Advances in Artificial Intelligence and Machine Learning; Research 4 (1) 1977-1990 Received 10-01-2024; Accepted 03-03-2024; Published 10-03-2024

Efficient Multi-Domain Text Recognition Deep Neural Network Parameterization With Residual Adapters

Jiayou Chao jiayou.chao@stonybrook.edu
Department of Applied Mathematics and Statistics
Stony Brook University
Stony Brook, NY 11794, USA

Wei Zhu wei.zhu@stonybrook.edu
Department of Applied Mathematics and Statistics
Stony Brook University
Stony Brook, NY 11794, USA

Corresponding Author: Jiayou Chao

Abstract

Recent advancements in deep neural networks have markedly enhanced the performance of computer vision tasks, yet the specialized nature of these networks often necessitates extensive data and high computational power. Addressing these requirements, this study presents a novel neural network model adept at optical character recognition (OCR) across diverse domains, leveraging the strengths of multi-task learning to improve efficiency and generalization. The model is designed to achieve rapid adaptation to new domains, maintain a compact size conducive to reduced computational resource demand, ensure high accuracy, retain knowledge from previous learning experiences, and allow for domain-specific performance improvements without the need to retrain entirely. Rigorous evaluation on open datasets has validated the model's ability to significantly lower the number of trainable parameters without sacrificing performance, indicating its potential as a scalable and adaptable solution in the field of computer vision, particularly for applications in optical text recognition.

Keywords: Deep neural network, Optical character recognition, Multi-domain adapter, Multi-task learning, Continual learning.

1. INTRODUCTION

As deep neural networks continue to dramatically improve results for nearly all traditional computer vision problems, the community has begun to shift its focus to more ambitious objectives [1]. One prevalent pragmatic constraint associated with deep neural networks is related to their notable propensity for specialization towards a singular task, as well as their substantial requirements in terms of data size and computational resources [2, 3]. This holds particularly true for the most efficacious deep neural networks, which are commonly trained on extensive datasets comprising

Citation: Jiayou Chao and Wei Zhu. Efficient Multi-Domain Text Recognition Deep Neural Network Parameterization With Residual Adapters. Advances in Artificial Intelligence and Machine Learning. 2024;4(1):113.

millions of images. This issue poses a challenge in numerous applications where the available data is constrained and the computational resources are limited [4, 5]. An additional limitation of this method is its lack of scalability, particularly when confronted with an increasing number of problems to be resolved. Furthermore, this approach lacks efficiency due to the repetitive acquisition of the same information and the inability of models to transfer knowledge across different tasks [6–10]. The concept of employing a singular model to address multiple tasks is highly attractive due to its capacity to facilitate the transfer of acquired knowledge from one task to another. The significance of this matter is particularly pronounced when considering tasks that exhibit interrelatedness, such as object detection and segmentation, or object detection and classification. In this particular scenario, the acquired knowledge from one task can be effectively utilized to enhance the performance of the other task.

The increasing focus on developing data representations that exhibit robust performance across diverse problem domains and datasets is indeed noteworthy. The realization that such adaptable representations are essential for developing machine learning systems that can effectively generalize beyond the constrictions of particular tasks and datasets is what is driving this burgeoning interest. As the field of artificial intelligence continues to evolve, the ability to create models that can seamlessly adapt and maintain high levels of accuracy across a variety of challenges has become a pinnacle pursuit for researchers and practitioners alike [11, 12]. Most of the works in this area focus on image classification [6, 7, 13–17] or text classification [16, 18–21], yet their application in optical character recognition (OCR) remains somewhat unexplored, to the best of our knowledge. The incorporation of multi-task learning in OCR offers substantial benefits, particularly in practical applications. A key aspect of OCR is the unique and valuable information each entry provides, which can significantly enhance recognition accuracy and speed [22–24]. For example, when digitalizing a business form, recognizing an entry as a phone number immediately implies that it contains only numerical digits. This context-specific insight is crucial for improving recognition accuracy and efficiency. Similarly, when processing entries in foreign languages, incorporating language-specific information can substantially reduce recognition errors. Leveraging domain-specific knowledge further refines the accuracy of OCR systems. Therefore, a versatile OCR model adept at utilizing domain-specific information is immensely beneficial for a wide array of real-world OCR scenarios, underscoring the value of such an approach.

The innovation introduced in this research paper is a multi-domain neural network architecture designed specifically for enhancing OCR across diverse applications. This architecture capitalizes on the concept of dynamic adaptability, employing adapter modules that function as interstitial components within the established neural network framework. These modules serve as vectors for domain-specific parameters, strategically integrated within a preexisting, pre-trained model to fine-tune its feature extraction capabilities to suit new tasks. The introduction of adapter modules into the neural network is a strategic response to the issue of catastrophic forgetting, a problem where sequential learning of new tasks can lead to a degradation of performance on previously learned tasks [25–27]. By preserving the adapters corresponding to previous domains intact, the network maintains its proficiency across all learned tasks. The architecture thus proposes a scalable solution that promotes efficient adaptation without compromising historical knowledge. The intricacies of the proposed methodology necessitate precise domain specification for optimal feature extraction during data input. In instances where the domain remains ambiguous, an ancillary neural network is suggested as a viable mechanism for domain prediction, before processing by the primary OCR-focused architecture [28].

The efficacy of the proposed multi-domain neural network architecture was rigorously assessed using publicly available datasets, offering a transparent and replicable benchmark for the evaluation process. The experiments underscored the model's proficiency in striking a balance between model complexity and performance. Notably, the architecture demonstrated a marked reduction in the number of trainable parameters—indicative of an efficient parameterization—without compromising the integrity of its OCR capabilities. The results affirm the model's potential as a scalable and adaptable solution for OCR challenges across a multitude of domains. The remainder of this paper is organized as follows: Section 2 provides a comprehensive review of the relevant literature, Section 3 outlines the proposed methodology, Section 4 presents the experimental results, and Section 5 concludes the paper with a discussion of the findings and future research directions. The code for this paper is available at https://github.com/Jiayou-Chao/Multi-domain-OCR.

2. RELATED WORK

Training a deep learning model for multi-domains or general purposes has long been the focus of academic research. The major topics usually include multi-task learning, adapting new domains and avoiding forgetting.

Multi-task learning (MTL) aims at learning multiple related tasks simultaneously by sharing information and computation among them. Early work [29] in this area focuses on deep neural network (DNN) models which share weights in the earlier layers and use specialized ones in the later layers. This line of research focuses on learning a diverse set of tasks in the same visual domain. In this case, the knowledge learned for one task can be used to improve the performance of the other one. However, this approach usually requires the different tasks to be related to each other, and to share the same input data. If the problem setting is an input distribution $p(X)$, then the goal of MTL is to learn a single model $f(X)$ that can be used to address multiple different tasks $T_1, T_2, \ldots, T_n$, where the label distribution $p(Y_i|X)$ is different for each task. In this case, the model $f(X)$ is a function of the input X and the task T_i, and it outputs the label Y.

Sequential Learning (Incremental Learning or Life-long Learning), is a theoretical framework that aims to acquire a model for a substantial number of tasks sequentially, while retaining the information acquired from previous tasks. Notably, this approach assumes that the data from previous tasks are no longer accessible during the training of subsequent tasks. Catastrophic forgetting is possible in sequential learning, albeit it does not usually occur [11, 30]. There is no guarantee that it can retain prior knowledge. If sometimes one is interested in maximizing the performance on a specific new task, sequential learning can be used as a form of initialization for the new task. In this case, the model is trained on the old tasks, and then it is fine-tuned on the new task. This approach is called Transfer Learning (TL).

Progressive Learning is yet another concept to solve complex sequences of tasks. This approach excels in leveraging knowledge transfer while avoiding catastrophic forgetting, distinguishing it from traditional methods. As elucidated by [31], Progressive Learning models are uniquely designed to be immune to forgetting and efficiently utilize prior knowledge through lateral connections to previously learned features. The general procedure of a Progressive Learning model begins with training a deep learning model on an initial task. Upon completion, the model's weights are frozen to preserve the learned knowledge. Subsequently, a new model is trained on a second task, with

its weights also being frozen post-training. Critically, the weights of this second model are interconnected with the first model's weights via lateral connections, facilitating knowledge transfer and feature integration. This process is iteratively applied to each subsequent task, culminating in a final model that amalgamates the knowledge acquired from all tasks. [32] highlights the effectiveness of this method in maintaining a robust knowledge base across multiple tasks. However, the scalability of Progressive Networks presents a significant challenge. As the number of tasks increases, the model's parameters grow exponentially, which poses limitations for practical applications. This issue, highlighted in recent studies, suggests a need for innovative approaches to manage parameter growth efficiently. Moreover, the implementation of Progressive Learning models, particularly in learning from a sequence of tasks, demands intricate design and execution strategies. The efficacy of these models is highly contingent on the optimal sequencing of task execution, a challenge that remains an active area of research.

Adapters serve as a lightweight alternative to complete model fine-tuning, as they involve the introduction of a small collection of parameters specifically at each backbone layer. Adapters address many constraints commonly encountered in the process of complete model fine-tuning. They exhibit advantages such as parameter efficiency, accelerated training iterations, and the ability to be shared and combined owing to their modular and compact nature. In addition, it is worth noting that adapters typically provide comparable performance to the current leading approach of full fine-tuning [30, 33–36].

3. METHOD

The proposed framework delineates an innovative Convolutional Recurrent Neural Network (CRNN) architecture that is holistically trainable and comprises a sophisticated feature extraction network augmented with adapter modules, in addition to a sequential network. The fundamental component of the feature extraction network is a convolutional neural network that draws inspiration from the ResNet architecture introduced by [37], specifically engineered to distill features from the input imagery. This network diverges from the quintessential ResNet by the incorporation of residual adapters after each stacked residual block within the feature extraction network. These residual adapters, inspired by the work of [38], are constituted by a vector of 1×1 convolutional filter banks functioning in concert with an identity skip connection, and are tailored to fine-tune the extracted features to various tasks.

The sequential aspect of the network employs a transformer model, a construct that excels in encoding sequential information [39]. This section of the network is further enhanced by bottleneck adapters, an innovation introduced by [21], which are situated subsequent to the multi-head attention and feed-forward layers in the transformer. These adapters are notable for their limited number of parameters relative to the attention and feedforward layers prevalent in traditional models. They also feature a skip-connection, enhancing the efficiency of training.

In the context of adapter tuning, the process is meticulously selective, concentrating solely on the parameters of the adapters, normalization layers, and the final classification layer, fostering a disentangled form of learning. The network as a whole is subject to an end-to-end training regimen. The model's architectural design, as depicted in FIGURE 1, exemplifies the cohesive interplay

between various innovative components designed to optimize character prediction accuracy from input images.

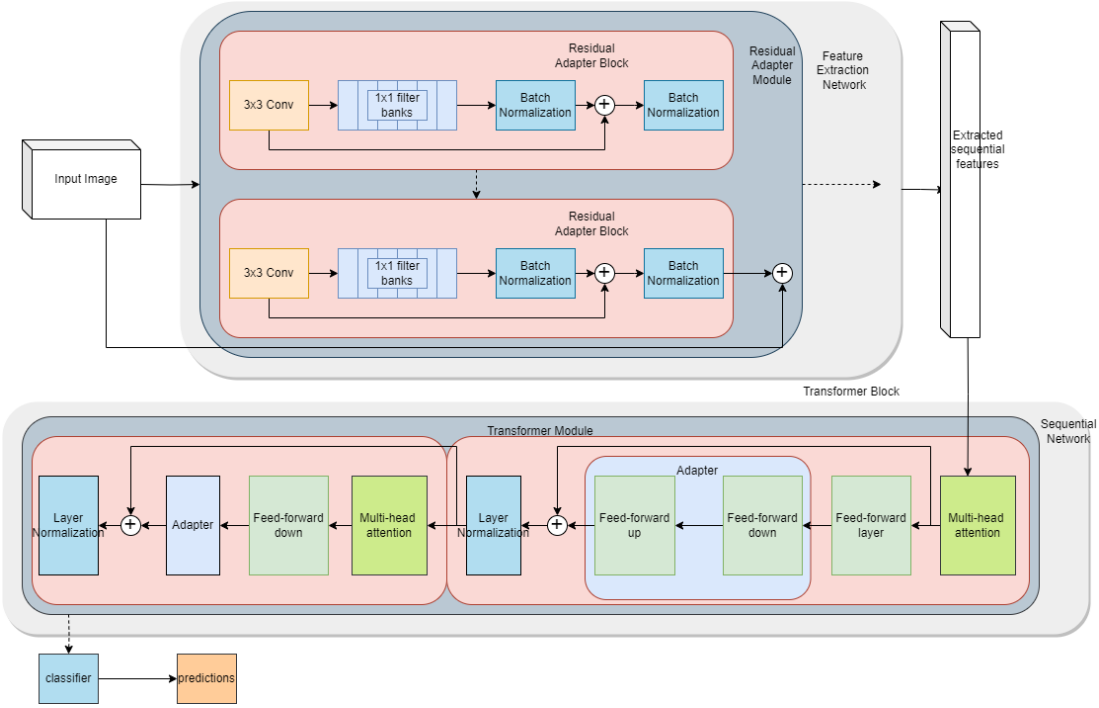

Figure 1: The proposed model's architecture. It primarily comprises a feature extraction network and a sequential network, augmented with adapter modules. This architecture is schematically depicted with solid lines representing data flow and dashed lines signifying the repetition of identical modules. The adapter module consists of a vector of identical adapters. Such a configuration facilitates a modular and scalable design.

The feature extraction network. Let's consider an input image $\mathcal{X} \in \mathbb{R}^{H\times W\times C}$, where H, W, and C represent the height, width, and number of channels of the image, respectively. The feature extraction network, modeled on the principles of a Resnet-like convolutional neural network, operates on $\mathcal{X}$ to extract salient features. Denoted as $f_{\theta,\phi}(\mathcal{X})$, this network encompasses two types of parameters: domain-agnostic parameters θ, which are common across various domains, and domain-specific parameters ϕ, which are unique to specific application areas. The network's architecture is composed of a series of stacked residual modules, each containing one or more residual blocks. These blocks are the fundamental units that process the input image. Specifically, the i-th residual block in the j-th module, $g^{(j)}_{\theta_i,\phi_i}(\mathcal{X})$, can be expressed as: $g^{(j)}_{\theta_i,\phi_i}(\mathcal{X}) = a((I + \alpha_i)(\omega_i * (\mathcal{X})))$, where $i = 1, 2, \ldots$; a denotes the activation function; I is the identity skip connection, facilitating gradient flow during training; ω_i are the domain-agnostic weights; α_i are the domain-specific weights; and $*$ symbolizes the convolution operation. For the sake of simplicity, normalization layers, typically essential for stabilizing training, are not included in this formula. Each residual module, represented as $G_j(\mathcal{X})$, is the functional composition of its constituent residual blocks: $G_j(\mathcal{X}) = g_1 \circ g_2 \circ \cdots \circ g_n(\mathcal{X})$, with n indicating the number of residual blocks in the j-th module and $\circ$ symbolizing the composition of functions. Thus, the feature extraction network can be holistically

represented as: $f_{\theta,\phi}(\mathcal{X}) = G_1 \circ G_2 \circ \cdots \circ G_m(\mathcal{X})$, where G_j denotes the j-th residual module and m the total number of modules. The output of this network is $\mathcal{F} \in \mathbb{R}^{1\times W'\times C'}$, where W' and C' are the width and number of channels of the feature map, respectively. Notably, the height is deliberately reduced to 1 for seamless integration into subsequent sequential models.

The sequential network, as a core component of our model, is a transformer-based architecture integrated with specialized bottleneck adapter modules. This network takes the features extracted by the feature extraction network as input and outputs the final predictions. At the heart of this network are the bottleneck adapters, each comprising three distinct layers: a linear down-projection layer, a non-linearity layer, and a linear up-projection layer. These layers work in tandem to refine the feature representation: the down-projection layer condenses the input features from the transformer layer into a lower-dimensional space, represented mathematically as $W_d\mathcal{X}$. Here, W_d denotes the weights of the down-projection layer. After down-projection, a non-linear function, symbolized by a, is applied to these features. This step introduces non-linearity to the model, enhancing its capacity to capture complex patterns. Functions like ReLU or tanh are typically used for this purpose. The transformed features are then projected back to their original dimension through the up-projection layer, with weights represented by W_u. The adapter module can thus be denoted as $h_{w_d,w_u}(\mathcal{X}) = W_u a(W_d\mathcal{X})$. The down-projection layer and up-projection layer are typically much smaller than the transformer layer itself, which makes the adapter layers much faster to train and allows them to be added to a larger number of layers in the transformer model. These adapters are strategically placed after the attention or feed-forward layers in the transformer. This placement is critical as these layers are instrumental in learning complex representations. By integrating adapters here, we can fine-tune the model more effectively for specific tasks such as natural language understanding, machine translation, and text summarization. This approach has been corroborated by various studies [21, 33, 34, 40, 41]. The final stage involves the application of a softmax function to convert the output into a probability distribution over the output classes. The output $\mathcal{Y}$, denoted as $\mathcal{Y} \in \mathbb{R}^{1\times W'\times C'}$, reflects the width W' and the number of classes C'. To compute the loss, $\mathcal{Y}$ is fed into the Connectionist Temporal Classification (CTC) loss function [42]. This loss function is particularly effective for sequence-to-sequence problems, providing a robust framework for model training and optimization.

Training the Adapter Modules. The backbone of the network, namely the feature extraction network and the sequential network, is initially trained on a large dataset, with the exclusion of the adapter modules. In general, the efficacy of the backbone in extracting useful features correlates strongly with the diversity and size of the dataset. Exclusively training on a limited dataset inside a singular domain has the potential to result in overfitting, hence hindering the model's ability to effectively generalize across other domains. The following step of training involves incorporating a new task into the model and subsequently optimizing the model's performance on it. The new task may encompass a novel domain such as a different language, a distinct font, or a new environmental configuration. The existing model has the capability to be modified and applied to the new task while retaining knowledge of the past tasks. The prevention of forgetting is accomplished by freezing the weights of the backbone and solely updating the weights of the corresponding adapter modules during the second phase of training. The adapter modules can be conceptualized as a collection of task-specific modules that are incorporated to enhance the efficiency of feature extraction for the novel job. During the process of backpropagation, the data originating from the new domain is exclusively directed through its corresponding adapter module, while the remaining adapter modules remain unaffected. Therefore, the performance of the model on different domains is unaffected. Consequently, the model can accurately identify characters from several domains

without requiring the training of separate models for each domain. By freezing the weights of the backbone, the training process benefits from a significant reduction in the number of parameters to be optimized. This leads to a faster training period and mitigates the potential problem of overfitting. The utilization of information from the backbone by the adapters, which have been trained on a substantial dataset to extract the most valuable characteristics, leads to a reduction in the needed data size and number of training epochs for the adapter. Hence, our model exhibits a high degree of adaptability in character recognition across many domains, resulting in efficient utilization of training time and resources.

4. EXPERIMENTS

Datasets. We utilize the Meta Self-Learning for Multi-Source Domain Adaptation: A Benchmark dataset, as documented by [43], to validate the efficacy of our proposed model. This dataset, which is publicly available, comprises various Chinese text images curated specifically for multi-source domain adaptation studies. The selection of a Chinese dataset is motivated by the linguistic complexity of Chinese, with its character set exceeding 10,000 unique glyphs (11,376 in our study), as opposed to the mere 35 characters found in traditional English-digit datasets. Such complexity presents a greater challenge in character recognition, potentially enhancing the network's capacity for general feature learning and subsequent domain adaptability.

The dataset is categorized into five distinct text image types: handwritten, street scene, document, synthetic, and car license plates. For the initial experiment, the document category—consisting of 1,614,955 training and 179,345 test images—is employed to train the network's backbone. These images, which feature a uniform font and clean background, serve as an ideal starting point. Subsequently, we evaluate the model's domain adaptation capability using two additional categories: car license plates and synthetic images. The car license plate domain, containing 187,136 training and 20,792 test images, is considered less challenging due to the uniformity of font, clean background, and limited character set. Conversely, the synthetic domain, with its diverse fonts, noisy backgrounds, and non-meaningful synthetic sentences, creates a more strenuous test environment. This category includes 999,558 training and 111,062 test images. Each class in the dataset represents a distinct Chinese character, totaling 11,376 classes. The images are preprocessed to a standard size of 32x128 pixels and normalized to ensure a mean of zero and unit variance, facilitating consistent input for the network.

Implementation Details. Our feature extraction network employs a relatively shallow architecture due to the size of our training data. It comprises 4 residual modules, each containing 2 residual blocks. The sequential network utilizes a multi-head attention mechanism with 8 heads and incorporates positional encoding to capture sequence information. Network feedforward layers consist of 128 hidden units each.

For the training regimen, we employ a batch size of 256 and the Adam optimizer with an initial learning rate of 1×10^{-5}. The learning rate is subjected to a decay by a factor of 0.1 every five epochs. The first two epochs function as a warmup phase for the learning rate. We train the backbone for a total of 20 epochs under two distinct experimental scenarios to evaluate the adaptability of the model:

1. In the first scenario, the backbone is exclusively trained using the document dataset to test the adapters' generalization to unseen datasets.
2. In the second scenario, the backbone training also includes data from the car license and synthetic datasets to explore how well the adapters can optimize performance within specific domains.

Following the backbone training, we train the adapter models on the car license dataset or the synthetic dataset for 20 additional epochs. We compare the outcomes of exclusively training the adapter modules (adapter method) versus jointly training the adapter modules and the backbone (finetuning method).

Performance is assessed using three metrics: character accuracy, word accuracy, and recall. Character accuracy (precision) is the fraction of correctly identified characters in the OCR output relative to the ground truth. Word accuracy, effectively image-level accuracy due to the absence of Chinese word segmentation, measures correct word recognition. Recall is defined as $Recall = \frac{TP}{TP+FN}$, where TP is the count of accurately identified characters or words, and FN represents those that the OCR system failed to identify. As an OCR engine tuned for high recall may inadvertently reduce precision, achieving a balance between these metrics is vital. This consideration is particularly crucial in fields that demand comprehensive data extraction, such as legal and medical document processing. Our model is developed in PyTorch and trained on a suite of eight NVIDIA Tesla K80 GPUs.

Training Backone Results. The details pertaining to the backbone models can be found in TABLE 1. In the 1st experiment, the backbone is solely trained using the document dataset. The evaluation is conducted on all three datasets, namely document, car license, and synthetic. The backbone model demonstrates a character accuracy rate of 94.93%, a word accuracy rate of 65.15%, and a recall rate of 94.91% when evaluated on the document dataset. The metrics exhibit values in proximity to zero when the model undergoes evaluation on the car license or the synthetic dataset. This observation indicates that the current backbone model lacks the capacity to generalize effectively to unfamiliar domains. Consequently, we employ this model as a means to evaluate the efficacy of adapters in facilitating generalization to unseen domains. In the 2nd experiment, the backbone undergoes training and subsequent evaluation on all three datasets. On the document dataset, the backbone model exhibits a character accuracy of 99.29%, a word accuracy of 93.23%, and a recall rate of 99.26%. The current backbone demonstrates superior performance across all metrics in comparison to the previous experiment, which solely utilized the document dataset for training. The performance metrics of the backbone model on the car license dataset are as follows: 99.80% character accuracy, 98.52% word accuracy, and 99.79% recall. The performance metrics of the backbone model on the synthetic dataset are as follows: 92.17% character accuracy, 64.02% word accuracy, and 92.13% recall. This observation indicates that there is potential for improvement in the model's performance on the synthetic dataset, particularly in terms of word accuracy. The performance exhibited in this instance is highly commendable; however, it is important to acknowledge that the second backbone has been trained on a significantly larger dataset, thereby necessitating a substantial investment of time and computational resources. The backbone model is subsequently employed to train the adapter models in order to evaluate their potential for improving performance within a particular domain, while maintaining the same level of performance across other domains.

Table 1: The information of the backbone models. In the 1st experiment, the backbone is trained only on the document dataset. The model performs poorly on unseen datasets like the car license dataset and the synthetic dataset. In the 2nd experiment, the backbone is trained on all of the three datasets (document, car license, and synthetic). Three metrics, character accuracy, word accuracy and recall, are used to evaluate the backbone on all three datasets.

Evaluation Dataset	Character Accuracy	Word Accuracy	Recall
Experiment: 1			
document	94.93%	65.15%	94.91%
car license	2.44%	0.00%	1.84%
synthetic	0.43%	0.00%	0.55%
Experiment: 2			
document	99.29%	93.32%	99.26%
car license	99.80%	98.52%	99.79%
synthetic	92.17%	64.02%	92.13%

Training Adapter Results. The results of training only adapters in two differenet scenarios are shown in TABLE 2, and the results are compared with the finetuning method which updates all parameters in comparison to updating only adapters. When using the finetuning method, there are a total of 21,636,658 trainable parameters (the parameters of the irrelevant domains are excluded when calculating). In contrast, there are only 7,590,930 trainable parameters when using the adapter method, which is a marked reduction of 64.93%. Delving into the specifics, Experiment 1 unveils an enhancement in character accuracy from a meager 2.44% to an impressive 99.64% using adapters and a slightly higher 99.83% with finetuning on the car license dataset. This is a dataset that is considered relatively straightforward due to its simplicity, hinting that the adapter method can indeed parallel the performance of full finetuning on simple tasks. Moreover, this method has the added advantage of retaining prior knowledge without the risk of domain forgetting—a common hurdle in finetuning. When examining word accuracy, the adapter method even outperforms finetuning by achieving 99.63% compared to 98.81%. However, the recall rate with the adapter method at 97.50% underperforms the finetuning method at 99.83%. In the context of the synthetic dataset, which poses a greater challenge due to its inclusion of unseen characters and nonsensical sentences, the adapter method maintains its competence. Character accuracy ascends from a paltry 0.43% to 96.13% and recall from 0.55% to 96.04% for the adapter method, nearing the finetuning outcomes of 98.94% and 93.91%, respectively. Although the word accuracy for adapter lags at 79.81% versus the finetuning's 98.89%, the marginal disparity highlights a limitation in the backbone's generalization capabilities to unfamiliar domains. The evidence, as outlined here, suggests that adapters offer a promising alternative to full model finetuning, particularly in scenarios where computational efficiency and memory preservation are paramount. Despite the occasional dips in performance on more complex datasets, the adapter method's impressive recall and character accuracy signify its potential as a viable strategy for domain-specific neural network training.

Experiment 2 offers an extension of the inquiry into the efficacy of adapter training versus full model finetuning, this time with the backbone exposed to a more diverse training set. Indeed, the performance on the car license datasets exhibited minimal variance, which can be attributed to the already optimized state of the backbone for this particular domain, evidencing a saturation

Table 2: The training and comparison results of the adapter. The adapter method refers to the training of adapter modules while keeping the backbone frozen. The finetuning approach involves training the adapter modules and the backbone simultaneously. In addition to evaluating character accuracy, word accuracy, and recall, we also include the number of trainable parameters in the last table column. Consequently, the adapter method shows comparable performance to the finetuning method on a simple new domain, while its effectiveness is constrained by the underlying backbone on a more intricate new domain. The adapter method does not have the risk of forgetting the previous domain.

Method	Evaluation Dataset	Character Acc	Word Acc	Recall	Trainable Param
Experiment: 1					
Adapter	car license	99.64%	99.63%	97.50%	7,590,930
Finetuning	car license	99.83%	98.81%	99.83%	21,636,658
Adapter	synthetic	96.13%	79.81%	96.04%	7,590,930
Finetuning	synthetic	98.94%	98.89%	93.91%	21,636,658
Experiment: 2					
Adapter	car license	99.64%	99.63%	97.50%	7,590,930
Finetuning	car license	99.97%	99.97%	99.97%	21,636,658
Adapter	synthetic	98.48%	98.44%	91.56%	7,590,930
Finetuning	synthetic	98.94%	98.89%	93.90%	21,636,658

point in learning. On the synthetic dataset, which serves as the benchmark for complexity within this study, both methods demonstrated significant improvements in character and word accuracies, achieving near-parity. The adapter method enhanced character accuracy from 92.17% to 98.48%, while finetuning edged slightly ahead with a rise to 98.84%. Moreover, word accuracy for adapters soared from 64.02% to a remarkable 98.44%, closely shadowing the finetuning result of 98.89%. Interestingly, recall rates exhibited a different trend, remaining relatively stable for the adapter method with a slight decline from 92.13% to 91.56%, whereas finetuning saw a modest increase to 93.39%.

These outcomes are particularly noteworthy in that they illustrate the adapter method's robustness when the backbone network is enriched with diverse training data. In essence, the adapter-equipped model nearly matches the finetuning method in performance, even when confronted with complex datasets. Moreover, it underscores the adapter method's proficiency in domain-specific enhancement without compromising the existing knowledge encoded in the backbone—a notable advantage over training all domains concurrently on the backbone. Thus, Experiment 2 reinforces the conclusion drawn from the initial experiment: the adapter method not only boasts fewer parameters and reduced risk of catastrophic forgetting but also exhibits the potential to deliver performance comparable to finetuning, even under the pressures of dataset complexity. Given the additional benefits of selective domain enhancement, the adapter method emerges as a compelling choice for efficient and effective neural network training, especially when considering the computational and memory constraints often encountered in real-world applications.

5. CONCLUSION

In this paper, we introduce an innovative adapter network designed for multi-source OCR and present its effectiveness over traditional domain adaptation methods. The empirical evidence from the conducted experiments indicates that the adapter network outstrips the traditional methods. In comparison to the alternative strategy, which involves domain-specific fine-tuning of the backbone model, the adapter network shows an equivalent aptitude in performance. However, the standout feature of the adapter network is its reduction in the quantity of parameters that require training. This reduction is not trivial—it significantly streamlines the process of adapting to new domains, a vital factor in business environments that demand both quick adaptability and the capacity to handle multiple domains simultaneously.

Notwithstanding these advantages, the study also reveals a shortcoming of the adapter network when dealing with complex domains—as evidenced by the results from the synthetic dataset. When the backbone of the network is relatively weak, the ability of the adapter network to achieve commendable accuracy in intricate domains is compromised. The findings underscore the necessity of training a robust backbone model on an extensive dataset. Such comprehensive training is imperative for the model to discern and assimilate the essential characteristics inherent to the entities within the domain, thereby enhancing the model's capability to render high accuracy in demanding and complex domains.

In essence, the research posits that while the adapter network holds promise for flexible and efficient domain adaptation, the strength of the underlying model is a fundamental precept that governs the ultimate performance and robustness in challenging domain-specific tasks.

6. DATA AVAILABILITY

The data that support the findings of this study are openly available at Meta Self-Learning for Multi-Source Domain Adaptation: A Benchmark (https://github.com/bupt-ai-cz/Meta-SelfLearning).

References

[1] LeCun Y, Bengio Y, Hinton G. Deep Learning. Nature. 2015;521:436-444.

[2] Ruder S. An Overview of Multi-Task Learning in Deep Neural Networks. 2017. Arxiv Preprint: https://arxiv.org/pdf/1706.05098.pdf

[3] Reeve HWJ, Cannings TI, Samworth RJ. Adaptive Transfer Learning. Ann Statist. 2021;49: 3618-3649.

[4] Li B, Yan J, Wu W, Zhu Z, Hu X, et al. High Performance Visual Tracking With Siamese Region Proposal Network. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2018:8971-8980.

[5] Cannings TI, Fan Y, Samworth RJ. Classification With Imperfect Training Labels. Biometrika. 2020;107:311-330.

[6] Misra I, Shrivastava A, Gupta A, Hebert M. Cross-Stitch Networks for Multi-Task Learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016:3994-4003.

[7] Liu S, James S, Davison AJ, Johns E. Auto-Lambda: Disentangling Dynamic Task Relationships. 2022. Arxiv Preprint: https://arxiv.org/pdf/2202.03091.pdf

[8] Liu S, Johns E, Davison AJ. End-To-End Multi-Task Learning With Attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019: 1871-1880.

[9] Sinha A, Namkoong H, Volpi R, J Duchi. Certifying Some Distributional Robustness With Principled Adversarial Training. 2020. Arxiv Preprint: https://arxiv.org/pdf/1710.10571v5.pdf

[10] Rothenhäusler D, Bühlmann P. Distributionally Robust and Generalizable Inference. Statist Sci. 2023;38:527-542.

[11] Parisi GI, Kemker R, Part JL, Kanan C, Wermter S, et al. Continual Lifelong Learning With Neural Networks: A Review. Neural Netw. 2019;113:54-71.

[12] Wang L, Zhang X, Su H, Zhu J. A Comprehensive Survey of Continual Learning: Theory, Method and Application. 2023. Arxiv Preprint: https://arxiv.org/pdf/2302.00487.pdf

[13] Bhattacharjee D, Süsstrunk S, Salzmann M. Vision Transformer Adapters for Generalizable Multitask Learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2023:19015-19026.

[14] Zhao H, Zhang S, Wu G, Moura JM, Costeira JP, et al. Adversarial Multiple Source Domain Adaptation. Adv Neural Inf Process Syst. 2018;31.

[15] Peng X, Bai Q, Xia X, Huang Z, Saenko K, et al. Moment Matching for Multi-Source Domain Adaptation. In: Proceedings of the IEEE/CVF international conference on computer vision. 2019:1406-1415.

[16] Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, et al. Domain-adversarial training of neural networks. 2016. Arxiv Preprint: https://arxiv.org/pdf/1505.07818.pdf

[17] Rebuffi S-A, Vedaldi A, Bilen H. Efficient Parametrization of Multi-Domain Deep Neural Networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE Publications. Salt Lake City: IEEE Publications. 2018:8119-8127.

[18] Liu P, Qiu X, Huang X. Adversarial Multi-Task Learning for Text Classification. 2017. Arxiv Priprint: https://arxiv.org/pdf/1704.05742.pdf

[19] Guo H, Pasunuru R, Bansal M. Multi-Source Domain Adaptation for Text Classification via Distancenet-Bandits. 2020. Arxiv Preprint: https://arxiv.org/pdf/2001.04362.pdf

[20] Wang Z, Liu X, Yang P, Liu S, Wang Z, et al. Cross-Lingual Text Classification With Heterogeneous Graph Neural Network. 2021. Arxiv Preprint: https://arxiv.org/pdf/2105.11246.pdf

[21] Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, De Laroussilhe Q, et al. Parameter-Efficient Transfer Learning for NLP. 2019: Arxiv Preprint: https://arxiv.org/pdf/1902.00751.pdf

[22] Veeramachaneni S, Nagy G. Adaptive Classifiers for Multisource OCR. Int J Doc Anal Recognit. 2003;6:154-166.

[23] Mathis C, Breuel T. Classification using a hierarchical Bayesian approach. Int. Conf. Pattern Recognit. Quebec City, QC, Canada. 2002;4:103-106.

[24] Ho TK, Nagy G. OCR with no shape training. Proceedings 15th Int. Conf. Pattern Recognit. ICPR-2000, Barcelona, Spain, 2000;4:27-30.

[25] French RM. Catastrophic Forgetting in Connectionist Networks. Trends Cogn Sci. 1999;3:128-135.

[26] Goodfellow IJ, Mirza M, Xiao D, Courville A, Bengio Y, et al. An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks. 2015. Arxiv Preprint: https://arxiv.org/pdf/1312.6211.pdf

[27] Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, et al. Overcoming Catastrophic Forgetting in Neural Networks. Proc Natl Acad Sci U S A. 2017;114:3521-3526.

[28] Bengio Y, Courville A, Vincent P. Representation Learning: A Review and New Perspectives. 2014. Arxiv Preprint: https://arxiv.org/pdf/1206.5538.pdf

[29] Caruana R. Multitask Learning. Mach Learn. 1997;28:41-75.

[30] Zhao H, Wang H, Fu Y, Wu F, Li X, et al. Memory Efficient Class-Incremental Learning for Image Classification. 2021. Arxiv Preprint: https://arxiv.org/pdf/2008.01411.pdf

[31] Rusu AA, Rabinowitz NC, Desjardins G, Soyer H, Kirkpatrick J, et al. Progressive Neural Networks. 2022. Arxiv Preprint: https://arxiv.org/pdf/1606.04671.pdf

[32] Fayek HM, Cavedon L, Wu HR. Progressive Learning: A Deep Learning Framework for Continual Learning. Neural Netw. 2020;128:345-357.

[33] Hu Z, Lan Y, Wang L, Xu W, Lim EP, et al. LLM-Adapters: An Adapter Family for Parameter Efficient Fine-Tuning of Large Language Models. 2023. Arxiv Preprint: https://arxiv.org/pdf/2304.01933.pdf

[34] Rohanian O, Jauncey H, Nouriborji M, Kumar V, Gonalves BP, et al. Using Bottleneck Adapters to Identify Cancer in Clinical Notes Under Low-Resource Constraints. 2023. Arxiv Preprint: https://arxiv.org/pdf/2210.09440.pdf

[35] Pfeiffer J, Rücklé A, Poth C, Kamath A, Vulić I, et al. AdapterHub: A framework for adapting transformers. In Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP 2020): systems demonstrations. Online: Association for Computational Linguistics 2020:46-54.

[36] Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. AdapterDrop: On the Efficiency of Adapters in Transformers. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 2021.:7930–7946.

[37] He K, Zhang X, Ren S, Sun J. Identity mappings in deep residual networks. InComputer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. Springer International Publishing.2016:630-645.

[38] Rebuffi SA, Bilen H, Vedaldi A. Learning Multiple Visual Domains With Residual Adapters. Nips. 2017:30.

[39] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, et al. Attention Is All You Need. 2023. Arxiv Preprint: https://arxiv.org/pdf/1706.03762.pdf

[40] Liu CC, Pfeiffer J, Vulić I, Gurevych I. Improving Generalization of Adapter-Based Crosslingual Transfer With Scheduled Unfreezing. 2023. Arxiv Preprint: https://arxiv.org/pdf/2301.05487.pdf

[41] Mao Y, Mathias L, Hou R, Almahairi A, Ma H, et al. UniPELT: Unipelt: A Unified Framework for Parameter Efficient Language Model Tuning. 2022. Arxiv Preprint: https://arxiv.org/pdf/2110.07577.pdf

[42] Graves A, Fernández S, Gomez F, Schmidhuber J. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data With Recurrent Neural Networks. In: Proceedings of the 23rd international conference on machine learning – ICML 2006. Pittsburgh. ACM Press. 2006:369-376.

[43] Qiu S, Zhu C, Zhou W. Meta Self-Learning for Multi-Source Domain Adaptation: A Benchmark. In: Proceedings of the IEEE/CVF international conference on computer vision. 2021:1592-1601.