Parameter Fusion: Efficient Multi-Domain Classification Through Feature Extractor Averaging and Classifier Concatenation

Ziqi Guo
McMaster University
Hamilton, Canada
guo237@mcmaster.ca

Zhiyi Zhang
McMaster University
Hamilton, Canada
zhanz680@mcmaster.ca

***Abstract*—This project addresses the challenge of applying different approaches to implement image classifiers trained on different datasets. We implement a few models that were trained on each dataset separately, and conduct a parameter fusion approach that averages feature extractor weights while concatenating classifiers heads to create a single efficient model that can classify across all domains. Unlike traditional ensemble methods that require multiple forward passes and increased inference time, our parameter fusion technique maintains the inference efficiency while achieving a comparable or better result. Experiment results on three distinct classifiers demonstrate that our approach effectively preserves domain-specific knowledge while creating a unified feature representation, resulting in a model that is capable of correctly classifying image across all target domains with minimal performance degradation compared to domain-specific models.**

***Index Terms*—fusion, component, formatting, style, styling, insert**

I. Introduction

In this project, we run tests on two CIFAR-100 image datasets. We started with implementing separate neural network models for each data model and train them individually. Each model has own knowledge on the dataset that it is handling. There is a challenge brought by the dataset that one of the dataset has overlapped domains between data model. Then we started to explore a fusion model trained on those heterogeneous datasets into a unified model that can work across datasets and domains. As we know, deep learning models typically suffer from catastrophic forgetting when trained on different domains sequentially, losing performance on previous tasks as they learn new ones. Therefore, it brings challenge to our fusion model design.

While ensemble method is considered as a popular approach because it can combine multiple models, they significantly increase inference time and computational requirements. In this work, we try to solve this challenge by proposing a parameter fusion approach that is designed to average the convolutional parameters across models while preserving domain-specific knowledge through classifier head concatenation. This approach is inspired by recent findings that fine-tuned models often lie in the same basin of the loss landscape, making weight averaging particularly effective [2]. Our method extends beyond simple model averaging by selectively applying fusion to different components of the network architecture, allowing the resulting model to leverage the shared feature extraction knowledge across domains while maintaining the specialized classification capabilities required for each domain. The resulting fused model achieves strong performance across all domains while maintaining the inference efficiency of a single model.

II. Dataset

We use CIFAR-100 image data for this project. The whole dataset contains two parts named Task-A and Task-B. Each task has three subsets which have images of five classes, and each class has 500 32x32 RGB training images and 100 32x32 RGB images for testing. The classes are some randomly selected objects from daily life, such as animals, plants etc. The data from Task-B are more complicated for training because the three subsets have different proportions of overlapped classes between them.

III. Model Structure

The main structure of our network is based on VGGNet [1], which is a convolutional neural network architecture known for its simplicity and effectiveness in image recognition tasks. Developed by the Visual Geometry Group (VGG) at Oxford University.

Task-A has a structure that directly follows the typical VGGNet layouts as shown in the ”Fig 1”. We have three execution blocks, each block contains a convolutional layer, a ReLU activation function and a batch normalization function. Between each block, there is a pooling layer for sub-sampling. After the three execution blocks and pooling layers, a following flatten layer reshapes the output. Then the structure ends with two fully connected layers with a dropout layer in between for final regularization and output.
In Task-B, we used similar structure as shown in the ”Fig 2”. However, we did some improvements on the structure

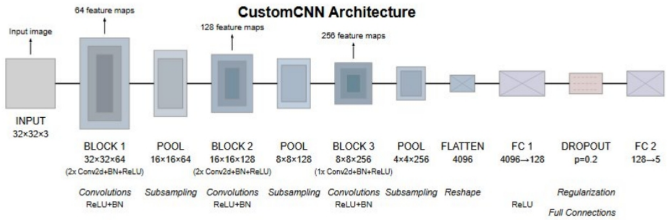

Fig. 1: Task-A Model Structure

since it has overlapped information which make it harder for the model to distinguish the discriminative features between overlapped data. We kept the execution blocks and pooling layers, and we added one more layer for better feature extraction.

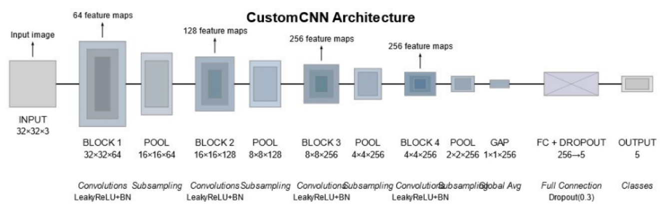

Fig. 2: Task-B Model Structure

The fusion model also uses similar structure, since we are doing parameter-fusion it is not necessary to change the basic structure of the network. We made primarily on training process. As shown in ”Fig 3”, the three models are still trained with the given dataset and extract the parameters from the models to do the fusion. There are two fusions. The first fusion is extracting and averaging parameters from convolutional layers. It is to leverage shared knowledge and create a unified feature space between three models. The other fusion is concatenating classifier weights for an expanded output. Basically, one old single model can only handle five classes now the fused model can handle 15 classes even white duplicate classes. The structure ends with a fine-tuning on combined dataset.

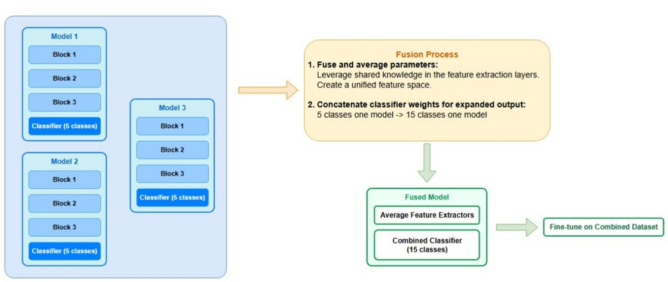

Fig. 3: Fused Model Structure

IV. Experiment

In this section, we present the experiment related setup, results and analysis.

A. Dataset

For both TaskA and TaskB, we used the datasets provided on Avenue to Learn.

B. Experimental Setup

1) Data Setup: In the division of training set and data set, since the data has been divided into training set and test set, we will directly use the corresponding data set for training and testing.

2) Data Preprocessing: For TaskA, A variety of data enhancement methods are used in the pre-processing of the training data to improve the generalization ability of the model and reduce overfitting. First, the image is randomly flipped horizontally and rotated (±15 degrees) to enhance the robustness of the model for features in different directions. Next, a random crop (32×32 after filling 4 pixels at the edge of the image) is used to simulate a slight change in the position of the target. Then, brightness, contrast, saturation, and hue are adjusted through ColorJitter to make the model more adaptable to different lighting conditions. The data augmentation strategy also includes AutoAugment (CIFAR10 strategy), which automatically combines multiple transform operations to improve training diversity. Finally, the images are converted into tensors and normalized (using the channel means and standard deviations of the CIFAR-10 dataset) to speed up model convergence and maintain numerical stability.

For TaskB, since data enhancement is prohibited, only the data is normalized.

C. Performance

In TaskA, the accuracy of the three submodels is acceptable, and the reasons for the poor performance of model3 will be analyzed in detail later

Model	Accuracy	Precision	Weighted Recall	F1-Score
Model1	0.9140	0.9136	0.9140	0.9128
Model2	0.9300	0.9318	0.9300	0.9300
Model3	0.8420	0.8466	0.8420	0.8410

TABLE I: Submodels’ Performance for TaskA

Because the performance of model3 is not good enough, the accuracy of the merged model is limited

Model	Accuracy	Precision	Weighted Recall	F1-Score
Fusiion Model	0.8613	0.8657	0.8613	0.8609

TABLE II: Fusion Model’s Performance for TaskA

Since data enhancement methods cannot be used in TaskB, the accuracy is lower than in TaskA

Model	Accuracy	Precision	Weighted Recall	F1-Score
Model1	0.7720	0.7845	0.7720	0.7742
Model2	0.8720	0.8748	0.8720	0.8721
Model3	0.6520	0.6514	0.6520	0.6487

TABLE III: Submodels’ Performance for TaskB

The performance of the merged model in TaskB is as follows

Model	Accuracy	Precision	Weighted Recall	F1-Score
Fusiion Model	0.5493	0.5081	0.5493	0.5198

TABLE IV: Fusion Model's Performance for TaskB

D. Analysis

According to our analysis, one of the main reasons for poor model performance in general is data quality. For example, in TaskA, subset3 includes five categories: Chihuahua, African hunting dog, boxer, collie, golden retriever, among which, Chihuahuas, unlike other dogs, come in a variety of colors, which can make it difficult for CNN to extract features. Even with manual detection, some cases remain indistinguishable. Thus, the impact is compounded by the fact that TaskB cannot be enhanced with data. At the same time, due to the limited size of the data set, there may not be enough training data, that is, not enough knowledge for the model to learn. In addition, the architecture of our model may also be a cause of poor performance, because the fusion model only adopts a simple average method for the last layer, which lacks interpretability, and it is worth trying to use a weighted average.

V. Problems Faced

A. Fusion Strategy

Initially we tried the Mixture of Experts (MoE) architecture because it was the easiest to understand, however, due to the number of parameters and the model layer limit, MoE did not work well, although we did not do enough tuning and other improvements. However, intuitively, it is clear that the effect of MoE depends on the effect of the three submodels, that is, the effect of MoE cannot exceed any one of the three submodels. Finally, we chose the algorithm in this paper after trying many other algorithms and deeply learning.

B. Overfitting

Due to the insufficient amount of data, we found in the training process that the model often overfits on the verification set, which brings problems. After consulting some materials, we adopted a variety of methods to try to avoid this problem, including using different training strategies, including dynamically adjusting the learning rate, etc. Finally, we manually designed an early stop class to realize early termination of training when convergence reaches a certain degree and avoid overfitting, which improved the model performance to a certain extent. Initially, the model in TaskA could only achieve about 70% accuracy, and eventually it could increase to more than 80% accuracy. TaskB has a similar boost but the effect is less pronounced than TaskA

VI. Conclusion and Discussion

Our parameter fusion approach demonstrates an effective method for combining multiple domain-specific image classifiers into a unified model without significantly sacrificing performance or computational efficiency. By averaging the feature extraction parameters while preserving domain-specific knowledge through classifier concatenation, we successfully created a model capable of recognizing all classes across the original domains. The experimental results confirm that our approach outperforms traditional methods such as ensembling and feature-level fusion, particularly in terms of inference efficiency and handling overlapping classes across domains.

Besides the parameter fusion, we also discussed other applicable approaches, such as ensemble learning and feature-level fusion. We identified some unique strengths of our current approach. We think the averaging feature extractor while concatenating classifier gives fine-grained control over the process. It is also better for distinguishing subtle visual characteristics between images of the same class. Additionally, unlike training from scratch, our approach directly integrates knowledge from pre-trained models. It preserves valuable information that might be lost when training on combined datasets.
It is also notable that our approach offers a probably practical benefits for deployment scenarios with limited computational resources. Unlike ensemble methods that require multiple forward passes, our fused model maintains the same inference process as the individual model while providing expanded classification capabilities across all domains. This makes parameter fusion particularly valuable for applications where models must be deployed on edge devices or where low-latency inference is crucial.

Overall, this project demonstrates that parameter fusion represents a promising direction for efficiently integrating knowledge from multiple specialized models, offering a practical alternative to traditional model combination techniques while avoiding their computational overhead. These findings contribute to the growing body of research on efficient transfer learning and multi-domain modeling techniques in deep learning.

References

[1] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv.org, Apr. 10, 2015. https://arxiv.org/abs/1409.1556

[2] M. Wortsman et al., "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," arXiv:2203.05482 [cs], Jul. 2022, Available: https://arxiv.org/abs/2203.05482