International Journal for Housing Science and Its Applications
Publish August 4, 2025. Volume 46, Issue 3 Pages 1304-1313

https://doi.org/10.70517/ijhsa463101

Research on Efficient Migration Learning Algorithms Based on Large Models

Kezhi Zhen[1,*], Zheng Qi[1], Xu Li[1], Xin Yin[1], Jifu Wang[1] and Jing Chen[1]
[1] China Tobacco Guizhou Industrial Co., LTD. Guiyang, Guizhou, 550001, China
Corresponding authors: (e-mail: 17708541846@163.com).

Abstract Natural language processing and computer vision technologies have greatly contributed to the development of large language models. In this paper, we focus on the introduction of Adaptive Module method in the pre-trained model to realize the efficient migration of the model and improve the performance of the model. In the applied research in the field of natural language processing, the Adapters module is introduced into the ALBERT-BiLSTM-CRF model to tune the overall model. The adapter mechanism is utilized to improve the representation ability in the visual Transformer model. The results show that, through the comparative analysis of a large number of transfer learning methods, it can be seen that Adapters achieved a high average performance, with a tuning parameter number of only 0.23%. Therefore, Adapters is selected for the case study.The average number of parameters in the ALBERT-BiLSTM-CRF model with the addition of Adapters module is only 30M with an F1 value of 94.41%.The Adapters adapter component mechanism is capable of adapting to a wide range of downstream tasks and obtaining a better image representation.

Index Terms adapters module, computer vision, natural language processing, adapter mechanism, transformer model

I. Introduction

In the field of artificial intelligence, large models are those deep learning models that are trained using large amounts of data and arithmetic power and have large parameter scales [1], [2]. These models usually have stronger generalization and learning abilities, and can show superior performance on multiple tasks [3], [4]. Large models have become one of the most important research directions in artificial intelligence in recent years [5].

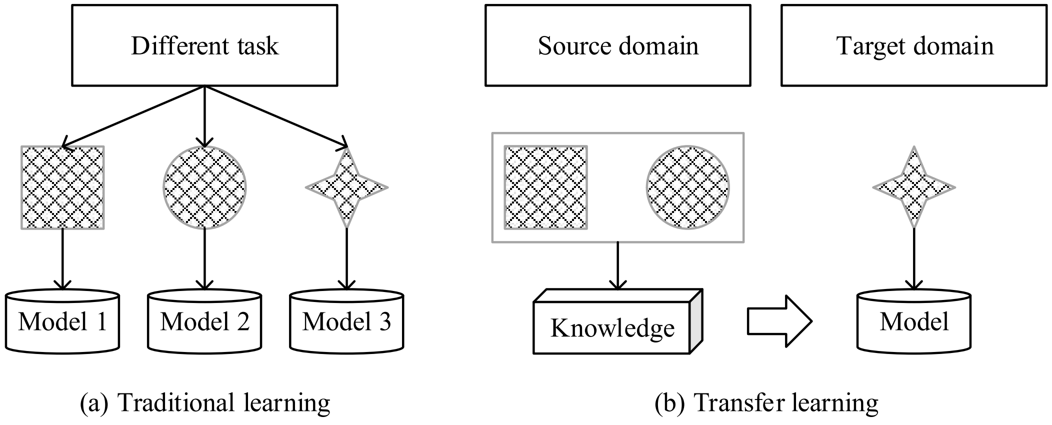
And transfer learning is the technique of taking knowledge learned from one task or domain and applying it to another task or domain [6], [7]. It improves the performance of a new task by reusing and transferring previously learned relevant knowledge [8]. The goal of transfer learning is to improve the performance of a target task by utilizing the experience of the previous task [9]. Its basic assumption is that there are some identical features between different tasks which can be used to extract knowledge [10], [11]. Migration learning can reduce the data requirements for new tasks, accelerate the learning process, and improve the generalization ability of the model [12]. With the rapid development of artificial intelligence technology, transfer learning is widely used as an effective machine learning method [13]. In real-world scenarios, there are often differences in data distribution between different tasks, which requires us to utilize existing knowledge and experience to solve new tasks through transfer learning [14], [15]. However, how to better optimize the transfer learning model in practice is still a challenge, and it is difficult to meet the application requirements in resource-constrained scenarios. In this context, efficient transfer learning based on large models aims to achieve fast adaptation and deployment of large models with minimal resource overhead [16]-[18].

To address the limitations of large model training for applications in different domains, there is a need to utilize migration learning of existing knowledge and models to migrate the source domain information to the target domain and reduce the training cost of the model. From 2 major aspects of natural language understanding and computer vision, and using GLUE as one of the evaluation benchmarks, we comprehensively compare the similarities and differences of different efficient migration methods through experiments. We point out the idea and method to realize the efficient migration by adding the Adapters module into the pre-trained model. In this paper, we propose the AABC model and the Adapters adapter component mechanism, and verify its effectiveness for natural language processing and visual task migration learning through comparative experiments.

II. A brief description of transfer learning

II. A. Transfer learning

Migration learning is an important research subfield of machine learning, and the two are very closely related. Machine learning through the search for the optimal function f, through f constraints on the model so as to achieve the minimum loss in the training set, the model not only has a strong ability to fit the training set, but also should have enough prediction ability for unknown data, to the structural risk minimization (SRM) as a criterion, in the fitting of the training set based on the complexity of the model also has a relatively simple, and this relative simplicity of complexity to give the model powerful generalization capabilities [19]. Migration learning improves the model's generalization ability by overcoming the main problem that constrains the model's generalization ability, namely the different data distributions between the training set and the test set. Migration learning using deep learning techniques enables the network to effectively utilize the semantic features acquired in the source domain. Migration learning is an effective machine learning method, which accelerates the learning of the target task and improves the efficiency and accuracy of machine learning by migrating the knowledge from the source domain to the target domain, the migration learning method is shown in Figure 1. In practical applications, transfer learning has been widely used in computer vision, natural language processing and other fields, and achieved very good results.



(a) Traditional learning (b) Transfer learning

Figure 1: Traditional learning and transfer learning

Domain Adaptation (DA) aims to regularize the difference in edge probability distributions between the source and target domains, which is mathematically described as follows: $X \times Y$ denotes the joint feature space and the corresponding label space, respectively [20]. The source domain S and the target domain T are defined on $X \times Y$ and have different probability distributions P_s and P_t. Suppose there are n_s labeled samples in the source domain, i.e:

$$\mathrm{D}_S = \{(x_i^S, y_i^S)\}_{i=1}^{n_S} \tag{1}$$

where D_S denotes the samples in the source domain and there are also n_t samples (with or without labels) in the target domain, i.e:

$$D_T = \{(x_j^T)\}_{j=1}^{n_s} \tag{2}$$

D_T denotes the samples in the target domain. Then the goal of DA is to transfer the knowledge learned from S to T in order to perform a specific task on T.

The formal description of domain adaptation is as follows

There is a source domain data distribution $p_s(x,y) \in P_S$, and a target domain data distribution $p_t(x,y) \in P_T(P_S \neq P_T)$. And now there is a dataset $\mathrm{D}_s = \{x_i, y_i\}_{i=1}^{N_s}$ obtained independently and identically distributed from $p_s(x,y)$, $\mathrm{D}_t = \{x_i, y_i\}_{i=1}^{N_t}$ is obtained from $p_t(x,y)$ independently identically distributed. The aim of the domain adaptive problem is to use the source domain data to learn a predictive function on the target domain, as shown in Equation (3):

$$f^* = \arg\min_f \mathrm{E}_{(x,y)\in \mathrm{D}_t} \in (f(x), y) \tag{3}$$

Unsupervised domain adaptation is the problem of domain adaptation when the target domain data is completely unlabeled $D_T = \{x_i, ?\}_{i=1}^{NT}$ a case. And it is often the case that we do not have just one source domain, but multiple source domains, i.e., there exists a set of source domains $D = \{D_s^j\}_{j=1}^{M}$ consisting of M source domains, where each of the source domains $D_s^j = \{x_i, y_i\}_{i=1}^{N_s}$ follows a different probability distribution P_S^M . At this point, M source domains can be used to help the target domain task to learn, degrading to a single source scenario when $M = 1$.

In the field of natural language processing, pre-trained language model (PLM)-based transfer learning has become the dominant paradigm and has demonstrated excellent performance in several tasks. In the field of computer vision, excellent performance has been achieved in a variety of tasks. Lightweight adaptation modules are inserted into pre-trained models, the weights of the pre-trained models are frozen, and then these modules are fine-tuned end-to-end to adapt to downstream tasks. These approaches have demonstrated the effectiveness of adapter modules in vision tasks.

II. B.Adapter Module Architecture

With the improvement of computer hardware performance, the number of pre-trained model parameters is increasing, and it becomes expensive and time-consuming to perform full-model fine-tuning when training downstream tasks, which is alleviated by the emergence of Adapter, which inserts parameters for downstream tasks into each layer of the pre-trained model, freezes the model body during fine-tuning, and trains only the task-specific parameters to reduce the computational power overhead during training [21]. The main architecture of Adapter is shown in Figure 2.

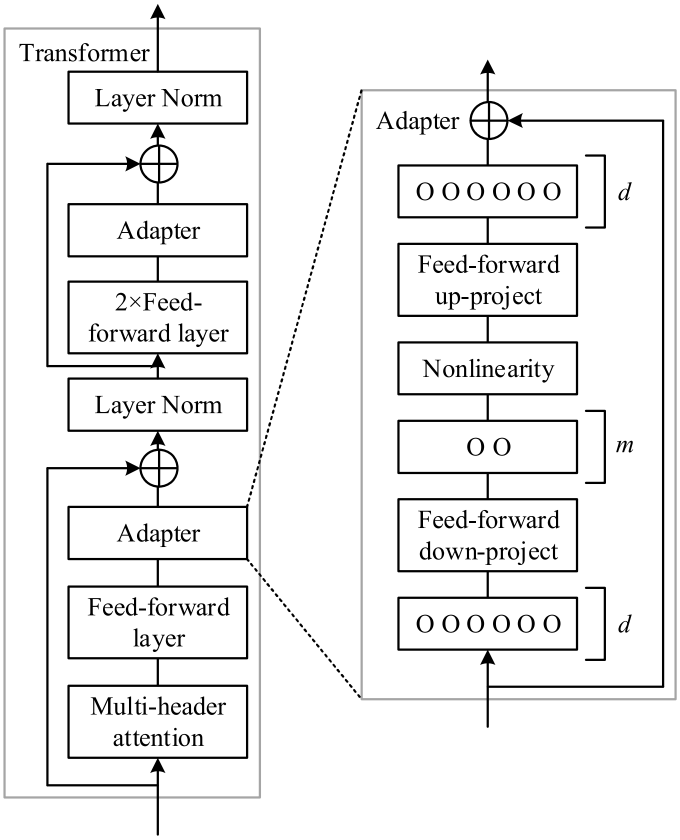


Figure 2: Adapter-based fine-tuning method

III. Efficient Migration Methods on NLP and Visual Models

With the rise of large model technology, some scholars have begun to focus on methods to reduce the computational overhead of deep learning, such as model compression, model distillation, etc. However, the existing migration learning review fails to fully consider the problem of efficient migration in resource-constrained scenarios.

Natural language processing is a very important application area of deep learning, and the specific applications include language understanding, translation, dialog, Q&A, etc.-considering that the current parametric macroscale models are all natural language macroscale models.

For the Transformer-based language models, the mainstream methods are based on injecting adaptive parameters and inducing adaptive modules to achieve efficient migration, and the more representative and efficient model migration methods in the field of natural language processing are summarized as shown in Table 1.

Table 1: Some of the efficient migration of the field of natural language processing

Method	Parameter ratio(%)	Memory overhead
Compactor	0.06	-43
Prefix-tuning	0.2	-
Pormpt Tuning	<0.02	-
LoRA	0.25	-68
QLoRA	-	-
BitFit	0.1	NA
AdapterFusion	-	-
IA	002	-
LST	1.85	-38
AttEMPT	0.06	-
Adapters	3.8	-
Diff Pruning	1	-
HyperFormer	0.32	-26
T5	-	-

In this paper, we conduct a comprehensive comparison of experimental results of some of the representative methods presented in the previous paper on GLUE (containing 8 sub-datasets), one of the most widely used evaluation benchmarks with the most comprehensive coverage of tasks, and the comparison results are shown in Table 2. It can be found that there is a wide variety of backbone networks for natural language processing tasks, and the performance of methods fine-tuned based on different backbone networks varies greatly. The number of parameters tuned by different methods also varies, and in particular, methods involving the addition of an adaptive module need to tune more parameters on average. BERT-large based Adapters achieved the best average performance among the compared methods with a lower number of tuned parameters (0.23%). Its excellent overall performance also lays the foundation for a large number of subsequent related research efforts.

Table 2: Experimental results in the glue benchmark data set

Method	Parameter ratio(%)	CoLA	SST-2	MRPC	QQP	STS-B	MNLI	QNLI	RTE
Adapters	0.23	63.5	96.2	90.8	90.99	92.2	88.6	94.6	87.3
Diff pruning	1.03	62.3	95.2	92.2	87.2	90.1	87.4	93.2	72.4
LST	1.83	70.5	95.2	91.7	89.2	91.4	86.9	94.4	72.2
ATTEMPT	0.05	66.7	94.7	88.6	90.3	90.9	94.5	94.5	80.7
Compacter	0.06	63.9	94	90.4	90.5	90.6	87.2	93.0	78.2
HyperFormer	0.3	63.9	95.6	90.8	90.5	90.2	86.8	93.4	75.8
BitFit	0.1	62.9	94.3	92.8	85.2	94.7	85.7	92.2	78.5
LoRA	0.65	64.7	95.8	90.9	91.2	92.2	88.6	93.5	87.4

The fields of computer vision and natural language processing are both everywhere different and closely related. Also as an application area of deep learning, the basic goals and definitions of implementing model migration in them are the same.

Computer vision has been developing at a high speed in recent years in application areas such as facial recognition, intelligent driving, image generation, and action segmentation. Taking smart driving as an example, the arithmetic power of in-vehicle platforms is often very limited, so efficient migration algorithms are needed to recognize road conditions and process large models with low overhead to the vehicle side. Visual information processing is also an indispensable basic capability in many wearable devices, and only efficient migration algorithms have the ability to migrate large visual models under the arithmetic power supported by such devices.

In order to demonstrate more intuitively the performance of comparing different methods, this paper collects the experimental results of representative methods in the field of computer vision on VTAB-lk, the most widely used vision benchmark dataset. The average experimental results of different methods on three types of tasks of VTAB-lk: natural category, specialized domain and structured are shown. The experimental results are shown in Table 3, where VIT-B/16 in the backbone network is obtained by pre-training on ImageNet-21k and VIT-B/16 is obtained by pre-training on □ImageNet-2lk . Among all the compared methods, the method based on adaptive modules (Adapters) shows a higher average performance, but correspondingly more parameters need to be tuned, e.g., Conv-Adapter, with a parameter share of 5.8%. In practice, performance and parameter efficiency need to be weighed according to specific needs.

Table 3: Experimental results on the vtab-lk data set

Method	Parameter ratio(%)	Natural	Specialized	Scructured
VPT	5.2	80.3	83.2	56
Conv-Adapter	5.8	81.2	85.9	63.1
SPT	0.5	83.6	86.2	61.5
AdapterFormer	0.5	80.7	85.5	59.7
Convpass	0.5	81.7	85.4	63.2
VQT	3.6	73.3	84.7	50.8
Head2Toe	1.2	69.4	83.2	47.1
EXPRES	<1.2	80.8	84.4	56.8

IV. Adapter-ALBERT-BiLSTM-CRF(AABC) model

IV. A. CRF model

The most commonly used in natural language processing today is the linear chain conditional random field. It is used for lexical annotation of serialized data and slicing of data, and it is used to compute the conditional probability distribution of a labeled sequence given a sequence of observations. A linear chain CRF is defined as follows: suppose there are two linear chains of sequences of random variables $x=(x_1,x_2,\cdots,x_n)$ and $y=(y_1,y_2,\cdots,y_n)$, and if x and y satisfy the Markov property $p(y|x,y_1,y_2,\cdots,y_n)=p(y|x_i,y_{i-1},y_{i+1})$, i.e., then $p(y|x)$ is said to be a conditional random field of a linear chain. Where x is the observation sequence of the input person and y is the labeling sequence corresponding to it, the parameterized representation of the conditional random field takes the following form:

$$p(y|x)=\frac{1}{z(x)}\exp\left(\sum_{i,k}\lambda_k t_k(y_{i-1},y_i,x,i)+\sum_{i,l}\mu_i s_i(y_i,x,i)\right) \tag{4}$$

where t_k and s_l are the eigenfunctions with corresponding weights λ_k, μ_l, and the normalization factor $z(x)$ is expressed as follows:

$$z(x)=\sum_{y}\exp\left(\sum_{i,k}\lambda_k t_k(\lambda_{i-1},y_i,x,i)+\sum_{i,l}\mu_i s_l(y_i,x,i)\right) \tag{5}$$

where t_k denotes the transfer feature, which depends on the current position and the previous position, and s_l denotes the state feature, which only depends on the current position.

The CRF model is widely used as a label decoder in deep learning-based named entity tasks, and the CRF is able to improve the accuracy of NER by effectively modeling the prediction of constraint relationships between labels. The weights f_1 in the figure correspond to the vector features μ_i obtained for each word. The following formula is used to calculate the probability obtained for the input sentence sequence x and the output sequence is y:

$$s(y|x)=\sum_{j=1}^{m}\sum_{i=1}^{n}\mu_i f_j(x,i,y_i,y_{i-1}) \tag{6}$$

Where i represents the position of the word in the sentence, y_i is the label of the current word, y_{i-1} is the label of the previous word, m corresponds to the number of features, and n represents the length of the input sentence, and then the obtained scores are normalized to convert the results into probability values, and the output with the largest probability value is used as the final label of the sequence.

IV. B. ALBERT-BiLSTM-CRF (ABC) training model

This model consists of word vector layer ALBERT, BiLSTM layer and CRF layer from top to bottom. Its input is the serialized text, and the corresponding annotated sequence of the output is obtained at the CRF layer, and the output sequence is annotated with the BMEO used [22]. Where B denotes the beginning of an entity, M denotes the middle of an entity, E denotes the end of an entity and 0 denotes a non-entity. In the process of model implementation, each character of the input is converted into vector form, which is used as the input of BiLSTM to extract the contextual features, and the output feature vector is used as the input of the CRF layer, which is normalized to the input, and finally outputs the annotation sequence.

The ALBERT layer is used as the first layer of the model, which employs matrix decomposition to reduce the number of parameters and introduces a low-dimensional vector space E with the decomposition formula:

$$O(V \times H) = O(V \times E + E \times H) \tag{7}$$

where V represents the vocabulary list vector and H represents the hidden layer vector size. Meanwhile, the ALBERT model proposes the use of SOP pre-training, which focuses on inter-sentence coherence, to improve the performance of the downstream multi-sentence coding task.

The last CRF layer is used to constrain the order of words through the Viterbi algorithm to get the highest score sequence annotation. It is used to ensure that the beginning of the entity must be B rather than M or E. For example, the corresponding labeling of the model output “Gas Explosion” should be “B-Class, M-Class, M-Class, E-Class”, if there is no constraint of CRF layer, the corresponding labeling information may appear as “M-Class, M-Class, E-Class”. If there is no CRF layer constraints, the corresponding output labeling information may appear “M-Class, B-Class, M-Class, E-Class” and other incorrect labeling.

IV. C. Model Evaluation Criteria

For training on the AABC model, the criteria used for this model are precision, recall, and $F1$-value, as specified in Eq:

$$p = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FP} \tag{9}$$

$$F1 = \frac{2 * P * R}{P + R} \tag{10}$$

Where TP, FP, TN, and FN, these four metrics form the confusion matrix of classification results, which denote the prediction of the positive category as a positive category, the prediction of the inverse category as a positive category, the prediction of the inverse category as an inverse category, and the prediction of the positive category as an inverse category, respectively.

V. Applied research on transfer learning in the field of natural language processing

The experiments mainly focus on the BERT-BC and ABC models, and also compare the effects of the two types of models on the three tasks after adding the Adapters module. The results of the experiments are based on the F1 value, the accuracy and the final number of parameters as a reference.

BERT-BC is used as the base control experiment, because there are a large number of Attention computational mechanisms in the model and the parameters are not shared among the layers, resulting in a huge number of parameters in the model (BERT-base and ALBERT-base are used as the participating experimental models in this paper). Therefore, in terms of model size, the ABC model is lighter compared to BERT-BC. The results of the four models on the five NER public datasets are shown in Table 4. On the complex datasets OntoNotesv5 and WeiboNER, BERT-BC works better because the datasets are relatively complex, and the task difficulty is higher compared to other datasets, and datasets of this difficulty are more suitable for BERT-BC with higher model complexity. While on the ResumeNER and On the MSRA dataset, the reason why AABC is better than ABC: the addition of the Adapters module increases the number of layers and complexity of the model in disguise, and adds the process of semantic learning in the process of model training. In terms of the average number of parameters, the average number of parameters of AABC is 30M, which is the lowest among the four models, only 20% of the 150M of BERT-BC. With the addition of the Adapters module, the number of parameters of BERT-BC has been halved, but the performance has dropped a little bit, which indicates that for the BERT model, a large number of parameters need to be adjusted to maintain the performance of the model.

Table 4: Performance on the ner task

Dataset	BERT-BC	Adapters-BERT-BC	ABC	AABC
CoNLL2003	92.12%	92.13%	92.73%	92.34%
OntoNotersv5	90.09%	88.34%	90.21%	86.32%
WeiboNER	63.21%	63.23%	63.56%	61.26%
ResumeNER	97.42%	96.93%	96.56%	96.78%
MSRA	93.03%	93.46%	93.40%	94.45%
ParamsAve	150M	80M	50M	30M

The average number of parameters of the models under the five datasets is shown in Table 5.The NER task evaluation index is the F1 value. In terms of F1 value, the ABC model has the highest F1 value on the CoNLL2003 dataset, and the BERT-BC has the highest F1 value on the OntoNotesv5 and WeiboNER datasets. On the ResumeNER and MSRA datasets, AABC has the highest F1 value of 94.41%.

Table 5: Special instructions in msra data set

MSRA	P(Precision)	R(Recall)	F1
BC	90.93%	90.23%	91.83%
BERT-BC	93.35%	92.35%	93.41%
ABC	94.09%	93.21%	92.41%
AABC	96.22%	93.76%	94.41%

VI. Adapter method for computer vision

VI. A. Transformer encoder

ViT consists of L identical encoders connected in series with an encoder structure, where layer 1 is Multihead Self Attention (MHSA) and layer 2 is a feed forward neural network. After the data is output from each layer, it is fused with the input data using residual linkage, normalized and then input to the next layer The output dimension of each layer is designed to be d -dimensional, and the classification flag bits z_L^0 after L encoders are input to the classification head composed of multilayer perceptron machine (MLP) to predict the image category y . The computational procedure for the l th encoder is shown in Equation (11) and Equation (12):

$$z'_l = MHSA(LN(z_{l-1})) + z_{l-1}, \quad l = 1, \cdots, L \tag{11}$$

$$z_l = MLP(LN(z'_l)) + z'_l, \quad l = 1, \cdots, L \tag{12}$$

The results of the category predictions are shown in equation (13):

$$y = LN(z_L^0) \tag{13}$$

VI. B. Adapter

The Adapter method for parameter-efficient fine-tuning is proposed in 2019 to achieve similar performance to full fine-tuning with high training efficiency by using adapter modules to add a small number of new parameters to the model to be trained in a downstream task. Two Adapters are added to each Transformer, each with two main sub-layers, the descending feedforward layer and the ascending feedforward layer.

Compressor is a method for fine-tuning large language models, and this proposed method in 2021 provides a better tradeoff between task performance and the number of trainable parameters compared to previous work by leveraging adapters, low-rank optimization, and parameterized hypercomplex multiplication layers.

Adapter fusion is a new two-stage learning algorithm designed to address the problems of catastrophic forgetting, interference between tasks, and training instability by combining knowledge from multiple tasks. The first stage is the knowledge extraction phase, where AdapterFusion learns task-specific parameters of adapters = these adapters encapsulate information relevant to a specific task, and is accomplished by adding a small number of □ parameters to a pre-trained language model without changing the underlying weights of the language model.

VII. Applied research on transfer learning in computer vision

VII. A. Comparative experiments

The grouping statistics are shown in Fig. 3, where it can be observed that Adapter achieves the best performance on all three image categorization datasets, further demonstrating the effectiveness of the adapter component mechanism proposed in this paper for transfer learning of visual tasks. Across different task groups, Adapter significantly outperforms the other methods in the natural and fine-grained groups, while its performance is comparable to that of Convpass in the specialized group. This paper suggests that this may be due to the fact that Adapter is able to effectively focus on localized information and thus performs better in scenarios such as low-resolution images. Thus, this study also demonstrates the key role of adapter structure in the performance of transfer learning, while emphasizing the specificity of the adapter component, i.e., its sensitivity to different datasets. Finally, the experimental results also clearly demonstrate that pre-trained models using a self-supervised approach show better migration performance in downstream tasks compared to pre-trained models under supervised training.

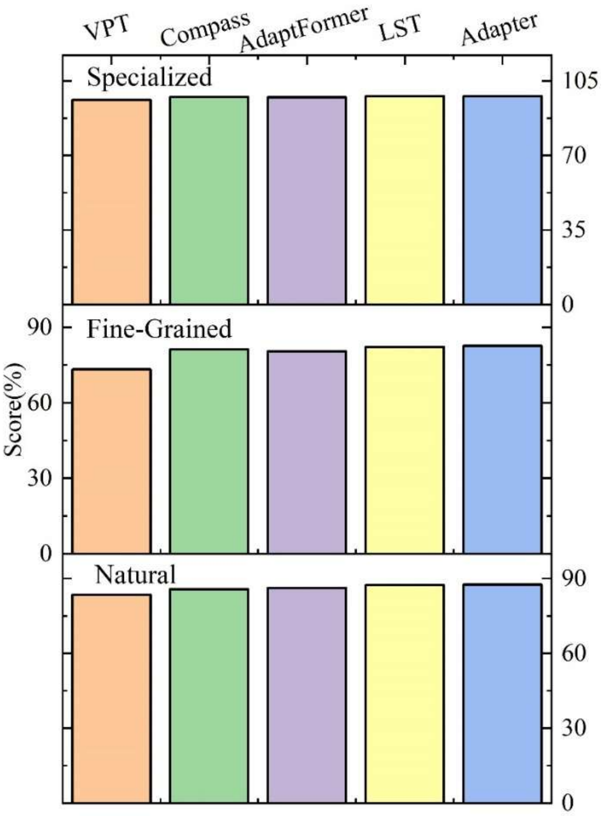


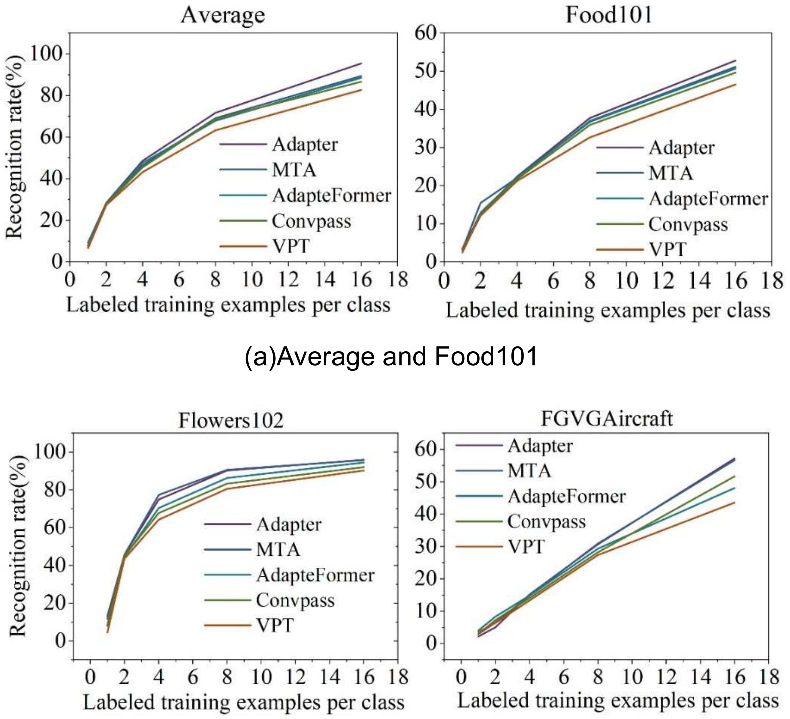
Figure 3: Average performance in different groups

VII. B. Small Sample Learning Experiments

For the small sample learning task, three fine-grained image recognition datasets, Oxford Flowers102, Food101 and FGVCAircraft, are selected to evaluate the performance of the proposed method in this paper. In this subsection, a series of experiments are conducted with different settings of 1, 2, 4, 8 and 16 samples and repeated using different random number seeds and finally the average of the results of the three experiments is taken for the different methods.

The results are shown in Figure 4, where the average performance of Adapter significantly outperforms the other baseline methods in the five settings. In particular, Adapter performs best on the FGVCAircraft dataset. And on the Flowers102 and Food101 datasets, all methods show similar levels of performance. With 16 samples, Adapter slightly outperforms the Convpass and AdaptFormer methods, although the latter two also perform quite well in that setting. These experimental results clearly show that Adapter can effectively enhance the learning ability of ViT in data-constrained scenarios. From the perspective of a small sample learning task, the Adapter mechanism is able to adapt to a variety of downstream tasks through its structure, rather than relying on larger scale parameters or more data for better image characterization.

The experimental results also highlight the importance of network structure for specific downstream tasks. The experimental results of the Adapter mechanism show its adaptability and generalization ability in scenarios with restricted data samples, further highlighting its value as an effective transfer learning strategy.



(a)Average and Food101

(b)Flowers102 and FGVGAircraft

Figure 4: Sample study results

VII. C. Visual Analytics

In this section, in order to be able to more intuitively see the performance enhancement effect achieved by our proposed method, we adopt the t-distribution-stochastic nearest-neighbor embedding (t-SNE) approach to visualize the features of the input image. Specifically, we visualize the features directly after global average pooling of the feature maps of the input images and the packet-level features obtained by our proposed multi-example learning framework, respectively. The experimental results are shown in Fig. 5. We set up 50 number of query samples, where dots of the same color indicate the same category. It can be seen that the distribution of samples of the same category in the feature space exhibits a better clustering structure under different settings of the number of query samples using our proposed method. The intra-class compactness of each category and the inter-class variability among different categories are significantly improved. This reflects that the introduction of multi-example learning in small-sample learning using our proposed method can essentially improve the representation of the input image in the feature space, which fully demonstrates the effectiveness of the proposed method in this paper.

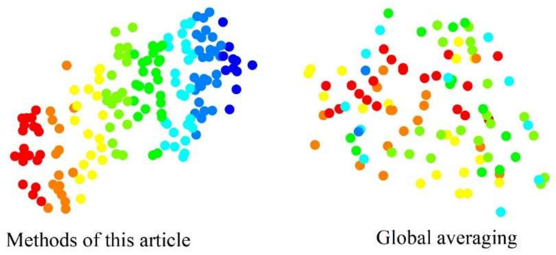


Figure 5: Feature visualization experiment

VIII. Conclusion

In this paper, an adaptive module approach is introduced from in the pre-training model to realize the efficient migration of natural language processing and computer vision macromodels. In the real case analysis, the AABC model with Adapters adapter component mechanism is proposed. By validating and analyzing the case study, the following results can be drawn:

(1) Compared to other efficient transfer learning methods, Adapters based on the BERT-large structure achieves the best average performance, and its excellent overall performance provides the basis for subsequent case studies.

(2) From the perspective of the average number of parameters, the AABC model has the lowest average number of parameters among several models and the best performance. On the MSRA dataset, the AABC model has the highest F1 value of 94.41%.

(3) Adapter is able to enhance the learning ability of ViT in data-constrained scenarios, enabling the visual Transformer to obtain better image representations.

About the Author

Kezhi Zhen, male, Han Nationality, October 1986, undergraduate, engineer, Bijie City, Guizhou Province, research direction: intelligent manufacturing, information technology, digital transformation of enterprises.

References

[1] Lin, H. Y. (2022). Large-scale artificial intelligence models. Computer, 55(05), 76-80.

[2] Pan, L., Zhao, Z., Lu, Y., Tang, K., Fu, L., Liang, Q., & Peng, S. (2024). Opportunities and challenges in the application of large artificial intelligence models in radiology. Meta-Radiology, 100080.

[3] Zhong, Y., Chen, Y. J., Zhou, Y., Lyu, Y. A. H., Yin, J. J., & Gao, Y. J. (2023). The artificial intelligence large language models and neuropsychiatry practice and research ethic. Asian journal of psychiatry, 84, 103577.

[4] Edwards, R. E., New, J., Parker, L. E., Cui, B., & Dong, J. (2017). Constructing large scale surrogate models from big data and artificial intelligence. Applied energy, 202, 685-699.

[5] Barreto, F., Moharkar, L., Shirodkar, M., Sarode, V., Gonsalves, S., & Johns, A. (2023, February). Generative artificial intelligence: Opportunities and challenges of large language models. In International conference on intelligent computing and networking (pp. 545-553). Singapore: Springer Nature Singapore.

[6] Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big data, 3, 1-40.

[7] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... & He, Q. (2020). A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1), 43-76.

[8] Pan, S. J. (2020). Transfer learning. Learning, 21, 1-2.

[9] Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., & Azim, M. A. (2022). Transfer learning: a friendly introduction. Journal of Big Data, 9(1), 102.

[10] Ying, W., Zhang, Y., Huang, J., & Yang, Q. (2018, July). Transfer learning via learning to transfer. In International conference on machine learning (pp. 5085-5094). PMLR.

[11] Neyshabur, B., Sedghi, H., & Zhang, C. (2020). What is being transferred in transfer learning?. Advances in neural information processing systems, 33, 512-523.

[12] Day, O., & Khoshgoftaar, T. M. (2017). A survey on heterogeneous transfer learning. Journal of Big Data, 4, 1-42.

[13] Iman, M., Arabnia, H. R., & Rasheed, K. (2023). A review of deep transfer learning and recent advancements. Technologies, 11(2), 40.

[14] Zhu, Z., Lin, K., Jain, A. K., & Zhou, J. (2023). Transfer learning in deep reinforcement learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(11), 13344-13362.

[15] Agarwal, N., Sondhi, A., Chopra, K., & Singh, G. (2021). Transfer learning: Survey and classification. Smart Innovations in Communication and Computational Sciences: Proceedings of ICSICCS 2020, 145-155.

[16] Ali, A. H., Yaseen, M. G., Aljanabi, M., & Abed, S. A. (2023). Transfer learning: A new promising techniques. Mesopotamian Journal of Big Data, 2023, 29-30.

[17] Cai, C., Wang, S., Xu, Y., Zhang, W., Tang, K., Ouyang, Q., ... & Pei, J. (2020). Transfer learning for drug discovery. Journal of Medicinal Chemistry, 63(16), 8683-8694.

[18] Wang, J., Chen, Y., Feng, W., Yu, H., Huang, M., & Yang, Q. (2020). Transfer learning with dynamic distribution adaptation. ACM Transactions on Intelligent Systems and Technology (TIST), 11(1), 1-25.

[19] Bo Zhang,Luoxi Wang,Cheng Zhang,Ran Zhao & Jinlu Sun. (2025) .No-reference image quality assessment based on improved vision transformer and transfer learning. Signal Processing: Image Communication,135,117282-117282.

[20] Cuiying Lin,Yun Kong,Qinkai Han,Xiantao Zhang,Junyu Qi,Meng Rao... & Fulei Chu. (2025) .An unsupervised multi-level fusion domain adaptation method for transfer diagnosis under time-varying working conditions. Mechanical Systems and Signal Processing,228,112458-112458.

[21] He Wang,Tianyang Xu,Zhangyong Tang,Xiao Jun Wu & Josef Kittler. (2025) .Multi-modal adapter for RGB-T tracking. Information Fusion,118,102940-102940.

[22] Yun Chen,Gengyang Lu,Ke Wang,Shu Chen & Chenfei Duan. (2024) .Knowledge graph for safety management standards of water conservancy construction engineering. Automation in Construction,168(PB),105873-105873.