electronics

MDPI

Article

Channel-Wise Attention-Enhanced Feature Mutual Reconstruction for Few-Shot Fine-Grained Image Classification

Qianying Ou * and Jinmiao Zou

School of Information Science and Technology, Fudan University, Shanghai 200433, China; 20210720278@fudan.edu.cn

* Correspondence: 20210720264@fudan.edu.cn

Abstract: Fine-grained image classification is faced with the challenge of significant intra-class differences and subtle similarities between classes, with a limited number of labelled data. Previous few-shot learning approaches, however, often fail to recognize these discriminative details, such as a bird's eyes and beak. In this paper, we proposed a channel-wise attention-enhanced feature mutual reconstruction mechanism that helps to alleviate these problems for fine-grained image classification. This mechanism first employed a channel-wise attention module (CAM) to learn the channel weights for both the support and query features. We utilized channel-wise self-attention to assign greater importance to object-relevant channels. This helps the model to focus on subtle yet discriminative details, which is essential to the classification process. Then, we introduce a feature mutual reconstruction module (FMRM) to reconstruct features. The support features are reconstructed by a support-weight-enhanced feature map to reduce the intra-class variations, and query features are reconstructed by a query-weight-enhanced feature map to increase inter-class variations. The results of classification depend on the similarity between reconstructed features and enhanced features. We evaluated the performance based on four fine-grained image datasets when Conv-4 and Resnet-12 were used. The experimental results showed that our method outperforms previous few-shot fine-grained classification methods. This proves that our method can improve fine-grained image classification performance and simultaneously balance both the inter-class and intra-class variations.

Keywords: few-shot learning; fine-grained image classification; channel-wise attention; feature reconstruction

Academic Editors: Martin Černý, Antonio G. Ravelo-Garcia, Morgado Dias and Ankit Gupta

Received: 18 December 2024
Revised: 10 January 2025
Accepted: 17 January 2025
Published: 19 January 2025

Citation: Ou, Q.; Zou, J. Channel-Wise Attention-Enhanced Feature Mutual Reconstruction for Few-Shot Fine-Grained Image Classification. *Electronics* **2025**, *14*, 377. https://doi.org/10.3390/electronics14020377

1. Introduction

With the rapid development of neural networks, remarkable progress has been made in image processing, such as classification [1,2], object detection [3], and semantic segmentation [4]. This improvement heavily relies on large-scale model training on numerous labelled images. However, data annotation is both costly and time-consuming, resulting in a limited number of labelled data. This issue leads to the overfitting or underfitting of the model, which may degrade the performance [5]. To address this challenge, the computer vision community has proposed few-shot learning methods. These methods mimic human reasoning and quickly acquire new knowledge with only a few examples. Specifically, few-shot learning adopts an episodic learning strategy. In every episode, the model is trained through a support set and evaluated through a query set.

Scholars have attempted to apply transfer learning [6] and meta-learning to achieve few-shot learning. Currently, meta-learning-based few-shot image classification primarily relies on metric learning. The main idea is to compute the distance between the query

feature and the support feature using predefined mathematical metrics or pre-trained classifiers. In other words, the classification results depend on the distance between the query set and the descriptors [7] or points of the support sets in the latent space. The typical ProtoNet [8] calculates the prototypes of support sets according to their Euclidean distances, believing that the prototypes capture the inductive bias of support images. The DeepEMD [9] introduced the Earth Mover's Distance (EMD) and compared the distance between feature representations. This approach divides images into small patches and calculates the best match to determine the distance. Additionally, DeepDBC [10] employs the distance through Brownian Distance Covariance, which quantifies the difference between the joint characteristic distributions of input images. Instead of the method mentioned above using predefined mathematical metrics, the Relation Network [11] trains a learnable classifier to compare the feature vector of input images for classification.

Applying few-shot learning to fine-grained image classification effectively overcomes the challenge of limited labelled data and achieves positive results [12]. However, the challenge in fine-grained image classification extends beyond the issue of limited labelled data. The difference between fine-grained image classification and general image classification can be seen in Figure 1. General images from different classes exhibit significant differences, so the background has a minimal impact on the classification. However, the significant intra-class differences and subtle inter-class similarities between subcategories severely impact the classification. As shown in Figure 1a, horizontally, images in the first row belong to California gulls. However, due to variations in their backgrounds and postures, they look quite different. In contrast, images in the same column belong to different gulls but share some similarities in background and posture, with slight differences in their wings and beaks. This challenge, unique to fine-grained images, amplifies the difficulties of few-shot classification. Therefore, accurately distinguishing these subtle but critical features has played a significant role in few-shot fine-grained image classification.

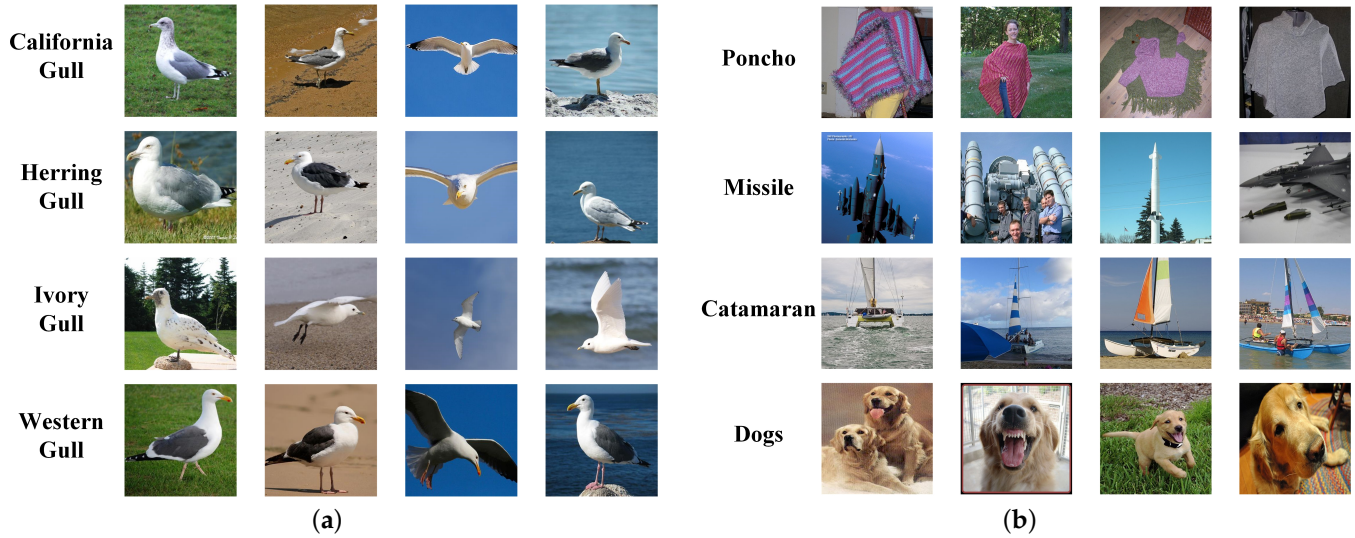

Figure 1. The difference between fine-grained images and general images. Rows represent different species and columns represent different backgrounds. (**a**) Fine-grained image examples; (**b**) general image examples.

Some existing metric-based methods for few-shot learning are directly employed in fine-grained image classification, relying on complex network structures to extract features [13]. FEAT [14] generates distinctive and task-specific features through set-to-set functions and embedding adaptation. CTX [15], which introduces a cross transformer to retrieve features, maps query instances to the supporting latent space and classifies targets

through self-supervised learning. HelixFormer [16] leverages the cross-image semantic relationships between the query and support features within a transformer-based structure. CSCAM [17] introduces a module combining channel attention and spatial attention, and aims to extract discriminative regions through a cross-attention module.

However, there is a problem with this kind of method. After extracting feature maps using various efficient extractors, it is necessary to convert them into single-vector representations for metric functions. The process of converting spatial features into vector representations results in the loss of spatial or positional information, as well as leading to potential overfitting for posture. Taking global pooling as an example, the commonly used softmax classifier averages the input image [8,11], preserving its overall details and location, but leads to overfitting to postures and overlooking potential information. Some attempts address this issue by expanding the receptive field [15], but induce new problem that the model overfits to irrelevant information like the background. Therefore, FRN [18] introduces a novel approach that reconstructs the query feature using the support feature and then compares the reconstructed feature with the query feature for classification. Specifically, it is easier for support images to reconstruct query images belonging to the same class because they share some similar feature mappings. On the contrary, reconstructing query images from different classes will cause substantial errors due to inter-class variations. This is why the category similarity between images can be measured by calculating the reconstruction error. Compared to metric learning, this method preserves spatial details and avoids overfitting to the posture, thus decreasing the influence of inter-class variations. However, while the support–query feature reconstruction encourages the model to learn distinct differences between classes, helping to address inter-class variations, it struggles to capture subtle differences within the same class.

Consequently, we propose a channel-wise attention-enhanced feature mutual reconstruction approach for few-shot fine-grained image classification. We treat feature reconstruction as a ridge regression problem and achieve the best reconstruction using the least square method. Besides the support–query feature reconstruction, we additionally adopt a reverse query–support reconstruction strategy, which aims to reduce the differences between same-class images. This strategy compresses the intra-class differences, encouraging the model to learn more consistent and compact representations for similar instances. The support–query feature reconstruction improves the separability between classes, while the reverse query–support reconstruction focuses on reducing discrepancies within the same class.

This seemingly simple method encourages the model to focus not only on the significant differences between categories (through the support–query feature reconstruction), but also on reducing the gap within the same category (through the query–support feature reconstruction). This mutual learning mechanism enables our model to perform more robustly in fine-grained image classification tasks, especially when the training samples are scarce.

Our channel-wise feature mutual reconstruction contains four modules: (1) a feature extractor, (2) a channel-wise attention module, (3) a feature mutual reconstruction module, and (4) a feature similarity calculation module. In order to weaken the semantic difference caused by background and posture, we propose a channel-wise attention module. This module highlights the key parts of the targets and ensures that the features accurately represent the category information.

In summary, our contributions can be listed as follows:

- We propose a channel-wise attention mechanism. This approach uses channel-wise self-attention to obtain object-specific channel weights. These weights help features to depress the background noise and focus on the salient feature of the target. To

reduce the classification errors caused by background similarities, we minimize the inter-class similarities.

- We introduce a feature mutual reconstruction module. This module reconstructs images using channel-wise enhanced features. This mutual reconstruction ensures a larger intra-class variation and smaller inter-class similarities. Ablation experiments show that mutual reconstruction promotes a stronger interaction between the support and query sets, maximizing their contributions to the classification task.
- To prove the validity of our approach, we conduct several experiments on classic fine-grained image datasets, including CUB-200-2011 [19], Stanford Cars [20], Stanford Dogs [21], and Aircraft [22], and compare them with other advanced methods.

The structure of this paper can be summarized as follows: Section 2 provides an overview of the materials and methods proposed in this paper, and details the application of using a channel-wise attention module, which is complementary to the feature mutual reconstruction. Section 3 presents the experimental results, comparing models across few-shot fine-grained datasets, while examining the impact of each branch on performance. Finally, Section 4 concludes the proposed method, discussing model results, limitations, and future directions for few-shot fine-grained image classification.

2. Materials and Methods

The overall architecture of our approach is illustrated in Figure 2. There is a feature extractor to compute the feature maps for both support and query instances in every episode. We then employ a channel-wise attention module (CAM) to generate attention weights that emphasize the most informative regions of the objects. This attention mechanism works by redistributing weights to object-relevant channels, effectively enhancing the feature maps for subsequent processing. After that, we apply a feature mutual reconstruction module (FMRM) to reconstruct both the support images and query images, leveraging the mutual relationships between the enhanced features. The classification results are determined by the similarity between reconstructed features and channel-wise enhanced features.

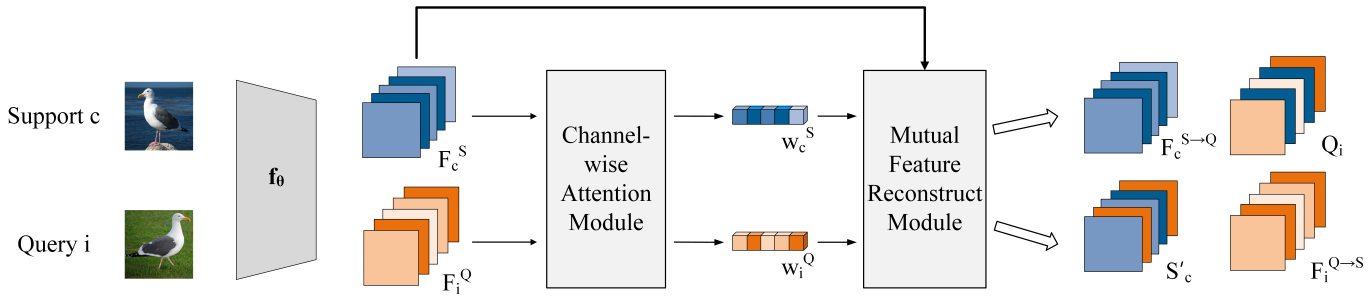

Figure 2. Overview of our approach. The channel-wise feature mutual reconstruction contains four sub-modules. F_c^S and F_i^Q are extracted features. w_c^S represents the attention weight of the cth class support images and w_i^Q represents the attention weight of the ith query instance. $F_c^{S\to Q}$ represents the query feature reconstructed by support feature F_c^S and $F_i^{Q\to S}$ represents the support feature reconstructed by query feature F_i^Q. Q_i represents the query features enhanced by w_c^S, while S'_c represents support features enhanced by w_i^Q. After that, we calculate the similarity between F_c^S and Q_i, as well as F_i^Q and S'_c, to obtain the results.

2.1. Problem Formulation

In a standard few-shot classification, we divided the datasets $D = \{(x_i, y_i), y_i \in Y\}$ into three parts, namely the training set $D_{train} = \{(x_i, y_i), y_i \in Y_{train}\}$, the test set $D_{test} = \{(x_i, y_i), y_i \in Y_{test}\}$ and the validation set $D_{val} = \{(x_i, y_i), y_i \in Y_{val}\}$, similar to other traditional model training processes. During training, the model improves its performance

on a C-way K-shot classification. In every training episode, the model is provided with a support set (meta-training set) and a query set (meta-test set), which are divided from the training set D_{train}. Specifically, in every episode, C classes are randomly selected from D_{train}, and for each of these classes, K labelled images are provided as the support set $S = \{(x_j, y_j)\}_{j=1}^{N \times K}$ and M unlabelled images are provided as the query set $Q = \{x_j\}_{j=1}^{N \times K}$. Images in the support set and query set belong to the same class but do not overlap. After data loading, the total number of samples in each episode is $C \times (K + M)$. This setup ensures that the model is trained to recognize new classes with a limited number of data.

2.2. Channel-Wise Attention Module (CAM)

After the feature extractor, we obtain the feature representations of the support and query images:

$$\begin{aligned} F_c^S &= f_\theta(x_c^S) \\ F_i^Q &= f_\theta(x_i^Q) \end{aligned} \tag{1}$$

where the x_c^S represents images of the cth class of support sets and x_i^Q is the ith instance of the query sets. The feature map $F \in \mathbb{R}^{D \times H \times W}$, where D, H, and W denote the number of channels, height, and weight.

Previous work like SeNet [23] has proved that channel attention weights are beneficial for image classification. They reassign weights across different channels, enabling the proposed model to focus on the distinct areas of input features. In fine-grained image classification, this helps to reduce the impact of the background and highlights the fine-grained objects. The channel-wise attention module we proposed is shown in Figure 3. The core idea is to calculate the correlation along the channel dimension of the input features, allowing the model to focus on distinctive regions of the input features. Specifically, we compute the correlation between feature channels and aggregate these correlations.

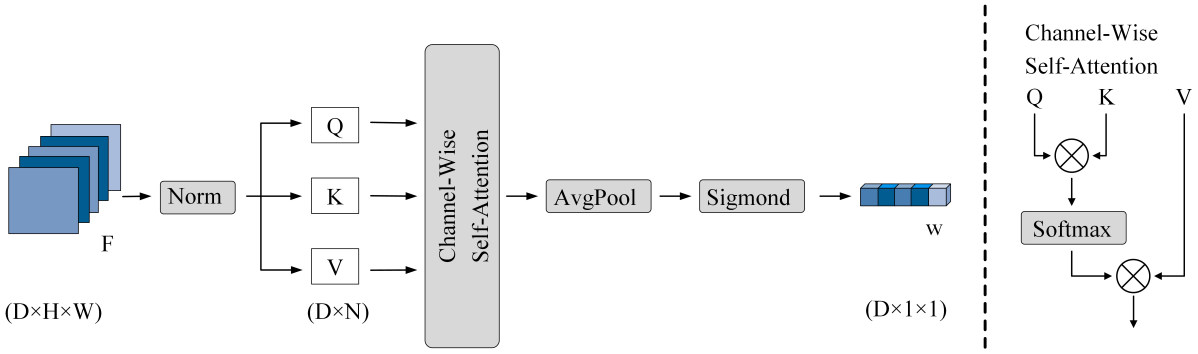

Figure 3. Channel-wise attention module. Q, K, and V are obtained through a 1×1 convolution kernel, representing the query, key, and value of the images in the channel dimension, respectively.

The input features of the channel-wise attention module are denoted as F. We normalize F in the channel dimension, and obtain the Q, K, and V through a 1×1 convolution kernel.

$$\begin{aligned} Q &= Conv1d^Q(F) \\ K &= Conv1d^K(F) \\ V &= Conv1d^V(F) \end{aligned} \tag{2}$$

where $Q, K, V \in \mathbb{R}^{D \times N}$ with $N = H \times W$. The structure of channel-wise self-attention (CSA) is quite similar to the Multi-Head Self-Attention (MHSA) introduced by ViT [24].

$$w = Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{D}})V \tag{3}$$

However, there are still some differences. In MHSA, $Q, K, V \in \mathbb{R}^{N \times D}$ and $N = H \times W$. In our proposed CSA, self-attention focuses on the channel feature, so our $Q, K, V \in \mathbb{R}^{D \times N}$.

2.3. Feature Mutual Reconstruction Module (FMRM)

The reconstruction process is shown in Figure 4, including two branches: the query feature reconstruction branch and the support feature reconstruction branch. QFR is designed to ensure the inter-class differences, while SFR is designed to minimize the inter-class similarity. Before the reconstruction, we enhanced the support feature S_c and query feature Q_i using the support channel-wise weights w_c^s:

$$\begin{aligned} S_c &= (w_c^S)^T \odot F_c^S = [(w_{c,1}^S)^T f_{c,1}^S, (w_{c,2}^S)^T f_{c,2}^S, \ldots, (w_{c,D}^S)^T f_{c,D}^S] \\ Q_i &= (w_c^S)^T \odot F_i^Q = [(w_{i,1}^S)^T f_{i,1}^Q, (w_{i,2}^S)^T f_{i,2}^Q, \ldots, (w_{i,D}^S)^T f_{i,D}^Q] \end{aligned} \tag{4}$$

where $w_{c,j}^S$ represents the scalar value at the jth dimension of w_c^S. $f_{c,d}^S$ is the dth channel of the support feature F_c^S.$f_{i,d}^Q$ is the dth channel of the query feature F_i^Q. $f_c \in \mathbb{R}^{H \times W}$.

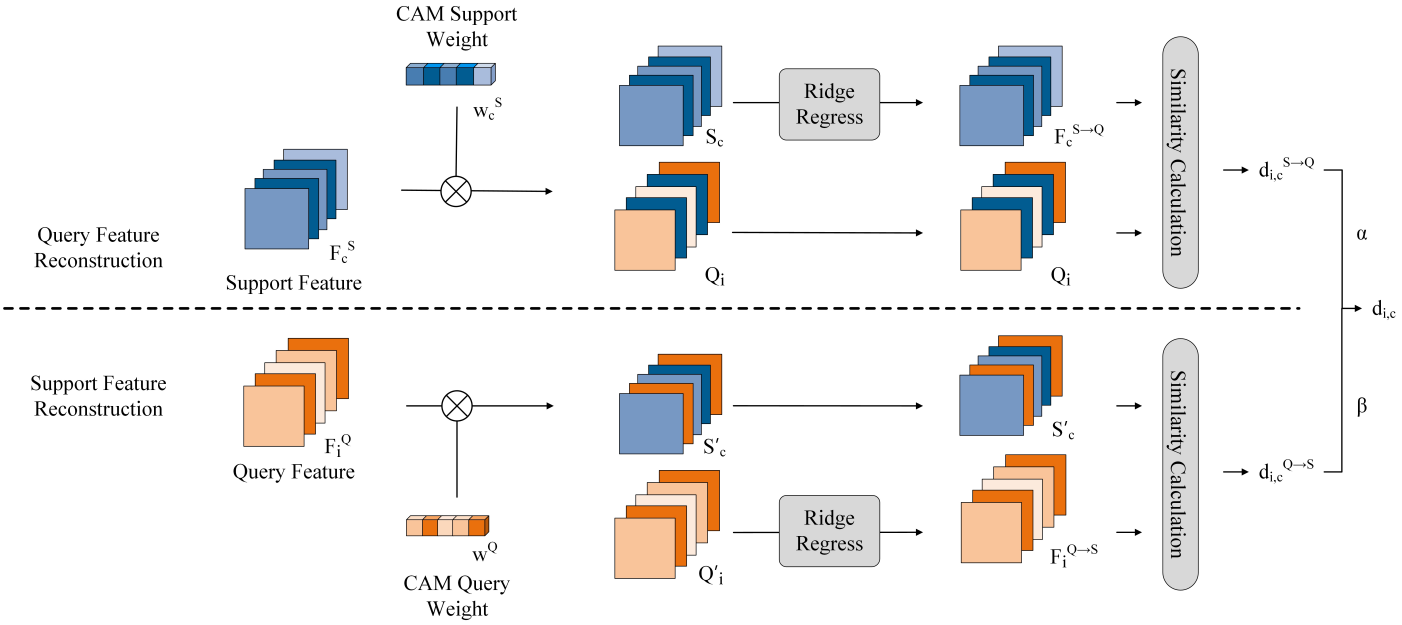

Figure 4. Overview of our feature mutual reconstruction module. S_c and Q_i are the features reassigned by the support attention weight w_c^s. S'_c and Q'_i are the features reassigned by the support attention weight w_i^s. $d_{i.c}^{S\to Q}$ calculates the similarity between $F_c^{S\to Q}$ and Q_i, while $d_{i.c}^{Q\to S}$ calculates the similarity between S'_c and $F_i^{Q\to S}$.

Feature reconstruction aims to figure out a matrix that satisfies $WS_c \approx Q_i$, $W \in \mathbb{R}^{r \times C}$, in which S_c is reshaped to $\mathbb{R}^{kr \times d}$, $r = H \times W$, which represents the feature pools of the cth class. Solving this formulation using the least square method, we can find that

$$\overline{W} = \underset{W}{argmin} \|Q_i - WS_c\|^2 + \lambda \|W\|^2 \tag{5}$$

where $\|.\|$ denotes the Frobenius norm and λ is the ridge regression penalty, which is designed to ensure the optimization is tractable. The reconstruction can be calculated as follows:

$$\begin{aligned} \overline{W} &= Q_i S_c^T (S_c S_c^T + \lambda_1 I)^{-1} \\ F_c^{S\to Q} &= \overline{W} S_c \end{aligned} \tag{6}$$

where $F_c^{S\to Q}$ is the new query feature reconstructed by the support feature and λ_1 is set as $\frac{Kr}{D}$ inspired by [18]. After a query feature reconstruction branch, we obtain $F_c^{S\to Q}$ and an enhanced query feature Q_i that focuses on support classes. Similarly, we enhanced the support feature and query feature by query weights w_i^Q:

$$\begin{aligned} S'_c &= (w_i^Q)^T \odot F_c^S = [(w_{i,1}^Q)^T f_{c,1}^S, (w_{i,2}^Q)^T f_{c,2}^S, \ldots, (w_{i,D}^Q)^T f_{c,D}^S] \\ Q'_i &= (w_i^Q)^T \odot F_i^Q = [(w_{i,1}^Q)^T f_{i,1}^Q, (w_{i,2}^Q)^T f_{i,2}^Q, \ldots, (w_{i,D}^Q)^T f_{i,D}^Q] \end{aligned} \tag{7}$$

We reconstruct a new support feature using Q'_i:

$$\begin{aligned} \overline{W'} &= S'_c Q_c^{'T} (Q'_c Q_c^{'T} + \lambda_2 I)^{-1} \\ F_i^{Q\to S} &= \overline{W'} Q'_c \end{aligned} \tag{8}$$

where λ_2 is set as $\frac{r}{D}$.

2.4. Classifier and Loss

We obtain the reconstructed features $F_c^{S\to Q}$ and $F_i^{Q\to S}$ and the enhanced features Q_i and S'_c after FMRM. We calculate the similarity between the enhanced query feature Q_i and the reconstructed query feature $F_c^{S\to Q}$ as

$$d_{i.c}^{S\to Q} = \|F_c^{S\to Q} - Q_i\|^2 \tag{9}$$

and compute the similarity between the enhanced support feature S'_c and the reconstructed suport feature $F_i^{Q\to S}$ as

$$d_{i.c}^{Q\to S} = \|F_i^{Q\to S} - S'_c\|^2 \tag{10}$$

Through Equations (9) and (10), we measure the reconstruction similarities of QFR and SFR. To measure the total reconstruction similarities of the FMRM, we calculate the total distance vis the weighted summation of the two distances. Thus, the total distance between the ith query instance and the cth class is

$$d_{i,c} = \gamma(\alpha d_{i.c}^{S\to Q} + \beta d_{i.c}^{Q\to S}) \tag{11}$$

Inspired by [15,18], we set γ, α, and β as three learnable parameters, and their initial value is 1.00. α and β are designed to dynamically adjust the importance of each branch. γ is introduced in order to control the peakiness of Equation (11). The possibility that the ith query instance belongs to the cth class is given by

$$P_i^c = \frac{e^{(-d_{i,c})}}{\sum_{i'\in C} e^{(-d'_{i,c})}} \tag{12}$$

During training, we employ a cross-entropy function to calculate the loss of our classification:

$$L_{entropy} = -\frac{1}{M\times C} \sum_{i=1}^{M\times C} \sum_{c=1}^{C} 1(y_i == c) \log(P_i^c)$$

To improve the quality of the reconstructed feature, we additionally introduce a reconstructing loss:

$$L_{recon} = \sum_{i\in C} \sum_{j\in C, j\neq i} ||\tilde{S}_i \tilde{S}_i^T||^2 + \sum_{i\in q} \sum_{j\in q, j\neq i} ||\tilde{Q}_i \tilde{Q}_i^T||^2 \tag{13}$$

where $\tilde{S}$ and $\tilde{Q}$ is row-normalized, and q is the number of query images. This loss ensures the orthogonality between features, so it encourages larger differences between features. This helps the module to reduce the similarity caused by the background. The total loss of the training is

$$L = L_{entropy} + \theta L_{recon} \tag{14}$$

Following [25], we set θ as 0.03.

3. Results

3.1. Datasets

We evaluate the performance of the proposed method based on four classic fine-grained datasets. For each dataset, we divide it into three parts: D_{train}, D_{val}, and D_{test}. The ratio of each part is shown in Table 1, and all images are resized to 84×84.

Table 1. The split of datasets.

Datasets	D_{all}	D_{train}	D_{val}	D_{test}
CUB-200-2011	200	100	50	50
Stanford-Cars	190	136	17	49
Stanford-Dogs	120	70	20	30
Aircraft	100	50	25	25

CUB-200-2011 (CUB) [19] is a widely used fine-grained image classification dataset, consisting of 11,788 images across 200 bird species. Following [10], we crop the images by annotated bounding boxes given by the dataset.

Stanford-Cars (Cars) [20] is a classic fine-grained image classification dataset as well, consisting of 16,185 images across 196 different kinds of cars. Image labels contain information including the brand, model, and year, such as the 2012 Tesla Model S and the 2012 BMW M3coupe.

Stanford-Dogs (Dogs) [21] is a classic fine-grained image classification dataset, consisting of 20,580 images across 120 dog breeds.

Aircraft [22] is a challenging fine-grained image classification dataset, consisting of 10,000 images across 100 aircraft models.

3.2. Implementation Details

3.2.1. Architecture

Following the standard protocols from recent few-shot classification works [5], we conducted our experiments on the backbone Conv-4 [26] and ResNet-12 [27]. Conv-4 consists of four convolutional blocks, where each block consists of a convolution layer with 64 3×3 kernels, followed by a BatchNorm operation, a ReLU activation, and a max-pooling layer with 2×2 pool size. After a Conv-4 backbone, a $3 \times 84 \times 84$ image is transformed into a $64 \times 5 \times 5$ feature map. The ResNet-12 architecture, on the other hand, is made up of four residual blocks. Each of these blocks contains three convolutional layers, followed by a BatchNorm normalization and a Leaky ReLU (with a slope of 0.1). The last convolutional layer in each block uses 2×2 max-pooling. After Resnet-12, we obtain a $640 \times 5 \times 5$ feature map.

3.2.2. Training Details

In our experiments, the models based on Conv-4 are trained for a total of 800 epochs, using Stochastic Gradient Descent (SGD) with Nesterov momentum set to 0.9. We set the initial learning rate as 0.1, and reduce it to 0.01 after 400 epochs. Models are trained on

20-way five-shot episodes and directly tested on five-way one-shot or five-shot episodes. For each training or testing episode, 15 query images are selected.

As for the Resnet-12 models, we train them on three stages with 400 epochs per stage. The initial learning rate is set as 0.1, and decreases by a scale factor of 10 every stage. The SGD optimizer with a Nesterov momentum of 0.9 is also used. To save the memory, we train the Resnet-12 models on 10-way 5-shot episodes, keeping other test setups unchanged.

The weight decay of our model's training is 5×10^{-4}. In addition, following existing methods [5,14,18], standard data augmentation techniques are used to achieve better training stability, including random crop, horizontal flip, and colour jitter. To prevent overfitting, we validate the model every 20 epochs and select the best-performing model based on the validation set.

3.2.3. Evaluation Details

In the N-way K-shot classification task, we examine few-shot classification for 10,000 episodes. In every episode, N classes are randomly selected, and each class contains K support images and 15 query images. The results are reported as the average classification accuracy along with 95% confidence intervals, as in [7,28].

3.3. Comparison of Results

To validate the efficiency of our approach, we conducted experiments on the three fine-grained image datasets mentioned above and the results are listed in Tables 2 and 3. The results marked with * were obtained with the official code provided by the authors, and the original dataset was replaced by the one used in this paper. The results marked with † were obtained from CSCAM [17]. We maintained the same dataset split ratio and used the same BBOX as in the original method. All other experimental settings remained consistent with the official implementation.

Table 2. Five-way few-shot classification performance based on three fine-grained datasets when Conv-4 is used.

Method	CUB		Dogs		Cars	
	1-Shot	**5-Shot**	**1-Shot**	**5-Shot**	**1-Shot**	**5-Shot**
Matching [29]	66.06 ± 0.88	74.57 ± 0.73	46.10 ± 0.86	59.79 ± 0.72	44.73 ± 0.77	64.74 ± 0.72
ProtoNet * [8]	63.84 ± 0.24	84.61 ± 0.15	47.13 ± 0.21	68.48 ± 0.17	48.64 ± 0.20	74.23 ± 0.17
RelationNet [11]	63.94 ± 0.92	77.87 ± 0.64	47.35 ± 0.88	66.20 ± 0.74	46.04 ± 0.91	68.52 ± 0.78
DN4 [7]	57.45 ± 0.89	84.41 ± 0.58	39.08 ± 0.76	69.81 ± 0.69	34.12 ± 0.68	87.47 ± 0.47
DeepEMD [10]	64.08 ± 0.50	80.55 ± 0.71	46.73 ± 0.49	65.74 ± 0.63	61.63 ± 0.27	72.95 ± 0.38
LRPABN [28]	63.63 ± 0.27	76.06 ± 0.58	45.72 ± 0.75	60.94 ± 0.66	60.28 ± 0.76	73.29 ± 0.58
CTX * [15]	71.16 ± 0.21	85.73 ± 0.14	56.18 ± 0.21	71.98 ± 0.16	65.15 ± 0.21	81.25 ± 0.14
MistFSL* [30]	56.45 ± 0.88	74.75 ± 0.75	45.61 ± 0.78	62.22 ± 0.68	44.43 ± 0.79	66.31 ± 0.75
FRN * [18]	74.01 ± 0.21	88.55 ± 0.13	58.42 ± 0.21	77.48 ± 0.15	66.00 ± 0.21	85.96 ± 0.12
BSFA † [31]	68.16 ± 0.52	82.41 ± 0.35	-	-	49.98 ± 0.48	67.52 ± 0.44
AGPF [32]	74.03 ± 0.90	86.54 ± 0.50	60.89 ± 0.89	78.14 ± 0.62	78.14 ± 0.84	87.42 ± 0.57
IDEAL-clean † [33]	69.93 ± 0.89	81.67 ± 0.69	-	-	52.64 ± 0.91	70.28 ± 0.69
CAML [34]	59.71 ± 1.46	73.09 ± 0.73	-	-	-	-
W3SL [35]	71.48	86.74	59.37	78.94	61.13	81.51
Ours	74.37 ± 0.22	89.20 ± 0.12	59.61 ± 0.22	78.56 ± 0.15	67.09 ± 0.22	87.95 ± 0.11

* Results were obtained with the official code provided by the authors, and the original dataset was replaced by the one used in this paper. † Results were obtained from CSCAM.

Table 3. Five-way few-shot classification performance based on three fine-grained datasets when Resnet-12 is used.

Method	CUB		Dogs		Cars	
	1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
ProtoNet * [8]	78.07 ± 0.21	90.54 ± 0.11	72.25 ± 0.22	86.82 ± 0.13	86.11 ± 0.18	95.02 ± 0.08
FEAT [14]	73.27 ± 0.22	85.77 ± 0.14	-	-	-	-
DeepEMD * [10]	74.87 ± 0.24	87.83 ± 0.28	70.03 ± 0.26	84.78 ± 0.18	79.63 ± 0.27	90.95 ± 0.38
VFD [36]	79.12 ± 0.83	91.48 ± 0.39	76.24 ± 0.87	88.00 ± 0.47	-	-
CTX * [15]	77.89 ± 0.20	90.84 ± 0.11	72.97 ± 0.22	85.60 ± 0.13	84.93 ± 0.19	92.63 ± 0.14
DeepDBC * [10]	81.89 ± 0.42	91.84 ± 0.31	72.57 ± 0.32	84.96 ± 0.17	80.93 ± 0.39	92.03 ± 0.14
HelixFormer [16]	81.66 ± 0.30	91.83 ± 0.17	65.92 ± 0.49	80.65 ± 0.36	79.40 ± 0.43	92.26 ± 0.15
FRN * [18]	81.5 ± 10.20	91.77 ± 0.11	76.43 ± 0.21	88.23 ± 0.12	87.95 ± 0.16	95.30 ± 0.08
BSFA [31]	82.27 ± 0.46	90.76 ± 0.26	69.58 ± 0.50	82.59 ± 0.33	88.93 ± 0.38	95.20 ± 0.20
AGPF [32]	78.73 ± 0.84	89.77 ± 0.47	-	-	-	-
TDM + CSCAM [17]	83.34 ± 0.19	92.28 ± 0.18	-	-	86.86 ± 0.17	95.63 ± 0.08
C2-Net [37]	-	-	75.50 ± 0.49	87.65 ± 0.28	88.96 ± 0.37	95.16 ± 0.20
FicNet [38]	80.97 ± 0.57	93.17 ± 0.32	72.41 ± 0.64	85.11 ± 0.37	88.81 ± 0.47	95.36 ± 0.22
W3SL [35]	73.16	89.75	62.94	82.16	64.85	84.25
Ours	83.09 ± 0.19	92.75 ± 0.10	77.21 ± 0.21	88.90 ± 0.12	89.03 ± 0.16	96.09 ± 0.07

* Results were obtained with the official code provided by the authors, and the original dataset was replaced by the one used in this paper.

We tested the five-way one-shot and five-way five-shot classification performance with other advanced methods based on the same backbone Conv-4 and Resnet-12. As shown in Table 2, our method performs best based on all three datasets on the backbone Conv-4 except for the comparison with AGPF. In addition to the results on the Cars dataset for one-shot training, we approached and even surpassed these data. As shown in Table 3, we achieved outstanding performance across nearly all datasets. The exceptional comparison with TDM+CSCAM on the CUB dataset for one-shot training is comparable. We can conclude that our proposed method achieves superior performance regarding the datasets CUB, Cars, and Dogs, regardless of the backbone. As for the performance based on the Aircraft dataset, the results are shown in Table 4. Compared with other advanced experiments, our approach achieves results that are highly competitive, demonstrating comparable performance.

Table 4. Five-way few-shot classification performance based on the Aircraft dataset.

Method	Conv-4		Resnet-12	
	1-Shot	5-Shot	1-Shot	5-Shot
ProtoNet * [8]	52.07 ± 0.21	82.98 ± 0.16	85.67 ± 0.18	91.89 ± 0.11
FRN * [18]	66.68 ± 0.22	84.17 ± 0.13	86.58 ± 0.17	92.28 ± 0.09
HelixFormer [16]	70.37 ± 0.57	79.80 ± 0.42	74.01 ± 0.54	83.11 ± 0.41
IDEAL-clean † [33]	52.26 ± 0.83	80.36 ± 0.69	61.37 ± 0.92	82.51 ± 0.55
BSFA † [31]	61.17 ± 0.49	76.96 ± 0.36	87.85 ± 0.35	94.93 ± 0.14
C2-Net † [37]	-	-	87.98 ± 0.39	93.96 ± 0.20
Ours	67.09 ± 0.20	85.02 ± 0.13	87.31 ± 0.15	93.85 ± 0.10

* Results were obtained with the official code provided by the authors, and the original dataset was replaced by the one used in this paper. † Results were obtained from CSCAM.

This is due to the proposed feature mutual reconstruction module. By capturing the similarities between images from the same class and the gap between different classes,

we reduce the intra-class variance and increase the inter-class variance, thus increase the classification accuracy.

3.4. Ablation Studies

3.4.1. Analysis of Learnable Parameters

We set α and β as learnable parameters to reduce the time and effort for manual tuning. To investigate the effects of different combinations of values on performance, we fixed α and β at various levels and conducted some experiments. These experiments used Resnet-12 as the backbone and CUB as the dataset. The results are shown in Table 5.

Table 5. Five-way few-shot classification with different combinations of α and β based on the CUB dataset when Resnet-12 is used.

α	β	1-Shot	5-Shot
0	1	82.05 ± 0.20	91.08 ± 0.12
0.5	1	82.57 ± 0.20	92.43 ± 0.11
1	1	82.26 ± 0.21	92.07 ± 0.13
1→0.67	1→7.83	83.09 ± 0.19	92.75 ± 0.10
1	0.5	82.01 ± 0.20	91.39 ± 0.11
1	0	81.71 ± 0.20	91.01 ± 0.12

From Table 5, we can notice that when α is fixed at 1, a decrease in β consistently leads to a worse performance. This indicates that a smaller β negatively affects the model. On the other hand, when β is fixed to 1, the model performance reaches its peak at $\alpha = 0.5$. After training, the values of the learnable parameters changed. α converged from 1 to 0.67, and β converged from 1 to 7.83, which aligns with the trend we observed.

3.4.2. The Effectiveness of CAM and FMRM

We analyse each module of the proposed methods by progressively removing components. Without CAM, the model (FMRM) directly uses the output from the feature extractor for reconstruction and calculates the reconstruction error. Without FMRM, after obtaining the channel attention weights, the model (CAM) performs classification by calculating the Euclidean distance. The results, including training on three datasets and two backbones, are reported in Table 6. We removed each of the two modules one at a time, and eventually eliminated both modules, which ended up with the baseline.

Table 6. Ablation studies on each module.

Backbone	Method	CUB		Dogs		Cars	
		1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
Conv-4	Baseline	63.84 ± 0.24	84.61 ± 0.15	47.13 ± 0.21	68.48 ± 0.17	48.64 ± 0.20	74.23 ± 0.17
	CAM	66.15 ± 0.23	85.84 ± 0.15	51.27 ± 0.22	70.82 ± 0.17	56.75 ± 0.23	76.96 ± 0.17
	FMRM	73.91 ± 0.21	88.11 ± 0.13	57.98 ± 0.22	76.99 ± 0.16	62.98 ± 0.22	84.58 ± 0.13
	CAM + FMRM	74.37 ± 0.22	89.20 ± 0.12	59.61 ± 0.22	78.56 ± 0.15	67.09 ± 0.22	87.95 ± 0.11
Resnet-12	Baseline	78.07 ± 0.21	90.54 ± 0.11	72.25 ± 0.22	86.82 ± 0.13	86.11 ± 0.18	95.02 ± 0.08
	CAM	80.72 ± 0.20	91.51 ± 0.12	74.59 ± 0.21	87.93 ± 0.12	87.49 ± 0.17	95.78 ± 0.08
	FMRM	82.90 ± 0.19	92.56 ± 0.10	76.95 ± 0.20	89.09 ± 0.16	87.20 ± 0.17	95.91 ± 0.07
	CAM + FMRM	83.09 ± 0.19	92.75 ± 0.10	77.21 ± 0.21	88.90 ± 0.12	89.03 ± 0.16	96.09 ± 0.07

Compared with the baseline, CAM calculates the weights of both support and query images, then computes the Euclidean distance between the weights-enhanced query and support feature for classification. FMRM utilizes two branches for reconstruction and computes the similarity, instead of Euclidean distance. According to Table 6, FMRM plays an effective role in reducing the classification errors caused by misaligned images. In

most cases, the performance declines after removing either component. Therefore, both of the modules we proposed are essential and complementary, working together to enhance overall performance and reduce classification errors.

3.4.3. The Effectiveness of Each Branch in FMRM

To validate the efficiency of mutual reconstruction in FMRM, we conducted experiments on each reconstruction branch. We remove the support image reconstruction in FMRM by setting α as 0 in Equation (11), which is noted as QFR. Similarly, SFR represented the removal of the query image reconstruction by setting β as 0 in Equation (11). Trials on all three datasets and two backbones were conducted, as shown in Table 7.

Table 7. Ablation studies on reconstruction branches of FMRM.

Backbone	Method	CUB		Dogs		Cars	
		1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
Conv-4	Baseline	63.84 ± 0.24	84.61 ± 0.15	47.13 ± 0.21	68.48 ± 0.17	48.64 ± 0.20	74.23 ± 0.17
	SFR	73.38 ± 0.22	88.91 ± 0.12	58.87 ± 0.22	77.98 ± 0.15	66.16 ± 0.22	88.14 ± 0.11
	QFR	72.88 ± 0.24	87.83 ± 0.17	57.91 ± 0.21	77.04 ± 0.16	65.39 ± 0.23	87.82 ± 0.17
	CAM + FMRM	74.37 ± 0.22	89.20 ± 0.12	59.61 ± 0.22	78.56 ± 0.15	67.09 ± 0.22	87.95 ± 0.11
Resnet-12	Baseline	78.07 ± 0.21	90.54 ± 0.11	72.25 ± 0.22	86.82 ± 0.13	86.11 ± 0.18	95.02 ± 0.08
	SFR	82.05 ± 0.20	91.08 ± 0.12	77.52 ± 0.21	88.29 ± 0.12	88.44 ± 0.17	95.65 ± 0.08
	QFR	81.71 ± 0.21	91.01 ± 0.13	75.98 ± 0.22	87.67 ± 0.14	87.88 ± 0.18	94.97 ± 0.09
	CAM + FMRM	83.09 ± 0.19	92.75 ± 0.10	77.21 ± 0.21	88.90 ± 0.12	89.03 ± 0.16	96.09 ± 0.07

According to Table 7, the mutual reconstruction method consistently outperforms the single-branch methods in most cases. This highlights the effectiveness of leveraging both SFR and QFR together, rather than relying on either one of them. The trends observed in these experiments are similar to those found in Table 5. SFR plays a more crucial role in the model's overall performance. The QFR is also indispensable, as removing it leads to a noticeable decline in performance.

3.4.4. The Effectiveness of Shots and Ways

To prove the robustness and stability of our method, we changed the number of ways or shots during inference on the same model to evaluate the effect of the number of ways and shots and compared these results with FRN and ProtoNet under the same settings. The evaluation model was trained for five-way five-shot classification with a Resnet-12 backbone based on CUB.

Figure 5 illustrates how the accuracy of the three models varies with different shot numbers for a five-way classification task. Our model outperformed the others for any number of shots. As the number of shots increases, the performance improves gradually. As the number of shots increases, the accuracy of our model increases by 11.5%, while that of FRN and ProtoNet increases by 12.39% and 14.90%, respectively. Compared to the other models, our model is less sensitive to samples per class than other models.

Figure 6 illustrates how the accuracy of three models varies with different numbers of ways for a five-shot classification task. Our model outperformed the others for any number of ways. As the number of ways increases, the performance decreases gradually. With various numbers of ways from 5 to 30, our model's accuracy decreases from 92.75% to 75.36%, while that of FRN decreases from 91.77% to 71.98%, and that of ProtoNet decreases from 90.54% to 68.87%. This indicates the robustness of our method against the change in the number of classes.

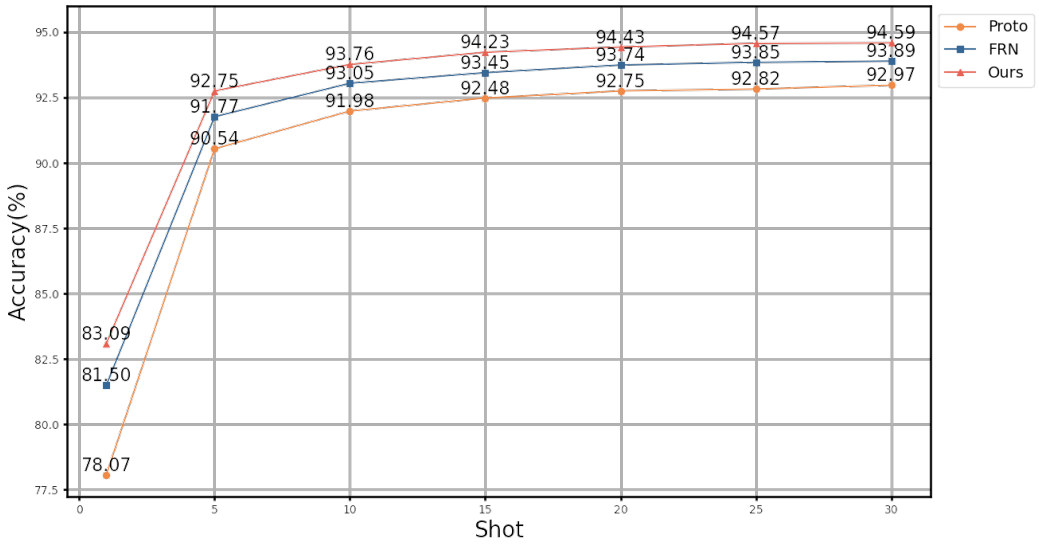

Figure 5. Five-way K-shot classification performance. We obtained the data by testing the same model on a 5-way K-shot classification task. This chosen model was trained on 5-way 5-shot classification with a Resnet-12 backbone on CUB.

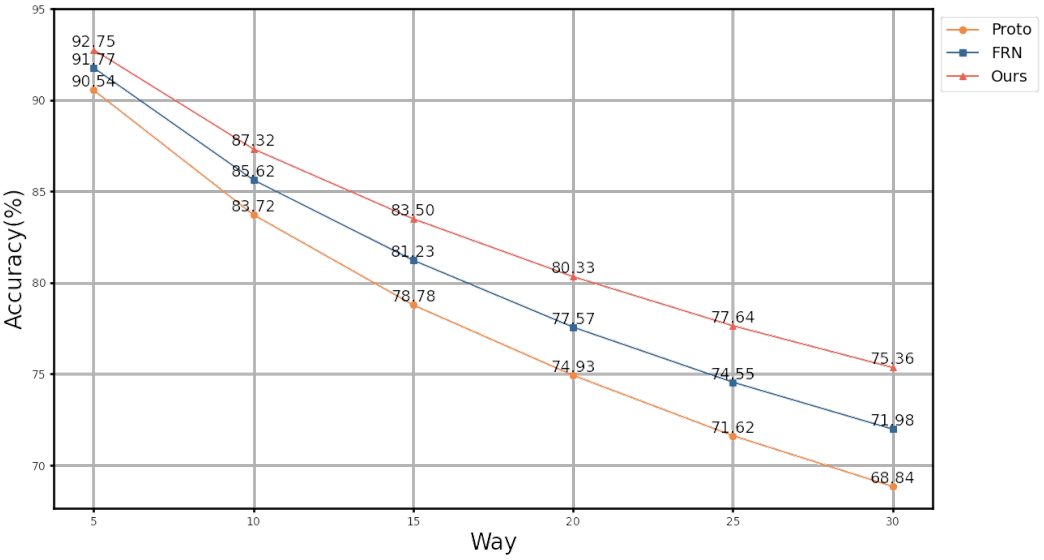

Figure 6. C-way 5-shot classification performance. We obtained the data by testing the same model on a C-way 5-shot classification task. This chosen model was trained on 5-way 5-shot classification with a Resnet-12 backbone on CUB.

3.5. Visualization Analysis

3.5.1. Reconstruction Visualization

We trained an inverse Resnet-12 as a generator to visualize the reconstructed features and enhanced features. This inverse Resnet-12 took the feature representations captured by feature exactor as the input and took the original image as the output to recover these features. The generator is trained through L1 loss and optimized with the Adam optimizer, starting with a learning rate of 0.01 and a batch size of 200. We trained the generator for 500 epochs, and reduced the learning rate by a factor of 4 every 100 epochs.

The recovery images are shown in Figure 7. The block on the far left displays the original support images in a five-way five-shot classification task, while the rightmost block displays the original query image. The query image is repeated five times for easier comparison of the reconstructed and enhanced images across different categories. The second left block is the support features S'_c enhanced by the channel attention weights from

query images w^Q. The third left block is the reconstructed support features $F^{Q \to S}$. The second right block is the query features Q_i enhanced by the channel attention weights from support images w^S. The third left block is the reconstructed query features $F^{S \to Q}$.

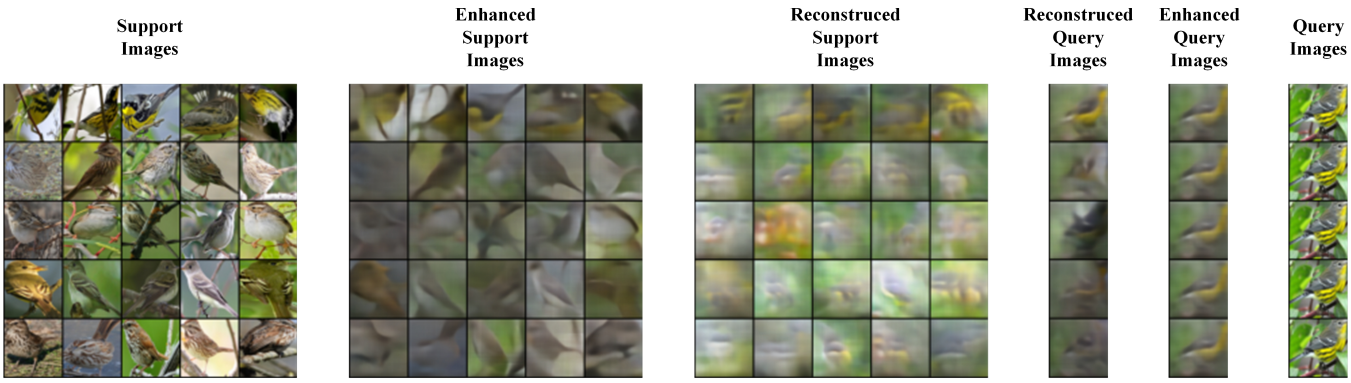

Figure 7. Visualization of image features recovered by inverse Resnet-12 in the CUB datasets. From left to right, the images represent the original support features, enhanced support features, reconstructed support features, reconstructed query features, enhanced query features, and the original query features.

From the second left column, it can be seen that enhanced support features focus more on the similarity between the support and query images, like the colorful pattern. For the support instances from different categories, the quality of the enhanced images from support images with different color patterns is not as good as those with the same color pattern. The reconstructed support images, in the third left column, exhibit the same pattern as the query image and the same pose as the support images. This shows that our SFR has indeed played a positive role in suppressing intra-class variations. Calculating the similarity between these two columns maximizes the differences introduced by the pattern and minimizes the errors caused by the pose.

The second and third right columns provide the enhanced query features and the reconstructed query feature. The reconstructed query features focus on the posture of the query image and patterns of the support image. The quality discrepancy of reconstructed images from different ways is substantial, which indicates that QFR maximizes the difference between classes. However, the query image features, enhanced by the feature weights of different categories, appear to be no different. This may be the reason why the support reconstruction branch outperformed the the query reconstruction branch.

3.5.2. Feature Visualization

We visualized discriminative regions of ProtoNet and our method using Grad-CAM [39], as shown in Figure 8. These visualized data are captured from the ResNet12 5-shot model based on the CUB dataset. After learning from support images, we evaluated the model's performance based on the shown query images. According to the visualization, we found that ProtoNet tends to focus on the whole object, while our model localizes the most delicate discriminative regions.

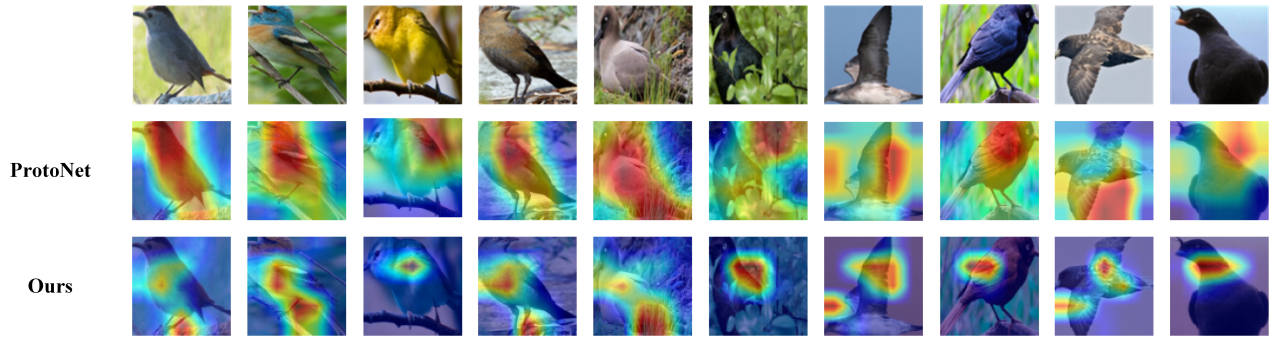

Figure 8. Visualization of the discriminative regions by GradCAM in the CUB datasets. From top to bottom, the images represent the original CUB image, ProtoNet's visualization, and our visualization.

4. Conclusions

In this paper, we introduced a channel-wise attention-enhanced feature mutual reconstruction mechanism, a reconstruction-based method for few-shot fine-grained image classification designed to alleviate significant intra-class differences and subtle inter-class similarities. We utilized a channel-wise attention module (CAM) to reassign the channel weights of support and query weights. This enabled the model to focus on the distinguishing parts of the targets. Then we reconstructed support and query features with these attention-enhanced features. Support features were reconstructed using a support-weight-reassigned feature map to minimize intra-class variation, while query features were reconstructed with a query-weight-reassigned feature map to maximize inter-class variation. We obtained the classification results based on the similarity between the reconstructed features and attention-enhanced features.

The results based on four widely-used fine-grained benchmarks indicate that our classification method is superior to the previous method and support the robustness of our model. Additionally, the ablation studies confirm that our CAM and FMRM play an essential and complementary role in enhancing overall performance and reducing classification errors. Each branch of reconstruction impacts differently on the module, and they are all indispensable. From the visualization of our model, we can conclude that SFR reduces the difference from the same classes and QFR helps to learn the differences.

Despite the positive results achieved by our model, it suffers from a few limitations. As shown in Figure 7, although SFR has demonstrated its effectiveness, the influence of pose variations still negatively impacts its performance. Furthermore, the model's performance is highly dependent on computational resources during training. Addressing these two limitations will be a focus of our future work.

Author Contributions: Conceptualization, Q.O. and J.Z.; methodology, Q.O.; software, Q.O.; validation, Q.O. and J.Z.; formal analysis, Q.O.; investigation, Q.O.; resources, Q.O.; data curation, Q.O. and J.Z.; writing—original draft preparation, Q.O.; writing—review and editing, Q.O. and J.Z.; visualization, Q.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Goldblum, M.; Souri, H.; Ni, R.; Shu, M.; Prabhu, V.; Somepalli, G.; Chattopadhyay, P.; Ibrahim, M.; Bardes, A.; Hoffman, J.; et al. Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 29343–29371.

2. Sheykhmousa, M.; Mahdianpari, M.; Ghanbari, H.; Mohammadimanesh, F.; Ghamisi, P.; Homayouni, S. Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 6308–6325. [CrossRef]
3. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974.
4. Zhou, H.Y.; Guo, J.; Zhang, Y.; Han, X.; Yu, L.; Wang, L.; Yu, Y. nnFormer: Volumetric medical image segmentation via a 3D transformer. *IEEE Trans. Image Process.* **2023**, *32*, 4036–4045. [CrossRef] [PubMed]
5. Chen, W.Y.; Liu, Y.C.; Kira, Z.; Wang, Y.C.F.; Huang, J.B. A Closer Look at Few-shot Classification. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
6. Li, Z.; Tang, H.; Peng, Z.; Qi, G.J.; Tang, J. Knowledge-Guided Semantic Transfer Network for Few-Shot Image Recognition. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–15. [CrossRef] [PubMed]
7. Li, W.; Wang, L.; Xu, J.; Huo, J.; Gao, Y.; Luo, J. Revisiting local descriptor based image-to-class measure for few-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7260–7268.
8. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
9. Zhang, C.; Cai, Y.; Lin, G.; Shen, C. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 13–19 June 2020; pp. 12203–12213.
10. Xie, J.; Long, F.; Lv, J.; Wang, Q.; Li, P. Joint distribution matters: Deep brownian distance covariance for few-shot classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 7972–7981.
11. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.; Hospedales, T.M. Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1199–1208.
12. Wei, X.S.; Wu, J.; Cui, Q. Deep learning for fine-grained image analysis: A survey. *arXiv* **2019**, arXiv:1907.03069. [CrossRef] [PubMed]
13. Zhu, Y.; Liu, C.; Jiang, S. Multi-attention Meta Learning for Few-shot Fine-grained Image Recognition. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20), Yokohama, Japan, 7–15 January 2021; pp. 1090–1096.
14. Ye, H.J.; Hu, H.; Zhan, D.C.; Sha, F. Few-shot learning via embedding adaptation with set-to-set functions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 13–19 June 2020; pp. 8808–8817.
15. Doersch, C.; Gupta, A.; Zisserman, A. Crosstransformers: Spatially-aware few-shot transfer. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21981–21993.
16. Zhang, B.; Yuan, J.; Li, B.; Chen, T.; Fan, J.; Shi, B. Learning cross-image object semantic relation in transformer for few-shot fine-grained image classification. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022; pp. 2135–2144.
17. Yang, S.; Li, X.; Chang, D.; Ma, Z.; Xue, J.H. Channel-Spatial Support-Query Cross-Attention for Fine-Grained Few-Shot Image Classification. In Proceedings of the ACM Multimedia 2024, Melbourne, Australia, 28 October–1 November 2024.
18. Wertheimer, D.; Tang, L.; Hariharan, B. Few-shot classification with feature map reconstruction networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 20–25 June 2021; pp. 8012–8021.
19. Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. *The Caltech-Ucsd Birds-200-2011 Dataset*; Technical Report CNS-TR-2011-001; California Institute of Technology: Pasadena, CA, USA, 2011.
20. Krause, J.; Stark, M.; Deng, J.; Fei-Fei, L. 3D object representations for fine-grained categorization. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 2–8 December 2013; pp. 554–561.
21. Khosla, A.; Jayadevaprakash, N.; Yao, B.; Li, F.F. Novel dataset for fine-grained image categorization: Stanford dogs. In Proceedings of the First Workshop on Fine-Grained Visual Categorization, Seattle, WA, USA, 18 June 2011; Volume 2.
22. Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; Vedaldi, A. Fine-grained visual classification of aircraft. *arXiv* **2013**, arXiv:1306.5151.
23. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
24. Vaswani, A. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
25. Simon, C.; Koniusz, P.; Nock, R.; Harandi, M. Adaptive subspaces for few-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 13–19 June 2020; pp. 4136–4145.
26. Lee, K.; Maji, S.; Ravichandran, A.; Soatto, S. Meta-learning with differentiable convex optimization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10657–10665.

27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
28. Huang, H.; Zhang, J.; Zhang, J.; Xu, J.; Wu, Q. Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification. *IEEE Trans. Multimed.* **2020**, *23*, 1666–1680. [CrossRef]
29. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; Kavukcuoglu, K. Matching networks for one shot learning. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; Volume 29.
30. Afrasiyabi, A.; Lalonde, J.F.; Gagné, C. Mixture-based feature space learning for few-shot image classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 10–17 October 2021; pp. 9041–9051.
31. Zha, Z.; Tang, H.; Sun, Y.; Tang, J. Boosting few-shot fine-grained recognition with background suppression and foreground alignment. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 3947–3961. [CrossRef]
32. Tang, H.; Yuan, C.; Li, Z.; Tang, J. Learning attention-guided pyramidal features for few-shot fine-grained recognition. *Pattern Recognit.* **2022**, *130*, 108792. [CrossRef]
33. An, Y.; Xue, H.; Zhao, X.; Wang, J. From Instance to Metric Calibration: A Unified Framework for Open-World Few-Shot Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 9757–9773. [CrossRef] [PubMed]
34. Subramanyam, R.; Heimann, M.; Jayram, T.; Anirudh, R.; Thiagarajan, J.J. Contrastive knowledge-augmented meta-learning for few-shot classification. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 2479–2487.
35. Li, L.; Deng, J.; Huang, Y.; Chen, Y.; Luo, W. Structural Subspace Learning for Few-shot Fine-grained Recognition. In Proceedings of the 2024 16th International Conference on Machine Learning and Computing, Shenzhen, China, 2–5 February 2024; pp. 693–699.
36. Xu, J.; Le, H.; Huang, M.; Athar, S.; Samaras, D. Variational feature disentangling for fine-grained few-shot classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 8812–8821.
37. Ma, Z.X.; Chen, Z.D.; Zhao, L.J.; Zhang, Z.C.; Luo, X.; Xu, X.S. Cross-Layer and Cross-Sample Feature Optimization Network for Few-Shot Fine-Grained Image Classification. In Proceedings of the 38th Annual AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; Volume 38, pp. 4136–4144.
38. Zhu, H.; Gao, Z.; Wang, J.; Zhou, Y.; Li, C. Few-shot fine-grained image classification via multi-frequency neighborhood and double-cross modulation. *IEEE Trans. Multimed.* **2024**, *26*, 10264–10278. [CrossRef]
39. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.