



Universiteit
Leiden
The Netherlands

Exploring deep learning for data-efficient and generalizable visual recognition

He, K.

Citation

He, K. (2026, September 22). *Exploring deep learning for data-efficient and generalizable visual recognition*. Retrieved from <https://hdl.handle.net/1887/4310276>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/4310276>

Note: To cite this publication please use the final published version (if applicable).

Chapter 5

Adaptive Multi-scale Feature Fusion Network for Few-shot Learning

In Chapters 3 and 4, we have proposed several methods to solve the challenges of the few-shot learning (FSL) theme. Unlike the core problem in FSL that is caused by the scarcity of labeled examples and the instability of feature representations under low-shot supervision, we move our research focus to a more general situation where models are supposed to learn with insufficient feature diversity. In this chapter, therefore, we aim to study effectively to enhance feature richness and discriminativeness under extreme data scarcity (**RQ 3**).

FSL presents critical limitations due to the scarcity of labeled training data, severely limiting the models' ability to learn discriminative feature representations. Issues such as insufficient feature diversity and scale sensitivity make image classification tasks challenging in FSL. To bridge this gap, we propose a novel adaptive multi-scale feature fusion (AMFF) network for FSL. AMFF includes two innovative modules, a feature enrichment module (FEM) and a feature fusion module (FFM). FEM increases feature richness through a multi-branch convolution structure and expands representation capacity with limited data. Then FFM reorganizes the multi-scale features with different weights, enabling the network to capture multi-scale information more effectively. Our approach systematically addresses the feature paucity problem through two complementary mechanisms: (1) expanding feature diversity via multi-scale feature extraction, and (2) enhancing feature discriminativeness through an adaptive feature fusion strategy. Extensive experiments on popular FSL benchmarks (miniImageNet, CUB) demonstrate our AMFF achieves competitive performances in few-shot classification tasks

This chapter is based on the following publication [35]:

- **He, K.**, Bakker, E. M., and Lew, M. S. "Adaptive Multi-scale Feature Fusion Network for Few-Shot Learning." International Conference on Intelligent Computing, 2025.

5.1 Introduction

Generally, deep learning approaches [285, 286, 287] achieve optimal performances based on a large amount of samples during the training process. However, in real-world scenarios, collecting and labeling large-scale datasets are time-consuming, or even infeasible. Few-shot learning (FSL) [6, 98] has emerged as a crucial part in the field of machine learning [288], solving the challenge of training models with limited labeled data. In FSL, only a few samples from each category are available for training. FSL is also denoted as an N -way K -shot task where N classes and K samples of each class are used to train the model. As a result, one of the challenges in FSL is to obtain a well-generalized model for unseen classes when trained with few samples.

In the constantly evolving field of FSL, feature fusion [289, 290] is a common way and has been widely-used in many few-shot learning models, such as adaptive feature fusion, inter-layer feature fusion and multi-layer feature fusion. By performing adaptive feature fusion, Ding et al. [291] proposed a meta-learning method which merges different features of each stage of the embedding model, and utilizes a particle swarm algorithm to optimize the feature fusion weights. Through the inter-layer fusion, Wu and Zhao [292] proposed a hierarchical metric FSL model, which constructs an intra-layer channel feature fusion method based on the network hierarchy and transfers the intra-layer fused features from the higher layer to the lower layer. Moreover, high-quality features can be extracted through this model driven by data and knowledge. In the multi-layer feature fusion model, Qin et al. [293] introduced multi-level feature representations when extracting features, and then exploit the fused multi-layer features as a subsequent metric to train the FSL model. Similarly, Gomes et al. [294] also introduced a multi-layer feature extractor and a similarity measurement approach for FSL, which effectively improves image pattern recognition performance through multi-layer feature fusion.

In this chapter, we propose a novel adaptive multiscale feature fusion (AMFF) approach for FSL. The overview of AMFF is presented in Figure 5.1. To address the insufficient feature problem, we first design a multi-branch convolution structure to extract multi-scale features, which is named as feature enrichment module (FEM). FEM enables the model to obtain more features from the very limited samples, effectively enriching the feature diversity. The model’s local perception capability can also be enhanced through these multi-scale features.

Furthermore, to better utilize the multi-scale features extracted by FEM and improve the model performance, we introduce a feature fusion module (FFM), which uses an adaptive strategy to merge these features. On one hand, FFM reorganizes the multi-scale features with different weights based on the adaptive strategy, aiming at obtaining discriminative features; on the other hand, the model’s performance on image classification tasks can be improved through these merged features.

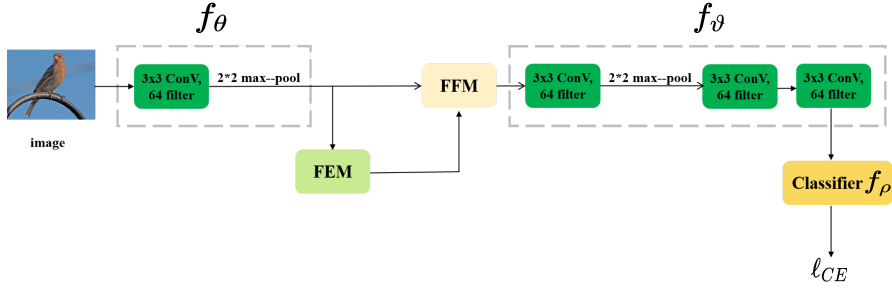


Figure 5.1: Overall training process of AMFF, where ℓ_{CE} denotes the standard cross-entropy loss, FEM is the feature enrichment module and FFM is the feature fusion module.

The main contributions of this work are summarized as follows: 1) We introduce a new multi-branch convolutional network to extract multi-scale features and to solve the insufficient features problem. 2) We propose an adaptive fusion strategy to merge multi-scale features and enable the obtained features to be more discriminative. 3) the experiments on several popular benchmark datasets demonstrate that our approach achieves new state-of-the-art performances in all the 5-way 1-shot and 5-way 5-shot tasks.

5.2 Related Work

Few-shot learning (FSL): FSL aims at classifying unseen tasks effectively through training with a few samples. Recently, a surge of FSL approaches have been proposed for image classification tasks. One popular branch of FSL is metric-based method [23, 295], which performs classification by measuring the metric (similarity or distance) between two samples. In 2015, Koch et al. [141] proposed a Siamese network to learn the similarity between images. Snell et al. [1] classified query samples based on the Euclidean distance to each class prototype, which is calculated by the mean value of samples from the same category. Prior works proved that a good embedding network is essential for FSL. Zhong et al. [296] introduced a novel generative model, which includes encoder, decoder, and discriminator, improving the model’s performance on few-shot Citrus aurantium L. diseases diagnosis. Lim et al. [284] explored knowledge transfer based on a pre-trained large scale model and then fine-tuned the model for the target few-shot task. In order to enhance the embedding representation, Lim et al. integrated FSL with self-supervised tasks.

Feature Fusion: In order to obtain a better feature representation and further improve the model’s performance, feature fusion is widely-used in the field of machine learning. By merging features from multiple different sources, feature fusion helps the model better understand data. Qiu et al. [297] proposed a traditional machine learning method based on multi-feature extraction, significantly improving

the classification accuracy. To alleviate the negative influence caused by inaccurate dependency tree information, Zhou et al. [298] applied a two-layer dynamic fusion mechanism, which adaptively adjusts the fusion weights. With the goal of enhancing the data and supplying the new category samples, Hua and Liu [299] fused multi-scale features extracted by a FPN network. Yu et al. [300] designed a deep multi-scale feature fusion module, interacting with feature information at different scales, which obtains good robustness and generalization ability.

5.3 Adaptive multi-scale feature fusion

In this section, we first make a brief introduction to common notions in FSL. Then we analyze and introduce a feature enrichment module (FEM) and a feature fusion module (FFM) in detail.

5.3.1 Problem formulation and notions

Firstly, we formulate the generalized FSL problem. We use y_{base} and y_{novel} to denote the labels of base and novel classes, respectively. Specifically, $y_{base} = \{1, \dots, C_s\}$ and $y_{novel} = \{C_s + 1, \dots, C_s + C_u\}$, where C_s and C_u are the number of base and novel classes, respectively. In FSL, samples from the base classes are used for training and those from the novel classes are used for evaluation. Besides, y_{base} and y_{novel} are disjoint, which means $y_{base} \cap y_{novel} = \emptyset$. During the evaluation process, samples from y_{novel} are split into two sets, support set and query set, where support set helps the learned model quickly adapt to new tasks and query set is used to evaluate the model's generalization ability. FSL is also named as N -way K -shot problem when the support set contains N classes and each class includes K samples.

5.3.2 Feature enrich module

The goal of feature enrich module (FEM) is to extract multi-scale features for few-shot tasks. In this work, we first introduce a feature extractor which consists of a 4-layer convolutional neural network as the backbone. We use f_θ and f_ϑ to denote the first layer and the subsequent three convolutional layers of the backbone, respectively. For any training image (x, y) from the base set D_{base} , its corresponding shallow features u_s can be obtained through f_θ , which is formulated as:

$$u_s = f_\theta(x). \quad (5.1)$$

Then, in order to obtain multi-scale features, we input the shallow features into FEM g_ϕ , where g_ϕ contains four branches with different kernel sizes. Both standard and depthwise convolutions are adopted in FEM. After that, we concat the multi-scale features which output from the four branches. The detailed diagram of FEM

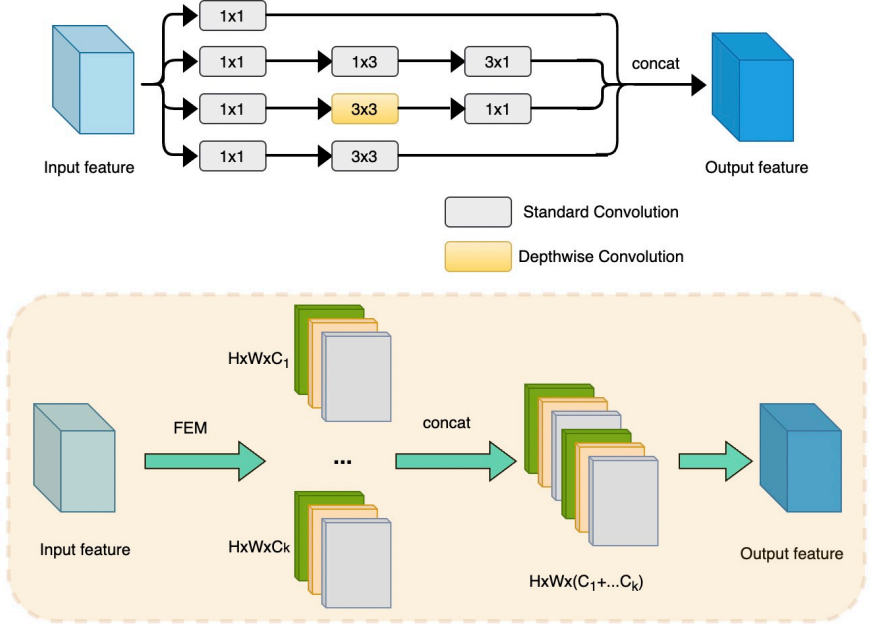


Figure 5.2: Structure of FEM.

is shown in Figure 5.2. Hence, the enriched features can be defined as:

$$u_e = g_\phi(u_s), \text{ where } u_s = f_\theta(x). \quad (5.2)$$

5.3.3 Feature fusion module

To extract discriminative features and improve the model's generalization ability for few-shot tasks, we design a novel feature fusion module (FFM) which consists of a feature interaction component (FIC) h_φ and a feature fusion component (FFC) h_ψ . Here, h_φ is implemented as a 1×1 convolution parameterized by φ , which captures dependencies among multi-scale features, while h_ψ is implemented as a 3×3 convolution parameterized by ψ , which integrates these interacted features to generate the final fused representation. The structure of FFM is shown in Figure 5.3. In order to enable the model to effectively capture multi-scale feature information, we first conduct a squeeze-and-excitation operation on the shallow features and then transmit them to FIC to improve the feature representation ability. The output features from FIC can be calculated as:

$$u_c = h_\varphi(u_s) = \text{Conv}_{1 \times 1}(u_s; \varphi). \quad (5.3)$$

After that, the updated features u_c and enriched features u_e are fused through FFC.

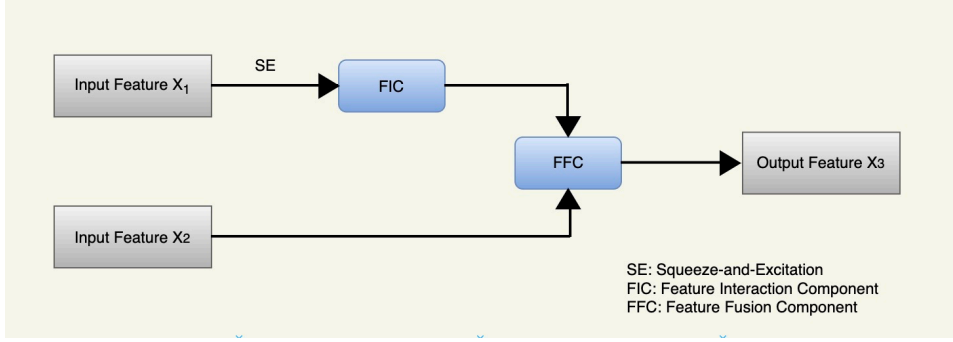


Figure 5.3: Structure of FFM.

Therefore, the fused features u_f can be defined as:

$$u_f = h_\psi(u_c, u_e) = \text{Conv}_{3 \times 1}(u_c + u_e; \psi). \quad (5.4)$$

Next, the fused features u_f are passed to the subsequent three layers, so the final deep features u_d can be calculated as:

$$u_d = f_\vartheta(u_f). \quad (5.5)$$

In the final step, we use a classifier f_ρ to perform classification tasks based on the deep features u_d , so the prediction y' can be calculated as:

$$y' = f_\rho(u_d). \quad (5.6)$$

As we formulate cross-entropy loss as the loss function, the optimization problem can be formulated as:

$$\begin{aligned} \rho, \theta, \vartheta, \phi, \varphi, \psi = \arg \min_{\rho, \theta, \vartheta, \phi, \varphi, \psi} \mathbb{E}_{(x, y) \sim D^{base}} \\ \ell_{CE}(f_\rho(f_\vartheta(h_\psi(h_\varphi(f_\theta(x)), g_\phi(f_\theta(x))))), y) \end{aligned} \quad (5.7)$$

The whole training procedure of our method AMFF is presented in Algorithm 2.

5.4 Experiments

In this section, we first describe the datasets, implementation details and evaluation protocols of our approach, and then make a comparison with state-of-the-art methods on popular benchmarks. Finally, we present an ablation study.

Algorithm 2: The training procedure of AMFF

Input: Training data D_{base} , training parameters such as epoch, batch size and learning rate.

Output: Backbone f_θ and f_ϑ , feature enrich module (FEM) g_ϕ , feature interaction component (FIC) h_φ and feature fusion component (FFM) h_ψ

for $iter = 1 \rightarrow epoch$ **do**

Sample a batch data from D_{base} ;
 Calculate the shallow features $u_s = f_\theta(x)$;
 Calculate the multi-scale features $u_e = g_\phi(u_s)$;
 Calculate the FIC output features $u_c = h_\varphi(u_s)$;
 Calculate the fused features $u_f = h_\psi(u_e, u_c)$;
 Calculate the deep features $u_d = f_\vartheta(u_f)$;
 Train the classifier f_ρ and calculate the loss ℓ_{CE} ;
 Update the parameters of f_ρ , f_ϑ , h_ψ , h_φ , g_ϕ and f_θ ;

end

Return f_ϑ , h_ψ , h_φ , g_ϕ and f_θ

5.4.1 Datasets, implementation details and evaluation protocol

Datasets. We select two benchmarks that have been widely used in FSL: mini-ImageNet [14] and CUB [113] datasets. Their details are presented in Table 5.1. MiniImageNet comes from the ImageNet dataset, containing 100 classes and each class consists of 600 images. MiniImageNet is divided into 64 training classes, 16 validation classes and 20 testing classes according to splitting protocol [115]. Images from miniImageNet are resized into $84 \times 84 \times 3$. CUB has a total of 11,788 images from 200 classes and was originally used for fine-grained bird classification tasks. We follow the split protocol proposed in [55] and divide CUB into three parts, 100 classes for training, 50 classes for validation, and 50 classes for testing. The size of the image is uniformly rescaled and set to $84 \times 84 \times 3$.

Table 5.1: Dataset description.

Dataset	miniImageNet			CUB		
	train	val	test	train	val	test
classes	64	16	20	100	50	50
samples	60000			11788		
image size	$84 \times 84 \times 3$			$84 \times 84 \times 3$		

Implementation details and evaluation protocol. To make a fair comparison with recent works, we use Conv4-64, which consists of 4 convolution layers with $64/64/64/64$ filters as our backbone. In each batch, we randomly select 64 samples from the base classes for training. The optimizer for the experiment is SGD with a

5. ADAPTIVE MULTI-SCALE FEATURE FUSION NETWORK FOR FEW-SHOT LEARNING

Table 5.2: Comparison of AMFF (our approach) to prior works on miniImageNet with mean accuracy (%) and 95% confidence interval.

Model	Backbone	miniImageNet 5-way	
		1-shot	5-shot
ProtoNet [1]	Conv4-64	49.42 $\pm$ 0.78	68.20 $\pm$ 0.66
RelationNet [24]	Conv4-64	50.44 $\pm$ 0.82	65.32 $\pm$ 0.70
MAML [100]	Conv4-64	48.07 $\pm$ 1.75	63.15 $\pm$ 0.91
IMP [301]	Conv4-64	49.60 $\pm$ 0.80	68.10 $\pm$ 0.80
Neg-Margin [126]	Conv4-64	52.84 $\pm$ 0.76	70.41 $\pm$ 0.66
MemoryNet [174]	Conv4-64	53.37 $\pm$ 0.48	66.97 $\pm$ 0.35
HFFCR[289]	Conv4-64	51.79 $\pm$ 0.85	65.74 $\pm$ 0.72
FEAT [239]	Conv4-64	55.15 $\pm$ 0.20	71.61 $\pm$ 0.16
MELR [302]	Conv4-64	55.35 $\pm$ 0.43	72.27 $\pm$ 0.35
SSL-ProtoNet [284]	Conv4-64	52.58 $\pm$ 0.45	70.87 $\pm$ 0.36
Match-sum [162]	SF4-64	55.74 $\pm$ 0.65	72.18 $\pm$ 0.70
AMFF	Conv4-64	56.00 $\pm$ 0.80	73.25 $\pm$ 0.60

momentum of 0.9 and a weight decay of $5e-4$. The initial learning rate is 0.05 and decayed with a factor of 0.1.

We train the model 100 and 150 epochs for miniImageNet and CUB datasets, respectively. Both of them are decayed twice during the training. In addition, random cropping, color jittering, and random horizontal flip are used as data augmentation. In this chapter, we take the common 5-way 1-shot and 5-way 5-shot settings to evaluate the performance of our method. In each task, 5 test classes (5-way) are randomly selected, and each class contains 1 or 5 support samples (K -shot) along with 15 query samples. To evaluate the classification performance, 600 unseen tasks are randomly constructed from the novel classes. Then, an N -way logistic regression classifier is trained for evaluation. Finally, we present the average 5-way K -shot classification accuracies with 95% confidence intervals.

5.4.2 Experimental results and ablative analysis

Experimental results. In this section, we carry out 5-way 1-shot and 5-way 5-shot experiments on miniImageNet and CUB datasets. For a fair comparison, we adopt Conv4-64 as the backbone. The experimental results of our AMFF and other SOTA approaches are reported in Table 5.2 and Table 5.3. Table 5.2 presents the performance comparison on the miniImageNet dataset. It is notable that our AMFF obtains 56.00% and 73.25% accuracy on 5-way 1-shot and 5-way 5-shot tasks, respectively. Compared to a deeper backbone SF4-64, our method AMFF outperforms with 0.26% and 1.07% improvements in 5-way 1-shot and 5-way 5-shot scenarios, respectively. Table 5.3 illustrates similar comparisons on CUB datasets. From Table 5.3, we can find that our method AMFF outperforms other methods

Table 5.3: Comparison of AMFF (our approach) to prior works on CUB dataset, with mean accuracy (%) and 95% confidence.

Model	Backbone	CUB 5-way	
		1-shot	5-shot
MatchingNet [14]	Conv4-64	61.16 $\pm$ 0.89	72.86 $\pm$ 0.70
ProtoNet [1]	Conv4-64	64.42 $\pm$ 0.48	81.82 $\pm$ 0.35
Hyperbolic [303]	Conv4-64	64.02 $\pm$ 0.24	82.53 $\pm$ 0.14
MAML [100]	Conv4-64	55.92 $\pm$ 0.95	72.09 $\pm$ 0.76
RelationNet[24]	Conv4-64	62.45 $\pm$ 0.98	76.11 $\pm$ 0.69
FEAT [239]	Conv4-64	68.87 $\pm$ 0.22	82.90 $\pm$ 0.15
DeepEMD [37]	Conv4-64	64.00	81.10
Match-sum [162]	SF4-64	67.35 $\pm$ 0.93	83.82 $\pm$ 0.61
AMFF	Conv4-64	69.75 $\pm$ 0.86	84.50 $\pm$ 0.57

in comparison with the highest accuracy of 69.75% and 84.50% in 1-shot and 5-shot scenarios, respectively. In particular, AMFF achieves improvements over the best-reported numbers by 0.88% under 5-way 1-shot and 0.68% under 5-way 5-shot settings.

Ablative analysis. To study the impact of FEM and FFM, we evaluate our approach with and without these modules on miniImageNet dataset. We use the original Conv4-64 as the baseline. The ablation results on miniImageNet are presented in Table 5.4. From the results, we can infer the importance of each module in our approach. From Table 5.4, we can find that FEM obtains a gain of 2.31% and 2.56% improvements compared to the baseline on miniImageNet 5-way 1-shot and 5-shot tasks, respectively. The significant improvements effectively demonstrate the contribution of FEM. FFM also helps achieve further 1.3% and 0.55% gains in 5-way 1-shot and 5-way 5-shot tasks, respectively. These results confirm the effectiveness of FFM.

Table 5.4: FSL results on miniImageNet with different models.

Model	miniImageNet 5-way	
	1-shot	5-shot
Baseline	52.39 $\pm$ 0.80	70.14 $\pm$ 0.64
Baseline+FEM	54.70 $\pm$ 0.78	72.70 $\pm$ 0.63
Baseline+FEM+FFM	56.00 $\pm$ 0.80	73.25 $\pm$ 0.60

5.5 Chapter Conclusions

In this work, we propose a novel adaptive multi-scale feature fusion network, which leverages multi-scale features and feature fusion to enhance the performance in few-shot image classification problems. First, to solve the insufficient features caused by

5. ADAPTIVE MULTI-SCALE FEATURE FUSION NETWORK FOR FEW-SHOT LEARNING

limited samples during the training process, we introduce a feature enrichment module, which consists of 4 branches with different kernel size. We exploit the feature enrichment module to generate multi-scale features and enrich feature representations. Then, to further obtain discriminative features and improve the model’s generalization ability, we design a feature fusion module to merge these multi-scale features. The feature fusion module adaptively reorganizes these multi-scale features and effectively improves the model’s robustness. Extensive experimental results demonstrate the importance of the two modules in our approach and we obtain better performance than other SOTA approaches on popular few-shot learning benchmark datasets. Since the proposed method is only used in image classification tasks in this chapter, we will pay attention to other fields in the future, such as object detection and semantic segmentation tasks.