mathematics

MDPI

Article

A Principal Component Analysis-Based Feature Optimization Network for Few-Shot Fine-Grained Image Classification

Meijia Wang, Boyuan Zheng, Guochao Wang, Junpo Yang *, Jin Lu and Weichuan Zhang *

School of Electronic Information and Artificial Intelligence, Shaanxi University of Science and Technology, Xi'an 710021, China; 4672@sust.edu.cn (M.W.); 231612138@sust.edu.cn (B.Z.); zheyiji12@gmail.com (G.W.); lujin@sust.edu.cn (J.L.)

* Correspondence: 4231@sust.edu.cn (J.Y.); weichuan.zhang@sust.edu.cn (W.Z.)

Abstract: Feature map reconstruction networks (FRN) have demonstrated significant potential by leveraging feature reconstruction. However, the typical process of FRN gives rise to two notable issues. First, FRN exhibits high sensitivity to noise, particularly ambient noise, which can lead to substantial reconstruction errors and hinder the network's ability to extract meaningful features. Second, FRN is particularly vulnerable to changes in data distribution. Owing to the fine-grained nature of the training data, the model is highly susceptible to overfitting, which may compromise its ability to extract effective feature representations when confronted with new classes. To address these challenges, this paper proposes a novel main feature selection module (MFSM), which suppresses feature noise interference and enhances the discriminative capacity of feature representations through principal component analysis (PCA). Extensive experiments validate the effectiveness of MFSM, revealing substantial improvements in classification accuracy for few-shot fine-grained image classification (FSFGIC) tasks.

Keywords: few-shot fine-grained image classification; principal component analysis; principal component analysis-based feature optimization network

MSC: 68T07

Academic Editor: Jonathan Blackledge

Received: 27 January 2025
Revised: 6 March 2025
Accepted: 25 March 2025
Published: 27 March 2025

Citation: Wang, M.; Zheng, B.; Wang, G.; Yang, J.; Lu, J.; Zhang, W. A Principal Component Analysis-Based Feature Optimization Network for Few-Shot Fine-Grained Image Classification. *Mathematics* **2025**, *13*, 1098. https://doi.org/10.3390/math13071098

1. Introduction

In recent years, deep learning models in the field of computer vision have achieved remarkable progress in image recognition and classification tasks [1–6]. These models have outperformed human capabilities in both speed and accuracy. Fine-grained image classification, a crucial task in this field, aims to identify subtle differences among various sub-classes within the same basic category, such as distinguishing between different species of birds or flowers. Compared to conventional classification tasks, fine-grained image classification poses greater challenges, necessitating advanced feature extraction techniques [7] and the support of deep learning models. However, such research often depends on large-scale annotated datasets, which are costly and time-consuming to obtain. Consequently, few-shot learning (FSL) has emerged as a prominent research area, enabling effective learning with limited samples and thus reducing the reliance on extensive annotated data.

In this context, researchers have proposed various approaches leveraging meta-learning [8–10] and metric learning [11–17]. Among these, metric learning has gained widespread adoption due to its simplicity and efficiency. Most existing works employ fixed metrics or learnable modules to learn effective metric embeddings [14,18]. These embeddings aim to distinguish image categories based on the similarity or dissimilarity of

elements within the metric space. Specifically, they strive to minimize the distance between samples of the same class while maximizing the distance between samples of different classes. This method constructs a feature space and employs similarity-based classification with high efficiency. However, it struggles to capture fine-grained feature relationships, thereby limiting its generalization ability to new categories. Therefore, an improved idea based on image reconstruction is proposed [19–21], with few-shot classification using feature map reconstruction networks (FRN) [19] as a representative example. FRN addresses classification tasks by reconstructing the feature map of the query image within the latent space. Subsequently, representative networks such as Bi-FRN [20], SRM [21], and others have been widely adopted and have achieved promising results. However, these few-shot image classification studies have not adequately addressed two critical issues.

Firstly, FRN demonstrates significant sensitivity to noise originating from multiple sources, which can be attributed to several underlying factors. (1) Noise from backbone networks: the feature extraction process in backbone networks can introduce noise, thereby degrading the quality of the feature maps. (2) Inherent image noise: real-world images often contain various types of noise, such as background clutter, lighting variations, and sensor noise. These factors can distort the feature maps, thereby making it difficult for FRN to reconstruct the relevant features accurately. (3) Data augmentation noise: data augmentation operations, such as scaling [22], rotation [23], cropping [24], and color jittering [25], are commonly used to enhance the diversity of training data. However, these operations can introduce additional noise, which may distort the feature maps and complicate the task of accurately reconstructing relevant features for FRN. (4) Feature map reconstruction: reconstructing the feature map in the latent space is inherently prone to noise interference. Even small perturbations in the input can result in significant errors in the reconstructed features, thereby adversely affecting the classification performance.

Secondly, FRN exhibits significant sensitivity to changes in data distribution. Owing to the fine-grained nature of the training data, the model is highly susceptible to overfitting. This susceptibility impairs its ability to extract effective feature representations in new classes, thereby reducing classification accuracy. This sensitivity can be attributed to the following factors: (1) Overfitting to training data: the fine-grained details in the training data can cause the model to learn specific patterns that are not generalizable to new, unseen data. Consequently, the model performs poorly when applied to new classes. (2) Generalization to new classes: the model's inability to generalize to new classes is further exacerbated by the lack of diverse training samples. When the model encounters new classes with different distributions, it struggles to extract meaningful features, leading to a drop in classification accuracy.

To effectively address these challenges, this paper proposes a novel main feature selection module (MFSM) that leverages principal component analysis (PCA) to mitigate the impact of noise and enhance generalization capabilities. PCA is a robust technique capable of reducing dimensionality, filtering out noise, and extracting the most significant features from high-dimensional data. The MFSM utilizes PCA to selectively retain the most informative components of the feature maps, thereby filtering out noise and improving the robustness of the classification task, as illustrated in Figure 1a. Furthermore, the MFSM selects key features that are more resilient to changes in data distribution, thereby enhancing the model's ability to generalize to new classes, as depicted in Figure 1b. To validate the effectiveness of the proposed approach, a series of extensive and in-depth experiments were conducted across multiple standard datasets. The results demonstrate that, compared with existing methods, the PCA-based principal component extraction and noise removal strategies offer substantial performance improvements across each dataset, presenting a novel and efficient solution for fine-grained image classification tasks.

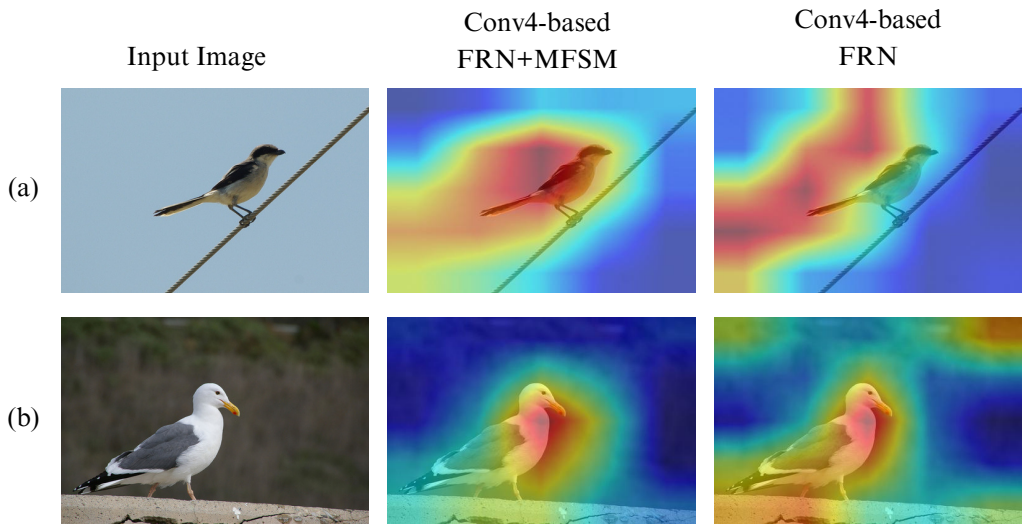

Figure 1. Visualization results of features extracted by the backbone network on the CUB dataset. In these two subfigures (**a**,**b**), we present photographs of two distinct bird species along with their corresponding feature visualization results. The first column shows the input raw images, the second column displays the visualization of features extracted by the FRN model with the added MFSM, and the third column shows the visualization of features extracted by the original FRN.

The contributions of this paper can be summarized as follows:

- The potential for optimization in feature characterization was identified in this study, and for the first time, a few-shot fine-grained image classification strategy based on principal component analysis (PCA) was proposed. The main feature selection module (MFSM) was introduced to analyze the orthogonal projections of sample features, thereby extracting and retaining the principal components that are useful for classification in feature characterization
- An automatic principal component selection mechanism, based on eigenvalue magnitude and the cumulative variance contribution rate, is proposed. This mechanism ensures the removal of irrelevant noise from the feature representation while minimizing the loss of useful information, thereby enhancing the accuracy of model training.
- Extensive validation was conducted across multiple few-shot datasets, thereby demonstrating the superior performance of the proposed method.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive review of the literature closely related to this study. The technical details of the main feature selection module (MFSM) are presented in Section 3. A substantial number of experimental results and ablation studies are reported in Section 4. Finally, Section 5 concludes the paper with a summary of the key findings.

2. Related Works

In the field of few-shot image classification, recent research advancements have introduced a variety of methodologies, which can be broadly categorized into two main paradigms. The first paradigm consists of meta-learning-based methods [8,26,27]. Strategies aim to develop an optimization mechanism that enables rapid adaptation to new tasks with minimal gradient adjustments. This is achieved by mapping training and test images to visual classification tasks using specific functions. The second main paradigm is metric-based few-shot learning [14,15,28–30]. The fundamental idea behind metric learning is to learn the distance relationships between samples, which allows the model to better understand the inherent features and category attributes of the samples. The metric function

is optimized by minimizing the distance between positive sample pairs and maximizing the distance between negative sample pairs.

2.1. Metric-Based Few-Shot Learning

In the domain of few-shot learning, metric-based learning approaches were among the first to be employed and have since garnered widespread attention due to their simplicity and efficiency. Good metric embeddings were acquired by utilizing fixed metrics or learnable modules in a multitude of early works [28]. Samples were classified based on distances or similarities. Siamese Network learns a similarity metric between two input examples [31]. The network inputs a pair of images and outputs a similarity score. It is trained to output a high similarity score for pairs of images from the same class and a low score for pairs from different classes. Triplet Network is based on the idea of learning a metric space in which the distance between examples of the same class is minimized while the distance between examples of different classes is maximized [32]. A triplet consists of an anchor example, a positive example (from the same class as the anchor), and a negative example (from a different class). The network is trained to minimize the distance between the anchor and positive examples and maximize the distance between the anchor and negative examples. Subsequently, prototypical networks were introduced [18], which obtain a prototype for a category in the feature space through samples of the same type. Classification is determined by measuring the distance between the prototype and other samples, thereby achieving proximity within the same category and separation between different categories. Feature reconstruction networks (FRN) were proposed to address the overfitting issue in few-shot fine-grained image classification by introducing feature reconstruction [19]. This approach retains spatial granularity and detail when transforming feature maps into vectors. However, the unidirectional reconstruction method of FRN only increases inter-class variation and fails to effectively handle intra-class issues [19]. To overcome these limitations, Bi-FRN was introduced, which employs a dual reconstruction mechanism [20]. In addition to using the support set to reconstruct the query set to increase inter-class variation, Bi-FRN also employs the query set to reconstruct the support set to reduce intra-class variation, thereby addressing both inter-class and intra-class changes simultaneously [20]. To mitigate the overfitting issue in metric learning, SRM introduced a self-reconstruction metric module to diversify query features and proposed a restrained cross-entropy loss to avoid overconfident predictions [21]. These innovations enable the self-reconstruction network to effectively alleviate overfitting. While these methods have demonstrated excellent performance, they have not considered the impact of noise on few-shot image classification. Building on FRN, the method proposed in this paper incorporates PCA to address the influence of noise, thereby reducing the sensitivity of feature reconstruction to data distribution and noise and achieving more accurate few-shot classification.

2.2. Meta-Based Few-Shot Learning

In model-agnostic meta-learning (MAML) [8], the concept of optimization-based methods was first introduced. The core idea of MAML is to learn an initial condition that facilitates rapid adaptation to new classification tasks by enabling easy adjustment of the model. In subsequent work [26], researchers innovatively proposed a "piecewise mapping" function within the classifier mapping module. This function learns a set of more attainable sub-classifiers to generate decision boundaries in a more parameter-efficient manner. Lee et al. [27] employed linear predictors as base learners to learn representations for few-shot learning, demonstrating that linear predictors offer better trade-offs between feature size and performance under linear classification rules. Reptile is a simple and

efficient meta-learning algorithm [9]. The model is trained on a sequence of tasks, and for each task, it performs multiple stochastic gradient descent steps. The key difference between Reptile and MAML is that Reptile does not explicitly compute the inner-loop gradient for each task. Instead, it updates the model parameters in a way that makes them move in the direction of the task-specific solutions. TAML focuses on adapting the learning rate for different tasks [33]. It has a meta-learner that learns the task-specific adaptation rules. The learning rate adaptation allows the model to better handle the diversity of different few-shot classification tasks. Meta-SGD aims to learn an optimizer that enables the model to adapt quickly to new tasks with only a small amount of data [10]. It achieves this by learning task-specific learning rates, thereby enhancing the model's adaptation speed and performance on new tasks. This method extends the traditional gradient descent optimization process, learning not only the model's initialization parameters but also the update direction and learning rate for each parameter. MTL combines meta-learning and transfer learning to address the issue of few-shot learning, hoping that by training on multiple related tasks, the model can quickly adapt to new tasks [34]. This method achieves this through parameter-level fine-tuning and a special "Scaling and Shifting" operation, adjusting only part of the weights, ensuring that most of the pre-trained parameters remain unchanged, preventing catastrophic forgetting, and also averting the problem of overfitting. These methods have achieved good results with limited data, but they may not have considered the impact of data noise on model performance. In contrast, our proposed MFSM can not only obtain more important features with limited data but also reduce the impact of noise on model performance caused by data corruption and limited data collection equipment.

3. Method

3.1. Problem Definition

In few-shot learning (FSL), given a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)|y_i \in \mathcal{Y}\}$, we can divide this dataset into three parts: $\mathcal{D}_{\text{base}} = \{(\boldsymbol{x}_i, y_i)|y_i \in \mathcal{Y}_{\text{base}}\}$, $\mathcal{D}_{\text{val}} = \{(\boldsymbol{x}_i, y_i)|y_i \in \mathcal{Y}_{\text{val}}\}$, and $\mathcal{D}_{\text{novel}} = \{(\boldsymbol{x}_i, y_i)|y_i \in \mathcal{Y}_{\text{novel}}\}$. Here, $\boldsymbol{x}_i$ and y_i represent the feature vector and class label of the i-th image, respectively. The class sets of these three subsets are disjoint, meaning $\mathcal{Y}_{\text{base}} \cap \mathcal{Y}_{\text{val}} \cap \mathcal{Y}_{\text{novel}} = \emptyset$ and $\mathcal{Y}_{\text{base}} \cup \mathcal{Y}_{\text{val}} \cup \mathcal{Y}_{\text{novel}} = \mathcal{Y}$. The primary objective of few-shot classification is to enable the model to perform well on the new classes in $\mathcal{D}_{\text{novel}}$ in an $\mathcal{N}$-way $\mathcal{K}$-shot setting by learning knowledge from $\mathcal{D}_{\text{base}}$ and $\mathcal{D}_{\text{val}}$. Here, $\mathcal{N}$ represents the number of classes randomly selected from $\mathcal{D}_{\text{novel}}$, with $\mathcal{K}$ labeled support samples chosen for each class and $\mathcal{R}$ unlabeled query samples in the query set $\boldsymbol{Q}$. During the meta-training phase, the model uses the support samples from tasks in $\mathcal{D}_{\text{base}}$ to infer the labels of the query samples. The optimal model is then selected by evaluating performance across multiple tasks in $\mathcal{D}_{\text{val}}$. In the meta-testing phase, the model is evaluated on the novel classes in $\mathcal{D}_{\text{novel}}$ to verify whether the knowledge learned during the meta-training phase can be effectively transferred and generalized to new classes.

3.2. Main Feature Selection Module

In few-shot fine-grained image classification, the model is tasked with learning to extract highly discriminative features from limited samples, which are essential for distinguishing between subtly different subclasses. However, during this process, the obtained feature maps contain significant noise due to several factors. Such noise complicates the extraction of effective features, thereby resulting in an unnecessary expansion of the feature space due to the potential presence of numerous redundant features in high-dimensional spaces. To address the aforementioned issues, this paper introduces the main feature

selection module (MFSM), which enhances model performance in complex image scenarios by selectively extracting key features from diverse and high-dimensional feature spaces.

By leveraging this module, models are enabled to filter and optimize features more efficiently, thereby eliminating redundant information and noise. This process significantly enhances the overall quality of feature representation and improves classification performance. In contrast to traditional methods that lack feature optimization, the MFSM is more focused on retaining the essential features necessary for the task. This targeted approach leads to substantial improvements in the network's training efficiency, model generalization capability, and overall effectiveness.

The main feature selection module (MFSM) employs principal component analysis (PCA) [35,36] to perform initial feature selection, thereby enhancing the feature representation derived from the backbone network. Principal component analysis (PCA) involves computing the covariance matrix of the data, identifying the eigenvectors and their corresponding eigenvalues, and sorting these eigenvectors in descending order based on their eigenvalues. The top eigenvectors are selected as the new basis vectors, and the projection of the data onto these vectors constitutes the principal components. By employing this method, the MFSM focuses on the most relevant information for image classification, effectively eliminating irrelevant features and noise while retaining only the principal components that significantly impact the data distribution. Taking FRN as an example (illustrated in Figure 2), the dataset is divided into a support set X_s and a query set X_q based on the parameter settings. After sampling and preprocessing, the backbone network extracts high-dimensional features from the original images, which are projected into a more discriminative latent space. The feature maps are reshaped into feature pools, and each image's feature channels are fed into the main feature selection module (MFSM) as input samples for principal component analysis (PCA). This process generates a refined feature representation that more effectively captures the key information of the samples. Subsequently, the Euclidean distance is used to measure similarity, comparing samples from the support set and the query set to identify the most closely related samples for learning. In this manner, the MFSM focuses on the information most pertinent to the classification task, thereby eliminating extraneous features and noise while retaining only the principal components that best represent the data distribution. The specific methodological workflow is detailed as follows.

First, a dataset comprising support instances and query instances is provided. Classic backbone networks, such as ResNet-12 and Conv-4, are selected as feature extractors to compute feature maps for preliminary feature representation extraction of the input samples. These backbone networks, which have been extensively trained and optimized, are capable of effectively capturing low-level and mid-level features of images, such as edges, textures, and shapes, as described in Equation (1):

$$F = f_\theta(I). \tag{1}$$

Here, $\boldsymbol{I}$ denotes the input query and support image samples, and f_θ represents the feature extractor parameterized by θ. The resulting feature map $\boldsymbol{F}$ has dimensions $\boldsymbol{F} \in \mathbb{R}^{r \times d}$, where r is the new dimension obtained by flattening the original feature map's height and width, and d is the number of channels in the extracted feature map. Thus, $\boldsymbol{F}$ can be defined as the sample matrix for this module, comprising d samples, each with r features. Following the principles of PCA, adhering to the principles of PCA, the proposed MFSM performs principal component selection on these samples, as illustrated in Figure 3.

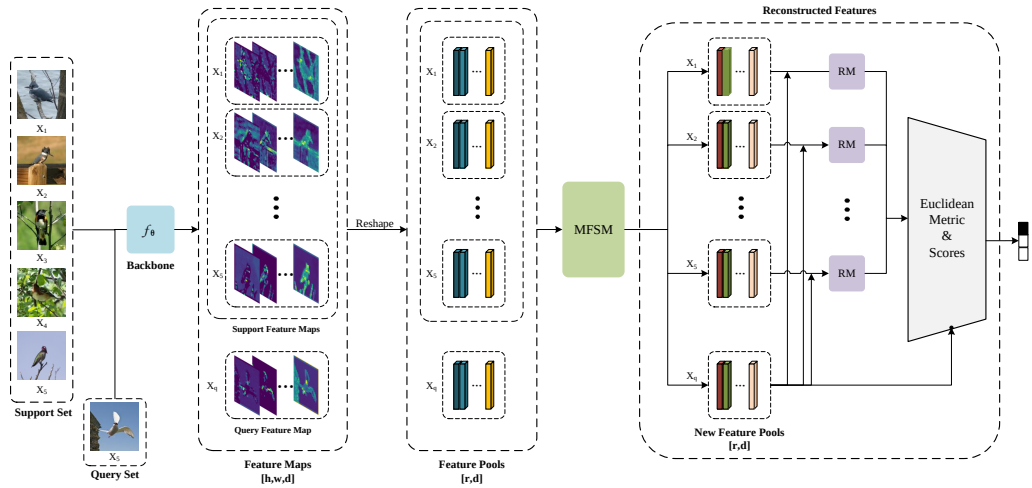

Figure 2. A schematic diagram of the FRN network with the MFSM module that illustrates the process of projecting support and query images into the feature space using the mapping function $f(\theta)$. After flattening the feature maps, their spatial structure is transformed into representations of relationships among samples. Specifically, the flattened feature map channels are treated as independent samples, with the features of each sample corresponding to the number of elements in the flattened feature pool. Subsequently, Principal component analysis (PCA) is applied to process these features, optimizing feature representation and enhancing classification accuracy.

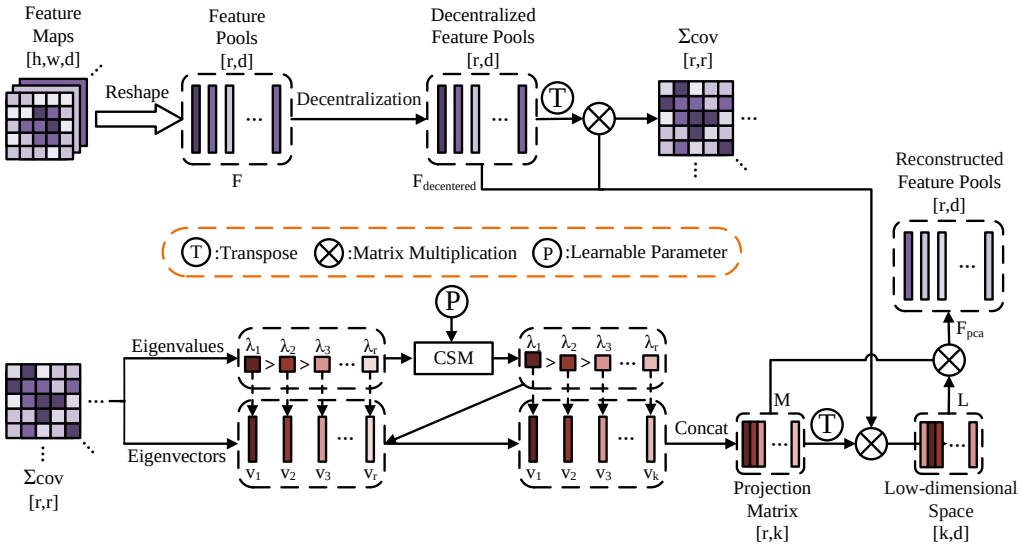

Figure 3. MFSM schematic diagram.

Based on the obtained feature representation $\boldsymbol{F}$, the corresponding centralized matrix $\boldsymbol{U}$ can be computed. Subsequently, $\boldsymbol{F}$ is decentralized by subtracting the mean from each feature using $\boldsymbol{U}$.

$$\boldsymbol{U} = \begin{bmatrix} \overline{\boldsymbol{x}}_0 & \overline{\boldsymbol{x}}_1 & \cdots & \overline{\boldsymbol{x}}_{d-2} & \overline{\boldsymbol{x}}_{d-1} \\ \overline{\boldsymbol{x}}_0 & \overline{\boldsymbol{x}}_1 & \cdots & \overline{\boldsymbol{x}}_{d-2} & \overline{\boldsymbol{x}}_{d-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \overline{\boldsymbol{x}}_0 & \overline{\boldsymbol{x}}_1 & \cdots & \overline{\boldsymbol{x}}_{d-2} & \overline{\boldsymbol{x}}_{d-1} \\ \overline{\boldsymbol{x}}_0 & \overline{\boldsymbol{x}}_1 & \cdots & \overline{\boldsymbol{x}}_{d-2} & \overline{\boldsymbol{x}}_{d-1} \end{bmatrix} \quad \left(\overline{\boldsymbol{x}}_j = \frac{\sum_{i=0}^{r-1} \boldsymbol{x}_{(i,j)}}{r}, \quad j = 0, 1, \ldots, d-1\right). \tag{2}$$

Using Equation (2), the decentered matrix $\boldsymbol{F}_{\text{decentered}}$ is obtained, with dimensions $\boldsymbol{F}_{\text{decentered}} \in \mathbb{R}^{r \times d}$. This matrix serves to center the data around the origin, effectively

eliminating feature biases and ensuring that the data distribution remains invariant to shifts. This step ensures that the data is accurately centered around the origin, thereby reflecting the variability of the data and facilitating the correctness of subsequent eigenvalue decomposition. Subsequently, the covariance matrix $\mathbf{\Sigma}_{\text{cov}}$ is computed to further analyze the variance and common characteristics of the data. The covariance matrix is a critical metric that reflects the correlations between features. By measuring the covariance between features, it is possible to identify the principal component directions and the weights associated with each feature, thereby facilitating the extraction of the most representative features. Specifically, the covariance matrix $\mathbf{\Sigma}_{\text{cov}}$ is constructed by treating each column of $\boldsymbol{F}_{\text{decentered}}$ as a feature vector and calculating the inner product relationships among these vectors, as shown in Equation (3).

$$\mathbf{\Sigma}_{\text{cov}} = \frac{1}{r-1} \boldsymbol{F}_{\text{decentered}} \boldsymbol{F}_{\text{decentered.}}^{T} \tag{3}$$

The covariance matrix $\mathbf{\Sigma}_{\text{cov}}$ is constructed with dimensions $\mathbf{\Sigma}_{\text{cov}} \in \mathbb{R}^{r \times r}$. To obtain an unbiased estimate of the covariance matrix, which corrects for the bias in the sample covariance, the matrix is scaled by dividing by $r-1$. This adjustment quantifies the linear relationships between each pair of features in the feature representation, leading to the formulation of a typical eigenvalue problem. Subsequently, the eigenvalues and corresponding eigenvectors of this problem are computed. The calculation of these eigenvalues and eigenvectors is essential for identifying the principal directions of variation within the feature representation. These solutions enable the identification and retention of the most discriminative components within the data.

$$\begin{aligned} \mathbf{\Sigma}_{\text{cov}} \boldsymbol{v} &= \lambda \boldsymbol{v}, \\ \lambda_1, \lambda_2, \ldots, \lambda_r &\rightarrow \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k. \end{aligned} \tag{4}$$

Given that the covariance matrix $\mathbf{\Sigma}_{\text{cov}}$ is symmetric, according to Equation (4), it is possible to obtain r eigenvalues and their corresponding eigenvectors. Each eigenvalue λ_i is associated with an eigenvector $\boldsymbol{v}_i$, with the eigenvalues ordered in descending magnitude such that $\lambda_j > \lambda_{j+1}$. The eigenvalues are subsequently fed into the component selection module (CSM) for principal component selection. The detailed process and associated pseudocode for the CSM module are presented in Section 3.3. The CSM module evaluates the magnitude and cumulative contribution rate of these eigenvalues to identify the principal components that capture the dominant features of the data, thereby discarding irrelevant or redundant information. The first k eigenvectors, which represent the principal components, are selected as follows:

$$\boldsymbol{M} = cat(v_1, v_2, \ldots, v_k) = [v_1, v_2, \ldots, v_k]. \tag{5}$$

In Equation (5), the CSM module identifies the top k eigenvectors corresponding to the largest k eigenvectors $\lambda_1, \lambda_2, \ldots, \lambda_k$. These eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k$ are designated as the principal component directions. The eigenvalues quantify the variance of the data along the corresponding eigenvector directions, thereby reflecting the information content. By selecting these eigenvectors, we determine the directions of the principal components. This selection enables the projection of the data into a lower-dimensional space while preserving the original structural information and discriminative power of the data. The resulting projection matrix $\boldsymbol{M}$ (with dimensions $\boldsymbol{M} \in \mathbb{R}^{r \times k}$) enables this dimensionality reduction process, where r denotes the original feature dimension, and k represents the number of selected principal components. This projection matrix effectively eliminates redundant

information while enhancing the model's sensitivity to the main features, thereby providing stronger discriminability for subsequent processing and classification.

Subsequently, the projection matrix $\boldsymbol{M}$ is multiplied by the centered matrix $\boldsymbol{F}_{\text{decentered}}$ to obtain the new reduced-dimensional data matrix $\boldsymbol{L}$:

$$\boldsymbol{L} = \boldsymbol{M}^T \boldsymbol{F}_{\text{decentered}}. \tag{6}$$

Here, $\boldsymbol{L}$ has dimensions $\boldsymbol{L} \in \mathbb{R}^{k \times d}$. The new data matrix $\boldsymbol{L}$ retains the main information from the original data in the k new dimensions while eliminating irrelevant sample noise. In Equation (6), by multiplying $\boldsymbol{M}$ with $\boldsymbol{L}$. Subsequently, by using Equation (7), the reduced-dimensional matrix $\boldsymbol{L}$ is projected back into the original feature space to obtain $\boldsymbol{F}_{\text{pca}}$:

$$\boldsymbol{F}_{\text{pca}} = \boldsymbol{M}\boldsymbol{L}. \tag{7}$$

$\boldsymbol{F}_{\text{pca}}$ has dimensions $\boldsymbol{F}_{\text{pca}} \in \mathbb{R}^{r \times d}$ maintains consistency with the original sample matrix $\boldsymbol{F}$ in terms of dimensionality and spatial representation. However, $\boldsymbol{F}_{\text{pca}}$ retains only the most representative principal components of the data, thereby more accurately capturing the core features of the samples. This denoising process enhances the signal-to-noise ratio of the data, rendering subsequent image classification tasks more reliable and efficient.

3.3. Component Selection Module

In the main feature selection module (MFSM), the sample matrix is first centralized to obtain the decentered sample matrix $\boldsymbol{F}_{\text{decentered}}$. Subsequently, the corresponding covariance matrix $\boldsymbol{\Sigma}_{\text{cov}}$ is computed, and its eigenvalues λ_i and corresponding eigenvector $\boldsymbol{v}_i$ are determined. The eigenvalues quantify the variance of the data along the directions of the corresponding eigenvectors. Generally, these eigenvalues reflect the amount of information or "energy" contained in the data along those directions. A larger eigenvalue indicates a more significant variation in the data along that direction, thereby highlighting its importance. The eigenvector $\boldsymbol{v}_i$ is a unit vector that represents a specific direction in the high-dimensional space, reveals the trend of data variation in that direction, and indicates the principal direction of the data during dimensionality reduction. The selection of the eigenvectors corresponding to the first k largest eigenvalues enables the construction of a new low-dimensional space that projects the data in these principal directions. This approach effectively reduces the dimensionality of the data while retaining the most important feature information, thereby achieving dimensionality reduction and feature optimization.

To ensure the selection of an appropriate number of feature components without losing important information, a component selection module (CSM) is proposed. This module is designed to select the optimal number of features, and the algorithm is described as Algorithm 1.

Initially, the module processes the eigenvalues from the set of eigenvalues and eigenvectors, as the magnitude of each eigenvalue λ_i reflects the importance of each principal component in explaining the variance of the data. Correspondingly, each eigenvector v_i represents the direction of this component mapped onto a new axis. Additionally, a learnable parameter P is initialized, which sets the desired cumulative variance percentage to retain, typically within the range $0 \leq P \leq 1$.

Algorithm 1: Component selection module.

Data: Learnable parameter $P \in [0, 1]$; Matrix E of size $B \times R$, where B represents the batch size (the number of samples in a batch), and R represents the resolution (the number of eigenvalues of the covariance matrix under each batchsize).

Result: Vector num of size B

Step 1: Sort Eigenvalues:
for $i = 1$ *to* B **do**
  Sort $E[i, :]$ in descending order to get $S[i, :]$. Record original indices in $I[i, :]$.
end
Step 2: Compute Total Eigenvalue Sums:
Initialize sum as a vector of size B with all elements 0.
for $i = 1$ *to* B **do**
  for $j = 1$ *to* R **do**
    $sum[i] \leftarrow sum[i] + S[i, j]$
  end
end
Step 3: Initialize Result Vector: Set num to a vector of size B filled with 0.
Step 4: Determine Selected Eigenvalue Counts:
for $i = 1$ *to* B **do**
  $tmpSum \leftarrow 0$
  for $j = 1$ *to* R **do**
    $tmpSum \leftarrow tmpSum + S[i, j]$
    if $tmpSum \geq P \times sum[i]$ **then**
      $num[i] \leftarrow j$ Break
    end
  end
end
Step 5: Return Result: return num

In the initial step, the top r eigenvalues λ are computed from the covariance matrix and sorted in descending order to form a sequence: $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r$. This sorting ensures that eigenvalues with the highest information content are positioned at the beginning of the sequence, while those with less information are placed towards the end. This arrangement prioritizes the most informative components in the cumulative sum.

Next, the sum of the sorted eigenvalues is computed, denoted as sum $= \lambda_1 + \lambda_2 + \ldots + \lambda_r$. This sum represents the total contribution of all the principal components to the variance of the original data, providing a benchmark for determining whether the cumulative sum of eigenvalues reaches the set variance retention threshold. Based on this, a temporary variable tmpsum is initialized and assigned the value of the largest eigenvalue λ_1. Starting with the largest eigenvalue, each subsequent eigenvalue is progressively added, dynamically assessing whether the cumulative sum has reached the threshold $P \times$ sum, which represents a certain percentage of the total variance.

The current cumulative sum tmpsum is then compared to $P \times$ sum. If tmpsum is greater than or equal to $P \times$ sum, it indicates that the selected number of components can explain $P \times 100\%$ of the variance in the data, and the selection process terminates. If the condition is not satisfied, the counter is incremented, and the next eigenvalue is added to the cumulative sum until the threshold is reached or exceeded.

Upon satisfying the condition, the module outputs k, the count of the top k eigenvalues, which represents the number of components required to achieve the preset variance

explanation rate. This automatic selection mechanism identifies the optimal number of principal components k based on eigenvalue magnitude and cumulative contribution, thereby eliminating the need for manually setting a threshold. The time complexity and space complexity of the CSM module are both O(n), meaning that the computational load is not heavy, which meets the requirement for the module to be lightweight.

4. Experiments

In this section, the performance of the main feature selection module (MFSM) is evaluated on several standard fine-grained classification benchmarks. To fully validate the broad adaptability and enhancement of feature representation provided by MFSM, it is integrated into existing classic network structures, ensuring consistent performance across different model frameworks. Initially, the main feature selection module (MFSM) is embedded into the feature reconstruction network (FRN) [19]. By leveraging the strong feature extraction capability of FRN and the proposed MFSM, the aim is to further enhance performance in fine-grained classification tasks. To ensure the fairness and comparability of the results, the same hyperparameter configuration and implementation details are maintained across all experiments. Specifically, consistent training procedures and data preprocessing methods are employed in both the original FRN network without MFSM integration and the improved version with MFSM. This design ensures that any performance improvement can be attributed to the contribution of MFSM itself rather than other external factors.

4.1. Datasets

To validate the effectiveness of the proposed method, four standard fine-grained image datasets were selected for performance evaluation: CUB-200-2011 [37], Stanford-Dogs [38], Stanford-Cars [39], and Aircraft [40]. The data division for each dataset is provided in Table 1.

Table 1. Dataset division. M_{all} represents the total number of classes, and M_{train}, M_{val}, and M_{test} are the numbers of classes in the auxiliary set, validation set, and test set, respectively.

Dataset	M_{all}	M_{train}	M_{val}	M_{test}
CUB-200-2011	200	100	50	50
Stanford-Dogs	120	70	20	30
Stanford-Cars	196	130	17	49
Aircraft	100	100	50	50

The CUB-200-2011 dataset [37] is a widely used fine-grained image classification dataset containing 11,788 images from 200 different bird species. Consistent with the methods of Zhang et al. [41] and Ye et al. [42], each image was cropped to match the annotated bounding boxes while preserving the original image form as specified in [43]. The data split aligns with that in [19].

The Stanford-Dogs dataset [38] is a highly significant dataset, covering 120 different dog breeds from around the world, with a total of 20,580 images. These images display the unique characteristics and forms of various dog breeds. The dataset split follows the configurations in [44,45] to ensure experimental comparability and validity.

The Stanford-Cars dataset [39] is a highly influential car image dataset, containing 196 different car categories defined by car make, model, and year. The images are primarily taken from the rear view of the cars, ensuring consistency across the dataset, which facilitates model training and comparison. The dataset is divided into a training set with 8144 images and a test set with 8041 images. This nearly 50–50 split allows for a comprehensive evaluation of the model's generalization capabilities.

The Aircraft dataset [40] contains 10,000 aircraft images covering 100 different aircraft models. Consistent with the approach in [19], each aircraft in the images has been accurately annotated with a bounding box and cropped accordingly. The dataset is divided for experimental use.

4.2. Implementation Details

Few-shot image classification experiments were conducted using two widely used backbone architectures: Conv-4 [27,46] and ResNet-12 [1,27]. These experiments are based on the FRN network method, operating on the feature representations extracted by these two networks. All experiments in this work were performed using the PyTorch 2.1.1 framework on a single NVIDIA 3090 Ti GPU (24GB). For the Stanford-Dogs and Stanford-Cars datasets, all settings are in accordance with the FRN replication experiment parameters mentioned in Bi-FRN [20]. For the Aircraft and CUB-200-2011 datasets, the parameter settings from FRN [19] are followed. Experiments are conducted in the code environment set up in the respective papers to ensure the rigor of the experiments. The initial learning rate for all datasets is set to 0.1, with a weight decay of $5e^{-4}$. Training is performed for a total of 1200 and 800 epochs, with the learning rate reduced by a factor of 10 every 400 epochs and validation performed every 20 epochs. Data augmentation is applied using random cropping, horizontal flipping, and color jittering. Experiments are conducted under the standard five-way one-shot and five-shot settings, with 10,000 tasks randomly generated on the test datasets. The average classification accuracy within a 95% confidence interval is reported.

4.3. Performance Comparison

In this section, the classification performance of the proposed MFSM module combined with FRN is compared against thirteen state-of-the-art and typical methods: ProtoNet [18], DN4 [44], DSN [16], CTX [47], DeepEMD [41], MattML [24], MixFSL [48], FRN [19], LMP-Net [49], OLSA [50], HelixFormer [51], TOAN [52], and BSFA [53]. The experimental results on the CUB-200-2010, Aircraft, Stanford-Dogs, and Stanford-Cars datasets are detailed in Table 2. For methods marked with †, experiments are reproduced under the same conditions using their publicly available code to ensure the reliability and consistency of the results. By re-implementing these methods in a unified environment, a fair evaluation of different methods' performance in few-shot fine-grained image classification (FSFGIC) tasks is achieved, providing a solid experimental foundation for the direct comparison of model performance.

Additionally, as illustrated in Figure 4, the gradient-weighted class activation mapping (Grad-CAM) technique was applied to the ResNet-12 layer to visualize the model's attention regions, thereby demonstrating the advantages of the proposed MFSM module. In the Grad-CAM method [54], this technique provides visual explanations for the decisions made by CNN-based models, enhancing the transparency and interpretability of the model's decision process. Areas with higher energy typically represent more discriminative parts of the image, which are crucial for classification tasks. As shown in Figure 4, FRN+MFSM exhibits a stronger ability to focus on classification targets than FRN. It captures target areas more accurately and effectively reduces attention to background interference areas. This improvement indicates that the MFSM module enhances the robustness of FRN, enabling it to focus more effectively on target features in fine-grained classification tasks, thereby improving classification accuracy.

Table 2. Experimental comparison results of different methods on the CUB-200-2011, Stanford-Dogs, Stanford-Cars, and Aircraft datasets under two backbone networks (methods marked with † are those we implemented). The best performance is highlighted in bold.

Backbone	Methods	CUB-200-2011		Aircraft		Stanford-Dogs		Stanford-Cars	
		1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
Conv-4	ProtoNet † [18]	61.76 ± 0.23	83.07 ± 0.15	47.72	69.42	46.66 ± 0.22	70.93 ± 0.16	50.57 ± 0.22	74.44 ± 0.17
	DN4 [44]	57.45 ± 0.89	84.41 ± 0.58	-	-	39.08 ± 0.76	69.81 ± 0.69	34.12 ± 0.68	87.47 ± 0.47
	DSN [16]	72.56 ± 0.92	84.62 ± 0.60	-	-	44.52 ± 0.82	59.42 ± 0.71	53.45 ± 0.86	65.19 ± 0.75
	CTX [47]	72.61 ± 0.21	86.23 ± 0.14	50.02	67.25	57.86 ± 0.21	73.59 ± 0.16	66.35 ± 0.21	82.25 ± 0.14
	DeepEMD [41]	64.08 ± 0.50	80.55 ± 0.71	-	-	46.73 ± 0.49	65.74 ± 0.63	61.63 ± 0.27	72.95 ± 0.38
	MattM L [24]	66.29 ± 0.56	80.34 ± 0.30	-	-	54.84 ± 0.53	71.34 ± 0.38	66.11 ± 0.54	82.80 ± 0.28
	MixFSL [48]	53.61 ± 0.88	73.24 ± 0.75	44.89 ± 0.75	62.81 ± 0.73	-	-	44.56 ± 0.80	59.63 ± 0.79
	FRN † [19]	73.82 ± 0.21	88.16 ± 0.12	53.20	71.17	60.41 ± 0.21	79.26 ± 0.15	67.12 ± 0.22	86.62 ± 0.12
	FRN+MFSM	**74.48 ± 0.21**	**89.17 ± 0.12**	**54.01 ± 0.21**	**71.81 ± 0.18**	**60.54 ± 0.22**	**79.29 ± 0.14**	**67.23 ± 0.22**	**87.48 ± 0.12**
ResNet-12	DeepEMD [41]	75.59 ± 0.30	88.23 ± 0.18	-	-	70.38 ± 0.30	85.24 ± 0.18	80.62 ± 0.26	92.63 ± 0.13
	LMPNet [49]	-	-	-	-	61.89 ± 0.10	68.21 ± 0.11	68.31 ± 0.45	68.31 ± 0.45
	MixFSL [48]	67.87 ± 0.94	82.18 ± 0.66	60.55 ± 0.86	77.57 ± 0.69	-	-	58.15 ± 0.87	80.54 ± 0.63
	OLSA [50]	77.77 ± 0.44	89.87 ± 0.24	-	-	64.15 ± 0.49	78.28 ± 0.32	77.03 ± 0.46	88.85 ± 0.46
	FRN † [19]	83.05 ± 0.19	92.47 ± 0.10	69.81 ± 0.22	82.99 ± 0.14	71.59 ± 0.21	85.41 ± 0.13	88.01 ± 0.17	95.75 ± 0.07
	HelixFormer [51]	81.66 ± 0.30	91.83 ± 0.17	-	-	65.92 ± 0.49	80.65 ± 0.36	79.40 ± 0.43	92.26 ± 0.15
	TOAN [52]	66.10 ± 0.86	82.27 ± 0.60	-	-	49.77 ± 0.86	69.29 ± 0.70	75.28 ± 0.72	87.45 ± 0.48
	BSFA [53]	82.27 ± 0.46	90.76 ± 0.26	-	-	69.58 ± 0.50	82.59 ± 0.33	**88.93 ± 0.38**	95.20 ± 0.20
	FRN+MFSM	**84.86 ± 0.18**	**93.73 ± 0.09**	**69.99 ± 0.21**	**83.74 ± 0.13**	**71.66 ± 0.22**	**85.46 ± 0.13**	88.19 ± 0.17	**96.02 ± 0.07**

Figure 4. Heatmaps of seven images visualized by FRN and the proposed FRN+MFSM.

To more comprehensively verify the effectiveness of the proposed algorithm, MFSM was integrated into ProtoNet, and experiments were conducted using Conv-4 as the backbone network. Figure 5 intuitively presents the performance differences and comparison results under the five-way one-shot and five-way five-shot settings, further demonstrating the effectiveness of the proposed method.

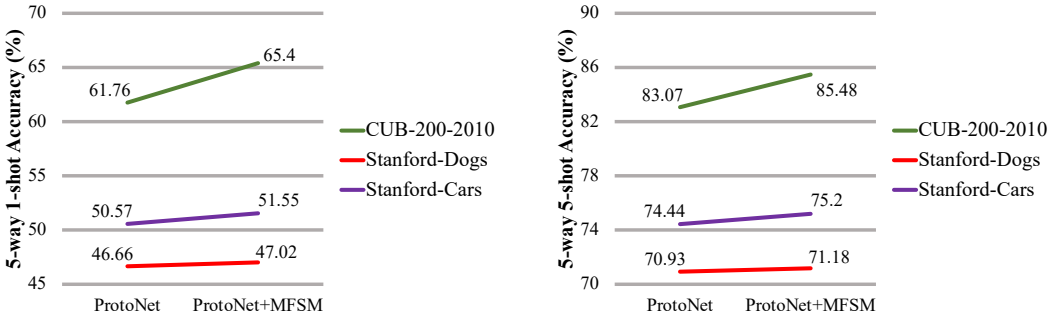

Figure 5. Under the 5-way 1-shot and 5-way 5-shot settings, our method is applied to ProtoNet for performance comparison with the original version.

4.4. Ablation Studies

The performance of the proposed MFSM module in the FSFGIC task depends on the choice of parameter P. To investigate its impact within the CSM module, additional experiments were conducted on FSFGIC. Using a Conv-4 backbone, we empirically selected four different P values $P \in \{0.95, 0.96, 0.97, 0.98\}$ to explore their effect on performance. The five-way one-shot and five-way five-shot FSFGIC tasks were conducted on three datasets: Aircraft, Stanford-Cars, and Stanford-Dogs datasets. According to the results shown in Table 3, the proposed MFSM module achieved the best classification performance on the Aircraft dataset when $P = 0.96$, and optimal classification performance on the Stanford-Cars and Stanford-Dogs datasets when $P = 0.97$.

Table 3. Performance metrics on the Stanford-Dogs, Stanford-Cars, and Aircraft datasets were evaluated under different values of P, with the best performance highlighted in bold.

P	Aircraft		Stanford-Cars		Stanford-Dogs	
	1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
0.95	52.46 ± 0.21	70.37 ± 0.18	-	-	-	-
0.96	**54.01 ± 0.21**	**71.81 ± 0.18**	67.23 ± 0.22	87.48 ± 0.12	60.23 ± 0.21	79.07 ± 0.25
0.97	51.83 ± 0.21	70.61 ± 0.18	**67.23 ± 0.22**	**87.88 ± 0.12**	**60.54 ± 0.22**	**79.29 ± 0.14**
0.98	53.18 ± 0.21	70.93 ± 0.18	66.74 ± 0.22	87.87 ± 0.12	60.05 ± 0.22	78.61 ± 0.15

This situation may be related to differences in feature characteristics and complexity across datasets. For example, in the Aircraft dataset, images often focus on the shape features of the aircraft, with relatively consistent angles and backgrounds. This results in higher structural similarity and typically high intra-class variance. A lower P value helps retain the main features while reducing noise interference as much as possible without affecting overall performance. In contrast, the Stanford-Cars and Stanford-Dogs datasets contain more detailed variations, such as different poses, background changes, and color differences for cars and dogs. These rich details increase the difficulty for the model in distinguishing fine-grained features, requiring a higher cumulative variance to fully capture and express these details. This difference reflects the need for flexibility in feature selection across datasets, explaining why optimal performance is achieved with different P values for different datasets.

4.5. Time Complexity Analysis

In this work, the time complexity of the FRN method with the proposed MFSM was compared with that of the FRN without this module in terms of the forward/backward pass size, estimated total size, number of parameters, and floating-point operations (FLOP). Both methods used Conv-4 as the backbone. As shown in Table 4, the proposed method maintains the same number of parameters and FLOPs as the original model, while the forward/backward pass size and estimated total size are slightly higher than those of FRN. Despite a negligible increase in computational overhead, the model achieves significantly better performance than FRN. This demonstrates that the MFSM module achieves a favorable balance between computational efficiency and performance improvement.

Table 4. Time complexity analysis for the proposed method.

Model	Forward/Backward Pass Size (MB)	Estimated Total Size (MB)	Parameters (M)	FLOPs (M)
FRN	19.43	19.95	0.113088	100.16
FRN+MFSM	19.45	19.96	0.113088	100.16

5. Conclusions

This paper proposes the main MFSM, which effectively removes redundant information and noise through PCA, thereby enhancing the robustness and discriminability of feature representations. MFSM optimizes the data distribution in the feature space, reduces interference from background regions, and makes the feature representation more compact and discriminative. Performance comparison experiments demonstrate that MFSM significantly improves classification performance on four widely used fine-grained datasets: CUB-200-2011, Stanford-Cars, Stanford-Dogs, and Aircraft, with particular advantages in complex background scenarios. Additionally, ablation experiments were conducted to investigate the impact of the parameter P on the fine-grained image classification task (FS-FGIC) across different datasets, providing a deeper understanding of how different datasets are sensitive to the choice of P and how to adjust it to optimize classification performance based on dataset characteristics and complexity. The following literature was also referenced during the course of this paper [55–63] to refine the research framework.

6. Future Work

In future work, we will explore combining other denoising techniques and feature selection methods to further improve MFSM's performance on complex datasets. Furthermore, considering the generalization ability of our method in few-shot learning, strategies like meta-learning will be incorporated to enhance MFSM's adaptability and robustness

to new categories. Additionally, MFSM can be integrated with other deep learning architectures. For example, combining MFSM with models like transformer to study its performance in tasks with long-tailed distributions or cross-modal tasks can help expand its application potential in other domains.

Author Contributions: Methodology, B.Z.; Software, B.Z. and G.W.; Validation, J.L.; Investigation, B.Z.; Writing—original draft, M.W. and B.Z.; Writing—review & editing, G.W., J.Y. and W.Z.; Visualization, B.Z. and G.W.; Supervision, W.Z.; Project administration, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The project is supported by the Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No. 22JK0303) and the Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2022JQ-175).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. He, K.; Zhang, X.; Ren, S. 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.
2. Yang, Y.; Hu, Y.; Zhang, X.; Wang, S. Two-stage selective ensemble of CNN via deep tree training for medical image classification. *IEEE Trans. Cybern.* **2021**, *52*, 9194–9207. [CrossRef]
3. Jing, J.; Liu, S.; Wang, G.; Zhang, W.; Sun, C. Recent advances on image edge detection: A comprehensive review. *Neurocomputing* **2022**, *503*, 259–271. [CrossRef]
4. Jing, J.; Gao, T.; Zhang, W.; Gao, Y.; Sun, C. Image feature information extraction for interest point detection: A comprehensive review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 4694–4712. [CrossRef] [PubMed]
5. Ren, J.; Li, C.; An, Y.; Zhang, W.; Sun, C. Few-Shot Fine-Grained Image Classification: A Comprehensive Review. *AI* **2024**, *5*, 405–425. [CrossRef]
6. Wang, J.; Lu, J.; Yang, J.; Wang, M.; Zhang, W. An Unbiased Feature Estimation Network for Few-Shot Fine-Grained Image Classification. *Sensors* **2024**, *24*, 7737. [CrossRef]
7. Zhang, W.; Sun, C.; Gao, Y. Image intensity variation information for interest point detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 9883–9894. [CrossRef]
8. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, Ningbo, China, 9–12 July 2017; pp. 1126–1135.
9. Nichol, A. On first-order meta-learning algorithms. *arXiv* **2018**, arXiv:1803.02999.
10. Li, Z.; Zhou, F.; Chen, F.; Li, H. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv* **2017**, arXiv:1707.09835.
11. Zhang, W.; Zhao, Y.; Gao, Y.; Sun, C. Re-abstraction and perturbing support pair network for few-shot fine-grained image classification. *Pattern Recognit.* **2024**, *148*, 110158. [CrossRef]
12. Zhang, W.; Liu, X.; Xue, Z.; Gao, Y.; Sun, C. NDPNet: A novel non-linear data projection network for few-shot fine-grained image classification. *arXiv* **2021**, arXiv:2106.06988.
13. Pan, Z.; Yu, X.; Zhang, M.; Zhang, W.; Gao, Y. DyCR: A Dynamic Clustering and Recovering Network for Few-Shot Class-Incremental Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *138*, 2162–2388.
14. 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, 27 March 2018; pp. 1199–1208.
15. Khrulkov, V.; Mirvakhabova, L.; Ustinova, E. Hyperbolic image embeddings. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6418–6428.
16. Simon, C.; Koniusz, P.; Nock, R.; Harandi, M. Adaptive subspaces for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4136–4145.
17. Li, X.; Yang, X.; Ma, Z.; Xue, J.-H. Deep metric learning for few-shot image classification: A review of recent developments. *Pattern Recognit.* **2023**, *138*, 109381.
18. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4080–4090.

19. Wertheimer, D.; Tang, L.; Hariharan, B. Few-shot classification with feature map reconstruction networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8012–8021.
20. Wu, J.; Chang, D.; Sain, A.; Li, X.; Ma, Z.; Cao, J.; Jun, G.; Song, Y.Z. Bi-directional feature reconstruction network for fine-grained few-shot image classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 2821–2829.
21. Li, X.; Li, Z.; Xie, J. Self-reconstruction network for fine-grained few-shot classification. *Pattern Recognit.* **2024**, *153*, 110485. [CrossRef]
22. Zhang, Y.; Tang, H.; Jia, K. Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 233–248.
23. Zhang, M.; Wang, D.; Gai, S. Knowledge distillation for model-agnostic meta-learning. In Proceedings of the 24th European Conference on Artificial Intelligence, Santiago de Compostela, Spain, 29 August–8 September 2020; pp. 1355–1362.
24. Zhu, Y.; Liu, C.; Jiang, S. Multi-attention Meta Learning for Few-shot Fine-grained Image Recognition. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, Yokohama, Japan, 7–15 January 2020; pp. 1090–1096.
25. Li, X.; Wu, J.; Sun, Z.; Ma, Z.; Cao, J.; Xue, J.H. BSNet: Bi-similarity network for few-shot fine-grained image classification. *IEEE Trans. Image Process.* **2020**, *30*, 1318–1331.
26. Wei, X.S.; Wang, P.; Liu, L. Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples. *IEEE Trans. Image Process.* **2019**, *28*, 6116–6125. [CrossRef] [PubMed]
27. 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.
28. Vinyals, O.; Blundell, C.; Lillicrap, T. Matching networks for one shot learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3637–3645.
29. Afrasiyabi, A.; Larochelle, H.; Lalonde, J.F.; Gagné, C. Matching feature sets for few-shot image classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 2 April 2022; pp. 9014–9024.
30. Wang, D.; Ma, Q.; Zheng, Q.; Cheng, Y.; Zhang, T. Improved local-feature-based few-shot learning with Sinkhorn metrics. *Int. J. Mach. Learn. Cybern.* **2022**, *13*, 1099–1114.
31. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the ICML Deep learning Workshop, Lille, France, 6–11 July 2015; pp. 1–30.
32. Hoffer, E.; Ailon, N. Deep metric learning using triplet network. In Proceedings of the Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, 12–14 October 2015; Proceedings 3, 2015; pp. 84–92.
33. Jamal, M.A.; Qi, G.-J. Task agnostic meta-learning 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. 11719–11727.
34. Sun, Q.; Liu, Y.; Chua, T.-S.; Schiele, B. Meta-transfer learning 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. 403–412.
35. Shlens, J. A tutorial on principal component analysis. *arXiv* **2014**, arXiv:1404.1100.
36. Ringnér, M. What is principal component analysis? *Nat. Biotechnol.* **2008** *26*, 303–304.
37. Wah, C.; Branson, S.; Welinder, P. *The Caltech-Ucsd Birds-200-2011 Dataset*; California Institute of Technology: Pasadena, CA, USA, 2011.
38. Khosla, A.; Jayadevaprakash, N.; Yao, B. Novel dataset for fine-grained image categorization: Stanford dogs. In Proceedings of the CVPR Workshop on Fine-Grained Visual Categorization (FGVC), Colorado Springs, CO, USA, 20–25 June 2011; Volume 2.
39. Krause, J.; Stark, M.; Deng, J. 3d object representations for fine-grained categorization. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Washington, DC, USA, 2–8 December 2013; pp. 554–561.
40. Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; Vedaldi, A. Fine-grained visual classification of aircraft. *arXiv* **2013**, arXiv:1306.5151.
41. Zhang, C.; Cai, Y.; Lin, G.; Shen, C. Deepemd: Differentiable earth mover's distance for few-shot learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 5632–5648. [CrossRef]
42. 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, Seattle, WA, USA, 13 June 2021; pp. 8808–8817.
43. Chen, W.Y.; Liu, Y.C.; Kira, Z.; Wang, Y.C.F.; Huang, J.B. A closer look at few-shot classification. *arXiv* **2019**, arXiv:1904.04232.
44. Li, W.; Wang, L.; Xu, 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.
45. Wu, J.; Chang, D.; Sain, A.; Li, X.; Ma, Z.; Cao, J.; Guo, J.; Song, Y.-Z. Bi-directional ensemble feature reconstruction network for few-shot fine-grained classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 6082–6096. [CrossRef] [PubMed]
46. Tong, M.; Wang, S.; Xu, B.; Cao, Y.; Liu, M.; Hou, L.; Li, J. Learning from miscellaneous other-class words for few-shot named entity recognition. *arXiv* **2021**, arXiv:2106.15167.
47. Doersch, C.; Gupta, A.; Zisserman, A. Crosstransformers: Spatially-aware few-shot transfer. *Neural Inf. Process. Syst.* **2020**, *33*, 21981–21993.

48. 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, Montreal, QC, Canada, 28 February 2022; pp. 9041–9051.
49. Huang, H.; Wu, Z.; Li, W.; Huo, J.; Gao, Y. Local descriptor-based multi-prototype network for few-shot learning. *Pattern Recognit.* **2021**, *116*, 107935. [CrossRef]
50. Wu, Y.; Zhang, B.; Yu, G.; Zhang, W.; Wang, B.; Chen, T.; Fan, J. Object-aware long-short-range spatial alignment for few-shot fine-grained image classification. In Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China, 30 August 2021; pp. 107–115.
51. 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, Lisbon, Portugal, 10–14 October 2022; pp. 2135–2144.
52. Huang, H.; Zhang, J.; Yu, L.; Zhang, J.; Wu, Q.; Xu, C. TOAN: Target-oriented alignment network for fine-grained image categorization with few labeled samples. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 853–866.
53. Zha, Z.; Tang, H.; Sun, Y. Boosting few-shot fine-grained recognition with background suppression and foreground alignment. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 3947–3961. [CrossRef]
54. 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.
55. Huang, X.; Choi, S.H.; Kim, S. Attentive Pooling Network for Few-Shot Learning. *Multimed. Inf. Syst.* **2022**, *9*, 269–274.
56. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1–9.
57. Ma, Z.X.; Chen, Z.D.; Zhao, L.J. Cross-Layer and Cross-Sample Feature Optimization Network for Few-Shot Fine-Grained Image Classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; pp. 4136–4144.
58. Tang, L.; Wertheimer, D.; Hariharan, B. Revisiting pose-normalization for fine-grained few-shot recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 1 April 2020; pp. 14352–14361.
59. Sun, M.; Ma, W.; Liu, Y. Global and local feature interaction with vision transformer for few-shot image classification. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, 17 October 2022; pp. 4530–4534.
60. Dosovitskiy, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
61. Li, Y.; Bian, C. Few-shot fine-grained ship classification with a foreground-aware feature map reconstruction network. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5622812. [CrossRef]
62. Lee, S.; Moon, W.; Heo, J.P. Task discrepancy maximization for fine-grained few-shot classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5331–5340.
63. Li, X.; Song, Q.; Wu, J. Locally-enriched cross-reconstruction for few-shot fine-grained image classification. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 7530–7540.