15] proposed the Swin Transformer. Through the operation of shifted window-based MSA, Swin Transformer greatly reduces the computational complexity. Moreover, a benefit from the hierarchical design, the size of the network's receptive field also increases layer by layer. Given the enormous success of these works, various transformer models based on ViT have emerged in an endless stream. In this paper, we compare the transfer learning performance of the Swin Transformer with several CNN series and transformer series networks, and the results indicated that the transfer learning performance of the Swin Transformer outperformed and outclassed the other models. the ImageNet-1K [19] is employed to perform pre-training, and CIFAR-10 and CIFAR-100 [21] datasets are used for fine-tuning.

2.3 Datasets

In this paper, the ImageNet [19] is employed to perform pre-training, and CIFAR-10 and CIFAR-100 [21] datasets are used for fine-tuning. ImageNet is one of the most popular datasets in the field of computer vision, with tens of millions of labelled images. Imagenet-1k is a sub-datasets of Imagenet, it constitutes 1000 categories, and each category contains 1200 high-resolution colour images. CIFAR-10 is a computer vision dataset consisting of 10 categories of 32×32 colour images. The dataset comprises 60,000 images, of which 50,000 images are for training and 10,000 images are for testing. The CIFAR-100 dataset has the same composition and image resolution as CIFAR-10, except that CIFAR-100 has 100 categories.

3 Method

3.1 Overall Architecture

Figure 3 presents the overall architecture of our method. In this paper, we adopted Swin-B [15] as the backbone for extraction. It follows a hierarchical modelling approach and comprises four phases denoted by Stage 1 $\sim$ Stage 4. The difference between stages is that Stage 1 is

composed of linear embedding and transformer blocks, and the others are composed of patch merging and transformer blocks.

The input image is first divided into non-overlapping patches with a patch size of 4×4 by a patch partition module to convert it into sequence embeddings, with a fixed size of $224 \times 224 \times 3$ (for RGB image). After this process, the channel dimension is changed to 48 ($4 \times 4 \times 3$), and the output is served as the input of Stage 1. In Stage 1, the patches are first extended with a linear embedding layer, and then several Swin Transformer blocks are employed for multi-scale feature extraction [22]. Stage 2 $\sim$ Stage 4 are similar to Stage 1 except replacing the linear embedding layer with a patch merging operation to generate hierarchical feature representations.

3.2 Patch Merging

Briefly, patch merging is a particular down-sampling method for keeping the information intact, which is analogous to the focus process in Yolo [23]. The main effect is forming a hierarchy with increased network depth by reducing the feature resolution to half of the original output. As shown in Fig. 4, a 4×4 with a single-channel image is first divided into four patches, with each patch of 2×2 adjacent spaced pixels (represented as the same colour). Then, the four patches are concatenated together, resulting in the feature resolution reduced by $2\times$ and the feature dimension increased by $4\times$. Finally, the feature dimension is controlled at $2\times$ the original dimension with a linear layer applied.

3.3 Swin Transformer Block

Swin Transformer block is composed of an even number of transformer block. Compared with the conventional transformer blocks, it replaces the standard multi-head self-attention module (MSA) with a window multi-head self-attention (W-MSA) and shifted-window multi-head self-attention (SW-MSA), while the other structures remain relatively constant. Figure 5 corresponds to two consecutive Swin Transformer blocks structure. Both are composed of

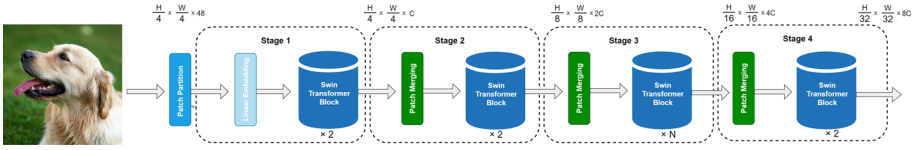

Fig. 3 The architecture of a Swin Transformer

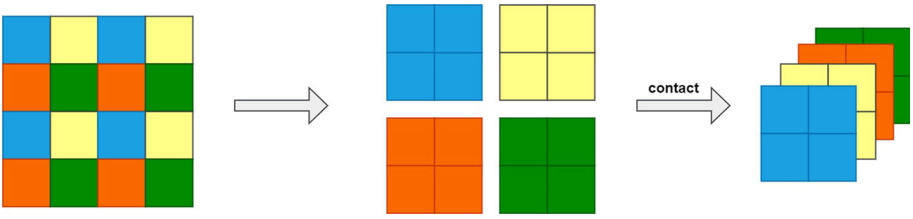

Fig. 4 The process of Patch Merging. (Color figure online)

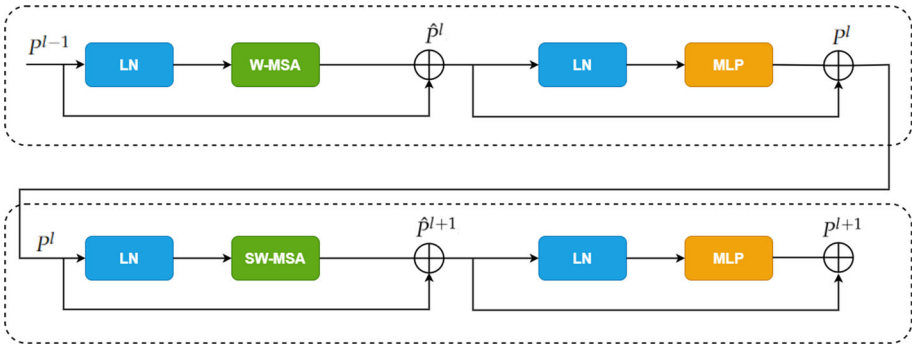

Fig. 5 Two consecutive Swin Transformer Blocks

Layer Norm (LN) layer, residual connection and 2-layer MLP, except W-MSA and SW-MSA are alternately used.

The specific process of the Swin Transformer block is as follows. In the first block, the Normalized P^{l-1} is put into the W-MSA module and is connected with P^{l-1} by a residual layer to produce $\hat{P}^l$. Similarly, $\hat{P}^l$ passes across LN and 2-layer MLP, and then a residual layer is applied to connect with itself for producing the P^l. In the second block, SW-MSA replaces W-MSA while all other operations are kept. Based on such a window partitioning mechanism, consecutive Swin Transformer blocks can be formulated as:

$$\hat{P}^l = \textit{W-MSA}(LN(P^{l-1})) + P^{l-1} \tag{1}$$

$$P^l = MLP(LN(\hat{P}^l)) + \hat{P}^l \tag{2}$$

$$\hat{P}^{l+1} = \textit{SW-MSA}(LN(P^l)) + P^l \tag{3}$$

$$P^{l+1} = MLP(LN(\hat{P}^{l+1})) + \hat{P}^{l+1} \tag{4}$$

where l represents block l, $\hat{P}^l$ represents the output features of the W-MSA module of block l, $\hat{P}^{l+1}$ represents the output features of SW-MSA module of block $l+1$, and P^l represents the module of block l. W-MSA represents multi-head self-attention using regular window partitioning configurations. LN represents Layer Normalization. MLP represents a multi-layer perceptron. SW-MSA represents multi-head self-attention using shifted window partitioning configurations.

3.4 W-MSA and SW-MSA

In NLP tasks, the relation between a single patch and the other patches is calculated with a standard MSA to conduct global self-attention. However, the global computation results in computational complexity which scales up exponentially with resolution size in complex object recognition tasks.

The W-MSA is designed to reduce the computational effort. Compared to conventional MSA, W-MSA divides the images ($h \times w$) into nonoverlapping local windows with a patch size of $M \times M$. Then, the relation of patches is calculated in local windows. The operation dramatically reduces the computational burden. The specific computational complexity of MSA/W-MSA is described as follows:

$$\Omega(MSA) = 4h \times w \times C^2 + 2(h \times w)^2 \times C \tag{5}$$

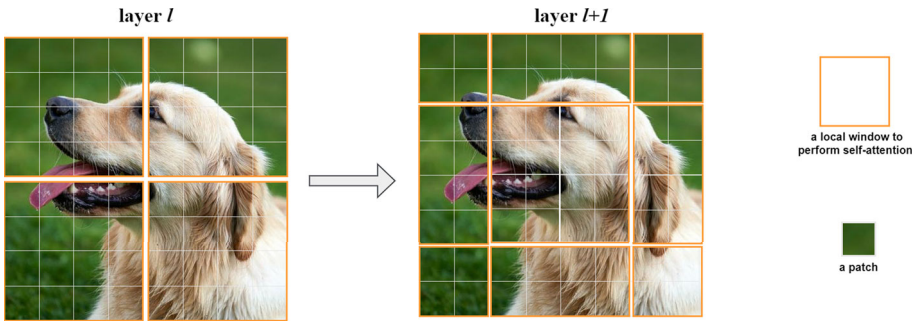

Fig. 6 A regular window partitioning strategy and a shifted window partitioning strategy

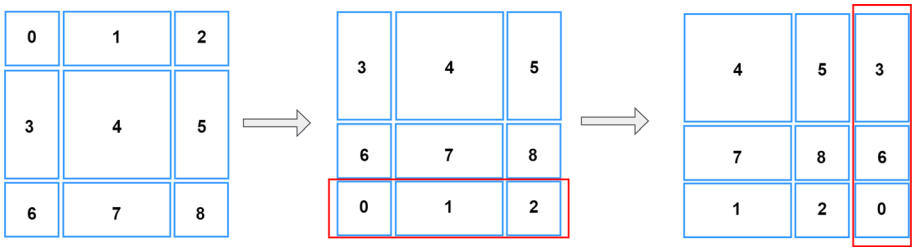

Fig. 7 Illustration of cyclic shift strategy

$$\Omega(W\text{-}MSA) = 4h \times w \times C^2 + 2M^2 \times h \times w \times C \tag{6}$$

The mechanism of partition windows dramatically reduces the computation cost but lacks connections across windows, limiting model capability. To extract interaction information while keeping the efficient computation of non-overlapping windows, SW-MSA is introduced. As shown in Fig. 6, *layerl* represents regular partition strategy (W-MSA), and *layerl*+1 represents shifted window partitioning strategy (SW-MSA). The 8×8 feature map is first divided into four local windows of size $4 \times 4 (M = 4)$ with W-MSA. Then, SW-MSA takes the place of local windows by traveling to the left upper corner by $(M/2, M/2)$ pixels from the regularly partitioned windows.

The cyclic shift is adopted in SW-MSA by combining the computation in four local windows like the regular partition strategy. As shown in Fig. 7, the local windows are indicated by 0–8. The (0, 1, 2) is first moved to the last row. Then, (3, 6, 0) of the new window is moved from the left to the right. After that, (5, 3), (2, 6, 8, 0), (1, 7), and (4) are considered a new local window.

Table 1 Performance comparison with other approaches on the CIFAT-10 and CIFAR-100 dataset

Model	Model type	Params (M)	CIFAR-10 (%)	CIFAR-100 (%)
VGG16bn [8]	CNN	132	94.00	73.17
WRN28-12 [24]	CNN	53	95.67	79.57
ResNeXt [9]	CNN	68	96.42	82.69
NesT-B [25]	Transformer	90	97.20	82.56
CCT-6/3x1B [26]	Transformer	3.17	95.29	77.31
Our method	Transformer	88	96.71	90.70

4 Results

In this section, we carry out several experiments that reflect the various aspects of the proposed method. At first, to investigate the effectiveness of our method, we compare it with some existing approaches on CIFAR-10 and CIFAR-100. Then, by freezing different layers, we study the influence of the pre-training model for the source task.

4.1 Experimental Settings

The experiments are performed on the framework of Pytorch (1. 4. 0), with NVIDIA-3070 for GPU acceleration, Intel (R) Core (TM) i7-9700k CPU and 16G memory. The operating system is Ubuntu 18.04 and the programming language is Python. We set the initial learning rate to 0.001, the learning rate decay to 0.95, and the minimum learning rate to 0.0000001. The AdamW optimizer is utilized during model training. Due to the GPU memory limitation, the batch size is set to 8.

4.2 Experiments

To evaluate the performance of our method, we compare it with existing methods, including three Transformer-based and three classic CNN-based models. Table 1 compares our approach with these architectures in terms of both accuracy and the number of network parameters. Typically, the higher accuracy indicates the more robust model's predictive capacity. Overly small parameters limit the model's capacity, while too many parameters lead to overfitting.

VGG16bn achieves the lowest accuracy with the highest complexity compared to the other models. It means that the performance of VGG16bn is significantly different from other models. The remaining two classic CNNs, WRN28-12 and ResNeXt64, perform better than VGG16bn. For ResNeXt64, the accuracy of 96.42% and 82.69% is achieved, 0.75% and 3.12% higher than WRN28-12, respectively. The above analysis implies that ResNeXt has the highest competitiveness among the three tested CNN models.

In Transformer-based models, NesT-B has superior accuracy compared to CCT-6/3x1B, with the accuracy of 97.20% and 82.56% on CIFAR-10 and CIFAR-100, respectively. It means that the NesT-B had a better prediction capability than CCT-6/3x1B. It is interesting to observe that although the accuracy of CCT-6/3x1B is inferior to CCT-6/3x1B, the number of parameters involved is far less than that in other models. Therefore, CCT-6/3x1B may have some advantages concerning some simple tasks. For our method, the accuracy on CIFAR-10 is

Table 2 The accuracy on CIFAR-10 for the original model and the model only using W-MSA

Module/test	CIFAR-10	CIFAR-100
W-MSA/SW-MSA	96.71	90.70
W-MSA × 2	90.13	82.89

Table 3 The accuracy of the cross-dataset test

Train/test	CIFAR-100-A	CIFAR-100-B	CIFAR-100-C	CIFAR-100-D
CIFAR-100-A	98.30	47.55	36.77	23.05
CIFAR-100-B	81.50	95.15	37.60	24.45
CIFAR-100-C	88.60	56.70	93.90	30.40
CIFAR-100-D	86.00	56.25	41.83	93.30

only 0.49% lower than the highest NesT-B, while the accuracy on CIFAR-100 is 8.01% higher than the second-highest NesT-B. The results clearly show that our method performs better than other tested models when meeting complex classification tasks. Although our approach shows no clear advantage in terms of the number of model parameters, it demonstrates the superior prediction accuracy and the stability of the model. Based on the results, our method performs better performance than others.

To test the role of SW-MSA in the model, we replace the SW-MSA with the W-MSA. As we can see from Table 2, the performance of the model only using W-MSA decreases sharply on CIFAR-10 and CIFAR-100 compared to the initial model. It is known from the above (see Method) descriptions that the main difference between W-MSA and SW-MSA is information interaction between the partitioned windows. Based on these reasons, we consider that the fixed window limits the predictive ability of models. This also indicates the necessity of SW-MSA.

To evaluate the association between the generalization capacity of the proposed model and the size of the source dataset, we carry out cross-dataset experiments on the CIFA-100. The CIFAR-100 is divided into four non-overlapping subsets: CIFAR-100-A (10), CIFAR-100-B (20), CIFAR-100-C (30), and CIFAR-100-D (40). CIFAR-100-A contains classes 1-10; CIFAR-100-B contains classes 11-30; CIFAR-100-C contains classes 31-60; CIFAR-100-D contains classes 61-100. Table 3 shows the accuracies of our proposed method with the cross-dataset settings, where rows and columns exhibit the datasets used for training and testing. Obviously, the best result is obtained when a model is trained and tested on the same dataset. The performance decreases rapidly when testing on the other datasets, implying a specific dataset carries distinctive characteristics. Moreover, the C model (the model is trained on the CIFAR-100-C) achieves higher accuracy than the D model, which is trained on more samples. While there are many possible reasons for this result, we considered it more probable that CIFAR-100-C contains greater species richness. We ascribe this performance boost to the fact that the CIFAR-100-C contains more similar object information to the target domain. For example, CIFAR-100-C contain bicycle, otter, and flatfish, while the target domain (CIFAR-10) contains motorcycle, beaver and aquarium fish. They are very close categories. The specific reasons need to perform studies on datasets containing more samples.

We also study the effect of correlations between the source and target dataset on model accuracy without fine-tuning. The four divided independent sub-dataset of CIFAR-100 are used as the source dataset and CIFAR-10 as the target dataset. As shown in Table 4, we carry

Table 4 The generalization accuracy on 10 classes of CIFAR-10 (without fine-tuning)

Target/source	CIFAR-100-A	CIFAR-100-B	CIFAR-100-C	CIFAR-100-D	CIFAR-100
Airplane	44.4	42.7	62.2	68.2	65.0
Automobile	70.4	85.0	91.5	71.7	82.6
Bird	49.3	53.4	57.5	59.1	65.4
Cat	34.5	44.3	56.7	49.6	58.4
Deer	44.4	49.8	62.5	46.7	54.4
Dog	60.0	49.0	61.5	51.9	62.8
Frog	77.2	83.6	77.5	81.3	85.0
Horse	72.0	75.9	60.4	68.2	73.5
Ship	49.4	72.0	67.1	61.6	76.4
Truck	78.6	78.3	58.7	82.7	69.4
Average accuracy	58.0	63.4	65.6	64.1	69.3

Table 5 The effect of the number of layers fine-tuned on model performance

Frozen layers	CIFAR-10 (%)	Time (min)	CIFAR-100 (%)	Time (min)
1, 2, 3, 4	76.21	174	34.4	232
1, 2, 3	85.98	228	67.84	276
1, 2	95.54	489	87.61	658
1	95.71	534	87.97	697
NONE	96.71	769	90.70	954
None pre-trained	96.43	–	90.39	–

'NONE' means no layer is frozen

out cross-dataset experiments and report the accuracy of the target dataset. On the automobile dataset, the accuracy of the C and D models is significantly higher than the other two models. The primary reason is that the source and target datasets have similar feature spaces, such as automobile-bus and automobile-pickup truck. Our analysis suggests that The ability to transfer learning is influenced by the correlations between the source and target datasets. In addition, we also find that although CIFAR-100 has the complete data of the other four sub-dataset, it does not achieve the highest accuracy in each class. It means that while source datasets have the same class, their amount of knowledge can be transferred differently.

This part investigates the effect of fine-tuning on the model's performance and studies the fine-tuning setting required to achieve the best performance. We first divided the model into five layers for fine-tuning according to the structure, with each stage being divided into a layer, and the remaining MLP is the last layer. Specifically, fine-tuning experiments are organized into two phases. We first initialize the model with the public ImageNet pre-trained and freeze all the layers of the pre-trained network. Then we adopt a fine-tuning strategy of gradually unfreezing one layer at a time, starting from layer 1.

Table 5 shows the effect of the number of layers fine-tuned on model performance. We first fine-tune the model by retraining the MLP layer while the rest of the layers are frozen. After that, we unfreeze the next layer and repeat the strategy until all layers are fine-tuned. As the fine-tuning goes deeper by layer, the accuracy of the model becomes higher. On CIFAR-10 and CIFAR-100, the prediction accuracy is improved from 76.21 to 96.71% and 34.40% to 90.70%, respectively. Compared to fine-tuning pre-trained models, the accuracy

Table 6 Comparison of performance over the model on different sizes of the training dataset

Pre-trained	1%-CIFAR-10	2%-CIFAR-10	5%-CIFAR-10	10%-CIFAR-10
CIFAR-100-A	83.52	87.00	91.56	91.43
CIFAR-100-B	84.26	87.20	91.90	91.53
CIFAR-100-C	85.42	88.30	92.33	92.19
CIFAR-100-D	83.66	87.67	92.15	92.39
CIFAR-100	85.61	87.88	92.52	92.55
ImageNet 1K	86.74	89.43	93.27	93.61
NONE	74.27	87.15	90.74	92.11

of training the models from scratch is decreased by 0.28% and 0.31%, respectively. However, we do not think that the best recognition accuracy is the best fine-tuning setting for small-scale object recognition. Because fine-tuning multiple layers would inevitably retrain the weights associated with the features, resulting in too many training parameters. For small-scale datasets, a huge number of training parameters usually lead to overfitting. Therefore, it is necessary to balance the degree of fine-tuning. As seen in Table 5, when layers 1 and 2 are frozen, the model achieves 95.54% and 87.61% of accuracy, which is close to the best accuracy and only takes less than 70% of training times compare to fine-tune all layers. It means using fewer parameters while maintaining good accuracy. Comprehensive consideration, freeze layers 1 and 2 is the more suitable choice for small-scale object recognition.

In this part, we follow the best fine-tuning strategy (freeze layers 1 and 2) to evaluate our method's performance on a small-scale dataset. The training datasets are 1%, 2%, 5%, 10% of CIFAR-10, selected at random, respectively, while keeping the complete training dataset of CIFAR-10. From the experimental results in Table 6, the best accuracies achieved from our fine-tuning strategy are all higher than without fine-tuning. And the smaller the number of training samples, the greater the accuracy gap. When only 1% of CIFAR-10 is used, the enormous accuracy gap reached 12.47%. The leading cause of the result is that the model trained on a small-scale dataset without fine-tuning, and learning too many insignificant features results in overfitting. Moreover, when 5% training data of CIFAR-10 is used, the highest accuracy of 92.52% is achieved. The accuracy is close to the original accuracy of 96.71% in Table 6. It is noteworthy that the model achieves better performance than the others when the ImageNet-1K is employed as a pre-training dataset. This suggests that we can select a larger pre-training dataset to gain better model performance for small-scale recognition in practical applications.

5 Conclusion

For a long time, the small-scale dataset has been a significant problem plaguing the field of object recognition. This paper takes the Swin Transformer as the base model and fine-tunes the model on the target dataset. In the case of only using 5% of the CIFAR-10, we achieve an accuracy of 92.52%, very close to the original accuracy. From the above numerous experiments, we conclude that our method effectively improves the accuracy and robustness of image recognition under the small-scale dataset. In our future work, we will apply the method in practice and investigate other complex object recognition tasks.

Funding Supported by: National Natural Science Foundation of China (62006150); Science and Technology Commission of Shanghai Municipality (21DZ2203100); Shanghai Young Science and Technology Talents Sailing Program (19YF1418400).

Declarations

Conflict of Interest The authors declare no conflicts of interest.

References

1. Kang Y, Chao G, Hu X, Tu Z, Chu D (2022) Deep learning for fine-grained image recognition: a comprehensive study. In: 2022 4th Asia pacific information technology conference, pp 31–39
2. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
3. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (SURF). Comput Vis Image Underst 110(3):346–359
4. Deng Z, Zhou L (2018) Detection and recognition of traffic planar objects using colorized laser scan and perspective distortion rectification. IEEE Trans Intell Transp Syst 19(5):1485–1495
5. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313:504–507
6. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25
7. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
8. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations, pp 1–14
9. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition, pp 770–778
10. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: IEEE conference on computer vision and pattern recognition, pp 4700–4708
11. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Conf Workshop Neural Inf Process Syst 30(11):6000–6010
12. Shi X, Chen Z, Wang H, Yeung D-Y, Wong W-K, Woo Wc (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. Conf Workshop Neural Inf Process Syst 1(9):802–810
13. Zhou J, Sun J, Zhang M, Ma Y (2020) Dependable scheduling for real-time workflows on cyber-physical cloud systems. IEEE Trans Ind Inf 17(11):7820–7829
14. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S et al (2021) An image is worth 16x16 words: transformers for image recognition at scale. In: International conference on learning representations
15. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: hierarchical vision transformer using shifted windows. In: IEEE international conference on computer vision, pp 10012–10022
16. Touvron H, Cord M, Douze M, Massa F, Sablayrolles A, Jégou H (2021) Training data-efficient image transformers & distillation through attention. In: International conference on machine learning, pp 10347–10357
17. Xu W, Xu Y, Chang T, Tu Z (2021) Co-scale conv-attentional image transformers. In: IEEE conference on computer vision and pattern recognition, pp 9981–9990
18. Touvron H, Cord M, Sablayrolles A, Synnaeve G, Jégou H (2021) Going deeper with image transformers. In: IEEE conference on computer vision and pattern recognition, pp 32–42
19. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition, pp 248–255
20. Zhou J, Cao K, Zhou X, Chen M, Wei T, Hu S (2021) Throughput-conscious energy allocation and reliability-aware task assignment for renewable powered in-situ server systems. IEEE Trans Comput Aided Des Integr Circuits Syst 41(3):516–529

21. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. Handb Syst Autoimmune Dis 1(4)
22. Chao G, Luo Y, Ding W (2019) Recent advances in supervised dimension reduction: a survey. Mach Learn Knowl Extr 1(1):341–358
23. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: IEEE conference on computer vision and pattern recognition, pp 779–788
24. Zagoruyko S, Komodakis N (2016) Wide residual networks. In: British machine vision conference, pp 1–13
25. Zhang Z, Zhang H, Zhao L, Chen T, Arik S, Pfister T (2022) Nested hierarchical transformer: towards accurate, data-efficient and interpretable visual understanding. arXiv preprint arXiv:2105.12723
26. Hassani A, Walton S, Shah N, Abuduweili A, Li J, Shi H (2021) Escaping the big data paradigm with compact transformers. arXiv preprint arXiv:2104.05704