

Article

FSCA: Few-Shot Learning via Embedding Adaptation with Corner Multi-Head Attention

Rui Xu ¹, Jitao Huang ², Yuqi Li ², Dianxin Dong ², Shuang Liu ³, Zijing Tian ³, Zhonghong Ou ^{3,*} and Meina Song ³

¹ Digital Department, China United Network Communications Co., Ltd., Beijing Branch, Beijing 100006, China; ruix7131@gmail.com

² Digital Department Data Operation Center, China United Network Communications Co., Ltd., Beijing Branch, Beijing 100006, China; ehuangjitao29@gmail.com (J.H.); liyuqi1566@sina.com (Y.L.); dianxindong@gmail.com (D.D.)

³ State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; lius2021@bupt.edu.cn (S.L.); tianzj@bupt.edu.cn (Z.T.); mnsong@bupt.edu.cn (M.S.)

* Correspondence: zhonghong.ou@bupt.edu.cn

Abstract: The demand for computer vision applications has been growing in recent years. However, most tasks in real-world scenarios are in the state of small amounts of data, and the discriminability of features results in low performance in few-shot settings. In order to obtain discriminability features in unknown classes more robustly, we first propose a simple and effective few-shot feature augmentation scheme called corner multi-head attention (CMA). The CMA module aims to project corner features through multi-head attention into a task-specific feature space through corner detection without sacrificing the generalization ability of the few-shot model. We then construct a representation space for few-shot classification by adaptively embedding the skeleton (EA). Finally, we proposed a few-shot classification framework, namely FSCA (Few-Shot Embedding Adaptation with Corner Multi-Head Attention). Its CMA module has low coupling, supports plug-and-play, and can easily be embedded into various models. To verify the effectiveness of the proposed scheme, we conducted experiments on three few-shot classification data sets, CUB-200-2011, miniImageNet, and tieredImageNet, and we performed a t-SNE clustering visual analysis. The experimental results show that it achieves continuous improvements over state-of-the-art methods on three widely used few-shot classification benchmarks. Specifically, on the miniImageNet data, the performance of FSCA under the five-way, one-shot and five-shot settings improved by an average of 1.6–4.5% and 2.2–5.4%, respectively.

Keywords: few-shot learning; embedding adaptation; corner multi-head attention; image classification



Academic Editor: Silvia Liberata Ullo

Received: 9 December 2024

Revised: 27 December 2024

Accepted: 30 December 2024

Published: 31 December 2024

Citation: Xu, R.; Huang, J.; Li, Y.; Dong, D.; Liu, S.; Tian, Z.; Ou, Z.; Song, M. FSCA: Few-Shot Learning via Embedding Adaptation with Corner Multi-Head Attention. *Electronics* **2025**, *14*, 130. <https://doi.org/10.3390/electronics14010130>

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, deep learning theories have made significant progress in the field of computer science, particularly in the practical application of computer vision. However, in the current fields of deep learning and image classification, a good learning algorithm often relies on massive data and a large amount of labeled training samples. Therefore, few-shot learning has garnered considerable attention in the field of computer vision. The goal of few-shot learning is to develop effective models that can generalize well from only a small amount of labeled samples. In tasks such as few-shot image classification, a typical setup involves a query set, which consists of new, unlabeled images, and a support set, containing a few labeled examples for each class. The task is to determine the class of each

query image based on the information provided by the support set, enabling the model to make accurate predictions despite limited training data. This ability to learn from just a few samples is crucial for real-world applications where acquiring large labeled data sets is often expensive or impractical.

Depending on whether the model training data contain a small amount of samples of the target category, it can be divided into zero-shot learning and few-shot learning. In zero-shot learning, the model training data do not contain samples of the target category. It is suitable for scenarios with many categories and in which it is difficult to collect comprehensive labeled data. It has a strong generalization ability. In few-shot learning, the model training data contain a small amount of samples of the target category. It is suitable for scenarios where a small amount of labeled data are available. The advantage is that it can learn quickly on a small amount of data.

Currently, most methods for addressing few-shot image classification are based on metric learning. The object of metric learning is usually the distance between the vectors of sample features. The purpose of metric learning is to reduce or limit the distance between samples of the same category and increase the distance between samples of different categories through training and learning. All of these methods are fitted into a metric function, and the category is determined based on the distance between the query set and the distribution of the support set. However, it is difficult to obtain a very tight clustering space from base-class training using only metric methods, and it is also impossible to learn a discriminable feature from the base class. On the one hand, in the case of a small sample data volume, the feature distribution of the cluster space cannot reflect the real category distribution. However, in the base class training stage, this type of method often only focuses on the characteristics of known classes, thereby ignoring the characteristic distribution of unknown classes in the testing stage.

In this context, we propose novel corner multi-head attention (CMA) to enhance the feature mining capability of few-shot classification. Firstly, a corner and edge extraction module is introduced to extract and construct edge and corner feature maps. The corner idea is inspired by character detection and recognition [1]. The foreground area of the image is often composed of these corner features. Similarly, for a class feature, each class basically has robust feature information, and class similarities can be determined by the degree of aggregation and boundaries of points. In the CMA module, in order to include it into the consideration of classification decision-making, we fuse these corner information and feature information through the multi-head attention mechanism [2] and combine it with feature adaptive embedding to make full use of these corner features as auxiliary information. In addition, to adapt to the irrelevance of different tasks in small samples, we adopt an adaptive embedding mechanism, which is called the EA module.

Our main contributions are as follows:

- We propose the CMA module and introduce edge and corner maps to enhance the feature extraction capabilities of small samples.
- The CMA module has low coupling and plug-and-play features, and it can easily be integrated into other models.
- Our model combines CMA and embedding adaptive (EA) functionality to achieve good performance on three few-shot benchmark data sets.

2. Related Works

Few-shot classification involves classifying query samples based on only a few labeled examples (usually no more than 10) per class, with the test set containing unknown classes. The main approaches to few-shot classification can be broadly categorized into two types: optimization-based and metric-based methods.

Optimization-based methods update model parameters online to quickly optimize few-shot learning tasks. Among them, MAML [3] obtains a parameter initialization state with higher sensitivity through task training. Reptile [4] improves MAML and is not limited to single task direction but is limited to multi-directional task updates. MetaNAS [5] introduces a neural architecture search (NAS) into the few-shot learning algorithm and simultaneously creates initialization parameters and network structure parameters. The MetaOptNet [6] method proposes that the nearest neighbor classification method has weak discriminability. For few-shot classification tasks, it can be learned as a linear classifier and planned as a convex learning problem. This is mainly done by solving how to quickly update the model online with a small number of support samples [7–10].

Metric-based approaches learn similarity metric representation using metric function learning. MatchNet [11] uses a neural network with an attention mechanism and a memory module to solve the problem of ordinary nearest neighbor algorithms over-relying on metric functions. ProtoNet [12] directly uses the mean value of each category feature vector as the prototype corresponding to that category and uses Euclidean distance for determination. DeepEMD [13] introduces the Earth mover's distance measurement method to calculate the distance by finding the best matching method between each tile. FEAT [14] has developed a method to embed instances into adaptive target classification tasks using set-to-set functions, converting task-irrelevant feature information into task-related feature information. These methods also use predefined metrics to learn deep representations [15–18]. In addition, there are local descriptor-based metrics [19] and graph neural network methods [20].

There is also a lot of work to further improve performance by leveraging the combination of pre-training and meta-training [21,22]. Based on the usability and effectiveness of the method above, we selected FEAT [14] as the baseline. In this paper, we use metric learning to solve the few-shot learning problem using corner features combined with a multi-head attention mechanism and embedding adaptive modules, which is the FSCA framework.

3. Method

In this section, we introduce corner multi-head attention (CMA) to learn robust feature expression capabilities. Figure 1 illustrates the overall architecture.

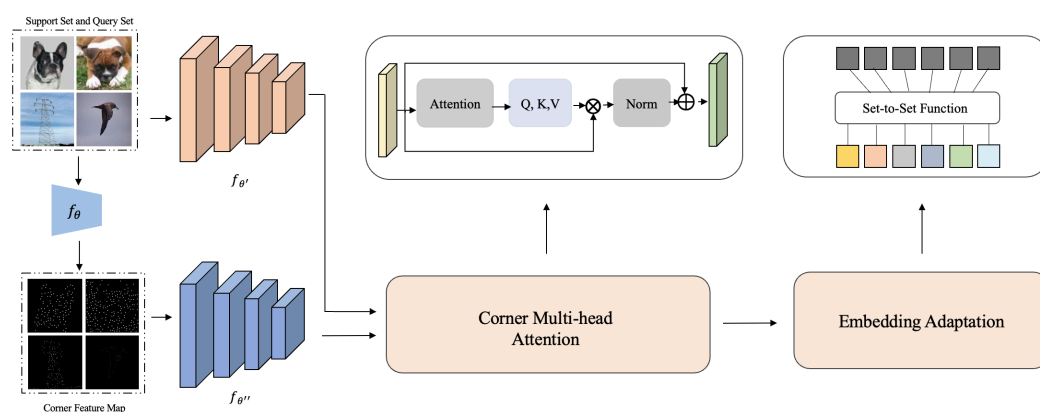


Figure 1. The framework of the proposed FSCA approach.

3.1. Problem Formulation

In few-shot classification tasks, meta-learning training methods are usually used, and they are mainly divided into two stages: meta-training and meta-testing. In the meta-learning method, both the original training data set and the test data set must undergo special processing. During the training process, two data sets are used: the support set,

S , and the query set, Q , and the same is true for the testing process. Few-shot learning (FSL) is usually divided into N -way K -shot problems, where N refers to how many types of samples are used in the meta-testing process, and K refers to how many samples there are in each category in the meta-testing process.

The N -way K -shot data set is divided by randomly selecting N categories from which there are many samples and then randomly selecting $K + X$ samples from each category from these N categories; X can be any number, but it must not exceed the maximum number of samples for each category. The selected K samples are called the support set, $S = \{(x_i^s, y_i^s)\}_{i=1}^{n \cdot k}$, and the X samples are called the query set, $Q = \{(x_i^q, y_i^q)\}_{i=1}^{n \cdot x}$. Among them, (x_i^s, y_i^s) represents the i -th image x_i^s in the support set, S , and the corresponding label y_i^s , (x_i^q, y_i^q) represents the i -th image x_i^q in the query set, Q , and the corresponding label y_i^q . During meta-training, a model for learning the mapping from S to Q is trained at each epoch iteration. At meta-testing, the model uses the learned mapping to assign the example I^q in Q to one of N categories from the sampled support set S .

3.2. Architecture Overview

In an epoch of iterative training, given a pair of support set I_s and query set I_q , the feature representation F_b is obtained through the backbone, the corner feature map (CFM) is obtained through the corner extractor, and F_c is obtained after the linear transformation of the features. The CMA module generates the joint attention feature, A_d , in a feature fusion manner, inputs it into the adaptive embedding module, and uses the similarity score in the metric function to query the one where I_q is closest to the embedding category. The overall architecture diagram is detailed in Figure 1.

3.3. Corner Multi-Head Attention (CMA)

The CFM module is used to extract corner feature maps. The CMA module takes the corner extractor feature basic representation F_s and converts it into a suitable form F_c , as well as the feature extractor feature F_b , to prepare reliable input for the CMA module to analyze feature correlations between images. Figures 2 and 3 show the architecture of the CFM module and CMA module, respectively.

Corner feature map (CFM). This map is generated via a corner detector, mainly using the Shi–Tomasi corner detector [23], which can produce higher-quality corner information. For each pixel $\{(x, y)\}$ in each image and the image tensor S , the angular response function expression is

$$f(\theta) = \min(r_1, r_2) > \beta \quad (1)$$

Among them, r_1 and r_2 are the eigenvalues of S . If the function $f(\theta)$ is greater than a certain threshold, β , the pixel $\{(x, y)\}$ is a corner point, and its point position is mapped to 1; otherwise, it is 0. That is, the corner point map is a sparse matrix, and the position information of the corner point exists at the matrix mapping position 1. After the function $f(\theta)$ is applied, a one-dimensional vector $F_s = (N, 1, H, W)$ is obtained. In order to comply with the feature vector fusion method, a three-layer Concat connection function is used to obtain $F_{s'} = (N, 3, H, W)$, and then a linear transformation Linner is used to obtain $F_c = (N, C)$.

The schematic diagram of corner feature map calculation is shown in Figure 2. The corner point map extracted via the CFM module is shown in Figure 4.

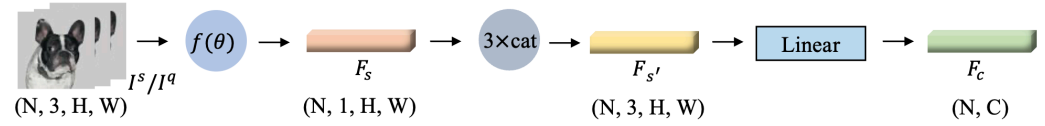


Figure 2. Schematic illustration of corner feature map (CFM).

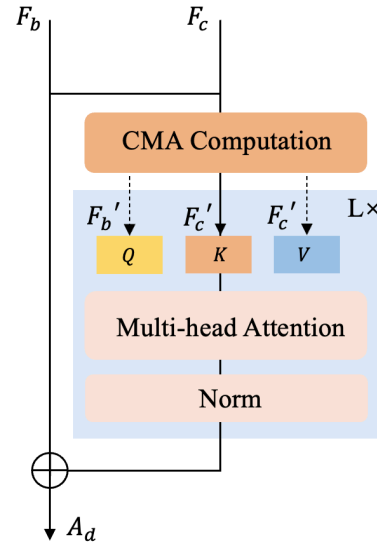


Figure 3. CMA computation.

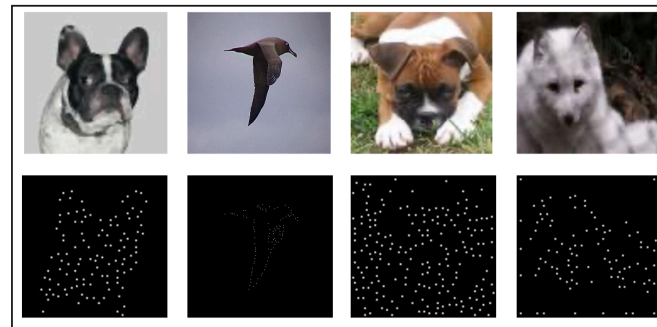


Figure 4. Corner feature extraction map.

CMA computation. This module mainly models the cross relationship between image features and corner features. The core part uses the attention mechanism. The formula of the attention mechanism is as follows:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V = Softmax\left(\frac{AS}{\sqrt{d}}\right)V \quad (2)$$

where Q is a matrix of size (n_q, d_q) , and K is a matrix of size (n_k, d_k) , $d_q = d_k = d$, which refers to the dimension of the key or query. The dot product result is the attention score matrix AS , which is of size (n_q, n_k) . Therefore, $AS(i, j)$ is understood as the attention score of the i -th query on the j -th key. The larger this value is, the greater the association between the i -th query and the j -th key. Of course, to prevent the attention value from being too large and to stabilize the training process, the attention mechanism usually scales this result by dividing it by $\sqrt{d_k}$. And the softmax function normalizes the attention scores of each query for all keys. Finally, it is multiplied by the weighted sum of the V matrix, and $Attention(i,)$

represents the value queried by the i -th query. In order to enhance the model's attention to different features, the attention mechanism is extended to multiple heads:

$$\text{MultiHeadAttention}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_H) \cdot W^O \quad (3)$$

where W^O is the weight matrix at the output, and H is the number of heads. It includes three vectors: a query vector, a key vector, and a value vector. For a given query vector, multi-head attention performs a weighted sum of key vectors. The weight is calculated by the similarity between the query vector and the key vector, and then the resulting weighted sum is multiplied by the value vector for the output [2,24–26]. In the multi-head mechanism, the input sequence is split into multiple heads, with each head performing independent computations. Each head generates a separate output, which is then combined to form the final result. This approach allows the model to capture different aspects of the input data simultaneously, enhancing its ability to learn diverse representations and improve overall performance. And the normalization layer is used to normalize the input features before or after applying the multi-head attention mechanism. It is used to normalize the input features to reduce the internal covariate shift problem in model training.

Then, the image features and corner features are added using the residual formula:

$$A_d = g(F_c) + F_b \quad (4)$$

It strengthens image-level features by enhancing features to help few-shot learners better understand “what to learn and distinguish” in images. Our experiments show that CMA is robust to intra-class invariant features and helps generalize to unseen classes.

3.4. Learning Embedding Adaptation

In this subsection, we derive adaptive embeddings for few-shot learning and then describe the learning objectives of our approach (Algorithm 1).

Algorithm 1: Few-Shot Learning via Embedding Adaptation with CMA

Input: Few-shot training set: $D = \{D_1, D_2, \dots, D_N\}$. The loss weight hyperparameters λ . For each $D_i = (S_k, Q_k) = \{\{(x_1^s, y_1^s), (x_2^s, y_2^s), \dots, (x_n^s, y_n^s)\}, \{(x_1^q, y_1^q), (x_2^q, y_2^q), \dots, (x_n^q, y_n^q)\}\}$

Output: learn Φ

- 1: **for** D_i in $\{D_1, D_2, \dots, D_N\}$ **do**
- 2: Extract S_i and Q_i
- 3: Calculate corner feature map: f_s by Equation (1)
- 4: Calculate the intersection feature map of image features and corner features: A_d by Equation (2)–(4)
- 5: Calculate task embedding by Equation (5)
- 6: Calculate few-shot classifier loss: L_{FSL} by Equations (6) and (7)
- 7: Calculate the total loss function: L by Equation (8)
- 8: **end for**
- 9: **return** Φ

Embedding adaptation (EA) for FSL. The main learning is a set-to-set function, rather than the traditional instance function, which contextualizes the image instances of the set. We adopt the set-to-set adaptive embedding method using a transformer in the FEAT [14] framework. It can be adapted to different types of task-independent embedding functions and computed using similarity measures.

Learning objective. The main goal is to learn a learnable parameter, Φ , through the meta-training process.

Firstly, we compute a prototype, h_c , for category K in support sets S as follows:

$$h_c = \frac{1}{k} \cdot \sum_{(x_i, y_i) \in S} f_\varphi(x_i) \cdot \mathbb{I}(y_i = c) \quad (5)$$

where f_φ is a feature extractor with learnable parameters, φ . Then, based on the prototype method, the probability distribution of the data point x_i in the query set Q_r on C classes is defined as follows:

$$P(y = k|h_i) = \frac{\exp(-d(h_i, c_r))}{\sum_{j=1}^C \exp(-d(h_i, c_r))} \quad (6)$$

where $d(\cdot, \cdot)$ represents the cosine distance. Therefore, the loss of the nearest centroid classifier on an episode can be calculated as follows:

$$L_{FSL} = \frac{1}{|Q_e|} \sum_{(h_i, y_i) \in Q_e} -\log P(y = k|h_i) \quad (7)$$

We adopt a joint training approach to improve the proposed module and backbone network by combining two losses: the cross-entropy-based classification loss L_{CE} and the embedding space-based classification loss L_{FSL} .

$$L = L_{CE} + \lambda L_{FSL} \quad (8)$$

where L is total loss, and $\lambda = 1$ is set by default.

4. Experiments

In this section, we evaluate FSCA on several standard few-shot classification benchmark data sets and compare the results with those of mainstream models.

4.1. Data Sets

For evaluation, we use three standard baselines of few-shot classification.

CUB-200-2011: It has a total of 117,888 bird images, including 200 bird subcategories, with 5994 images in the training data set and 5794 images in the test set. And the train/validation/test splits consist of 100/50/50 object classes, respectively.

miniImageNet: Its data set is excerpted from the ImageNet data set, which is a benchmark data set in the field of meta-learning and few-shot learning. miniImageNet contains a total of 60,000 color images in 100 categories, with 600 samples in each category. The train/validation/test splits consist of 64/14/20 object classes, respectively.

tieredImageNet: It is also a subset of ILSVRC-12, but the data volume is larger, with a total of 608 subcategories. Each subcategory contains varying numbers of images, totaling 779,165 images. The train/validation/test splits consist of 351/97/160 object classes, respectively.

4.2. Implementation Details

All experiments are implemented using Pytorch, running on two NVIDIA 2080Ti GPUs and one 3090 GPU.

Pre-training. Similar to the works [6,9,10,14,27], we train the classifier using cross-entropy loss on all base classes of the training set and then remove the final fully connected (FC) layer to obtain the embedding module. The few-shot classification performance of the embedding module is evaluated using the cosine distance, and the best pre-trained model is selected based on these results.

Experiment-setting. In meta-training/meta-testing, we use ResNet12 as the backbone. We follow the same process in FEAT [14] to pretrain the adaptive embedding (EA) backbone network. Most hyperparameter settings remain unchanged and require no adjustment. We use a batch size of 20 and a learning rate of 0.0005, decaying with a cosine schedule, setting the multi-head attention head to 8 and selecting the number of layers as 1/3/5/6/8, and we show three sets of optimal results. During the meta-testing, we randomly sample 2000 snippets from the test set and report the average classification top-one accuracy with a 95% confidence interval. We experiment with our method in five-way, one-shot and five-way, five-shot settings.

4.3. Comparison to Existing Methods

According to the standard settings, we conducted experiments on miniImageNet, tieredImageNet, and CUB-200-2011, and the classification results of our FSCA method and other few-shot method are shown in Tables 1–3.

Table 1. Results on miniImageNet dataset.

Method	Backbone	Five-Way, One-Shot	Five-Way, Five-Shot
TADAM [28]	ResNet12	58.50 ± 0.30	76.70 ± 0.38
MetaOptNet [6]	ResNet12	62.64 ± 0.61	78.63 ± 0.46
MTL [10]	ResNet12	61.20 ± 1.80	75.50 ± 0.80
CAN [18]	ResNet12	63.85 ± 0.48	79.44 ± 0.34
ProtoNet [12]	ResNet12	62.39 ± 0.21	80.53 ± 0.14
QSFormer [29]	ResNet12	65.24 ± 0.28	79.96 ± 0.20
DeepEMD [13]	ResNet12	65.91 ± 0.82	82.41 ± 0.56
FEAT [14]	ResNet12	66.78 ± 0.20	82.05 ± 0.14
IEPT [30]	ResNet12	67.05 ± 0.44	82.90 ± 0.30
FRN [31]	ResNet12	66.45 ± 0.19	82.83 ± 0.13
MABAS [32]	ResNet12	65.08 ± 0.86	82.70 ± 0.54
FSCA (Our, L = 1)	ResNet12	67.93 ± 0.31	84.67 ± 0.28
FSCA (Our, L = 3)	ResNet12	68.27 ± 0.29	84.74 ± 0.27
FSCA (Our, L = 6)	ResNet12	68.62 ± 0.35	85.03 ± 0.28

Table 2. Results on tieredImageNet dataset.

Method	Backbone	Five-Way, One-Shot	Five-Way, Five-Shot
MetaOptNet [6]	ResNet12	65.99 ± 0.72	81.56 ± 0.63
MTL [10]	ResNet12	65.62 ± 1.80	80.61 ± 0.90
CAN [18]	ResNet12	69.89 ± 0.51	84.23 ± 0.37
AM3 [33]	ResNet12	67.23 ± 0.34	78.95 ± 0.22
ProtoNet [12]	ResNet12	68.23 ± 0.23	84.03 ± 0.16
Distill [34]	ResNet12	71.52 ± 0.69	86.03 ± 0.49
DeepEMD [13]	ResNet12	71.16 ± 0.87	86.03 ± 0.58
FEAT [14]	ResNet12	70.80 ± 0.23	84.79 ± 0.16
IEPT [30]	ResNet12	72.24 ± 0.50	86.73 ± 0.34
FRN [31]	ResNet12	71.16 ± 0.22	86.01 ± 0.15
QSFormer [29]	ResNet12	72.47 ± 0.31	85.43 ± 0.22
FSCA (Our, L = 1)	ResNet12	71.82 ± 0.25	84.92 ± 0.31
FSCA (Our, L = 3)	ResNet12	72.33 ± 0.28	86.07 ± 0.27
FSCA (Our, L = 6)	ResNet12	73.12 ± 0.29	87.21 ± 0.26

Table 3. Results on CUB-200-2011 data set.

Method	Backbone	Five-Way, One-Shot	Five-Way, Five-Shot
ProtoNet [12]	ResNet12	66.09 $\pm$ 0.92	82.50 $\pm$ 0.58
RelationNet [16]	ResNet34	66.20 $\pm$ 0.99	82.30 $\pm$ 0.58
MAML [3]	ResNet34	67.28 $\pm$ 1.08	83.47 $\pm$ 0.59
cosine classifier [35]	ResNet12	67.30 $\pm$ 0.86	84.75 $\pm$ 0.60
MatchNet [11]	ResNet12	71.87 $\pm$ 0.85	85.08 $\pm$ 0.57
S2M2 [36]	ResNet34	72.92 $\pm$ 0.83	86.55 $\pm$ 0.51
FEAT [14]	ResNet12	73.27 $\pm$ 0.22	85.77 $\pm$ 0.14
DeepEMD [13]	ResNet12	75.65 $\pm$ 0.83	88.69 $\pm$ 0.50
FSCA (Our, L = 1)	ResNet12	75.06 $\pm$ 0.22	86.36 $\pm$ 0.31
FSCA (Our, L = 3)	ResNet12	76.01 $\pm$ 0.21	88.43 $\pm$ 0.25
FSCA (Our, L = 6)	ResNet12	76.97 $\pm$ 0.28	89.89 $\pm$ 0.32

Through extensive experiments on three benchmark data sets, we clearly validate the performance of FSCA in few-shot image classification. Our method is a metric learning approach. Traditional metric-based methods typically extract features for support and query samples independently, which can result in the model focusing on irrelevant or non-target regions of the image. In contrast, our CMA actively emphasizes the target object regions, enabling the model to capture more relevant and discriminative features. This targeted feature extraction improves the model's ability to distinguish between classes, leading to better performance in few-shot classification tasks. By focusing on the most important parts of the image, our method enhances the overall feature representation, which is crucial for tasks with limited labeled data.

Compared with ProtoNet [12], embedding the CMA module achieves 7.22% higher performance on the CUB-200-211 data set five-shot learning, which proves the superiority of our CMA module. Furthermore, our algorithm can easily be applied to other few-shot learning models, such as matching networks [11], FEAT networks [14], and relational networks [16]. The embedding results are shown in Table 4.

Table 4. Results from embedding the CMA module into other models on CUB-200-211 data set.

Method	Five-Way, Five-Shot	Method	Five-Way, Five-Shot
MatchNet [11]	85.08 $\pm$ 0.57	MatchNet + CMA	86.54 $\pm$ 0.30
ProtoNet [12]	82.50 $\pm$ 0.58	ProtoNet + CMA	89.72 $\pm$ 0.26
FEAT [14]	85.77 $\pm$ 0.14	FEAT + CMA	89.16 $\pm$ 0.31

4.4. Combined with Other Methods

The core ideas of the CMA mechanism mainly comprise the following three points:

Target area focus: One of the fundamental goals of CMA is to direct the network's attention to key target areas within the feature space, which may include specific parts of objects, corners, or other highly informative regions. This is especially important in few-shot learning, in which models often struggle due to the limited amount of data available for training. By concentrating on these critical areas, the CMA mechanism helps the model become more sensitive to distinguishing features, leading to better recognition of the target category.

In the context of image classification, this means that CMA helps the model focus on object parts like corners or textures, which are typically more discriminative and provide stronger cues for classification. In some cases, objects might be partially occluded or may

appear in novel configurations, and focusing on key areas ensures that the model is more robust to such variations.

Adaptive weighting: Through the multi-head attention mechanism, the CMA module can assign different weights to each attention head, thereby providing more personalized and accurate feature enhancement for samples in the query set based on the examples in the support set. Adaptive weighting enables the model to not only focus on the most informative features but also handle variations across different samples in the query set. In cases where the support set provides limited examples, this adaptability allows the model to generalize better and more efficiently from fewer examples.

Information interaction: The CMA module realizes information interaction between the support set and the query set, which enables the model to better identify and distinguish new categories in the case of few samples.

In addition, the CMA module has a low degree of coupling, supports plug-and-play, and can easily be embedded in various models. The combination with other models is mainly used in the feature extraction layer of the algorithm. The CMA module is added during feature extraction to perform feature fusion and embedding. After embedding, the fused features are input into subsequent classifiers or other modules.

5. Ablation Study

To evaluate the impact of each core module in our FSCA model, we conducted thorough ablation studies by removing one module at a time. We then compared the performance of the modified model against the full model in a five-way, five-shot setting. These experiments help identify the contribution of each module to the overall performance, providing insights into which components are most crucial for achieving optimal results in few-shot classification tasks.

Influence on CMA and EA. Table 5 reports the ablation analysis of the individual components of CMA and EA. We observe that both CMA and EA consistently improve classification accuracy. As shown in the first and second rows, CMA improves the baseline by 3.5%. This large gain confirms that identifying discriminative features for each category is crucial for few-shot classification. In addition, although the improvement effect of CMA is slightly lower than that of EA, CMA is proven to be effective for all scenarios. What's more, they can be combined to further boost performance. It has been verified that the two sub-modules are complementary according to the second, third, and fourth rows.

Integration method. We also studied the effects of using different fusion methods for CFM components, as described in Table 6. As shown in the second and third rows, both fusion methods can improve performance and be able to represent the discriminative information of objects. However, the direct feature splicing method has its limitations. It cannot fuse features well for corner point sparse images to enhance the discriminability information. The CMA method we proposed has a more outstanding ability to improve the feature identification effect.

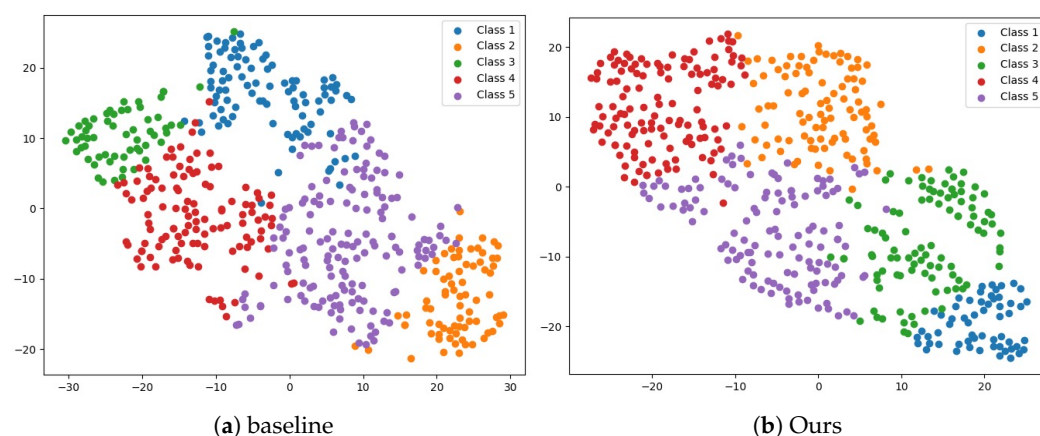
Table 5. The effects of CMA and EA.

CMA	EA	Five-Way, One-Shot	Five-Way, Five-Shot
×	×	61.23 ± 0.18	79.61 ± 0.13
✓	×	64.73 ± 0.26	82.43 ± 0.36
×	✓	65.16 ± 0.18	82.81 ± 0.41
✓	✓	68.62 ± 0.35	85.03 ± 0.28

Table 6. The effects of integrate function.

CFM	Integrate	Five-Way, One-Shot	Five-Way, Five-Shot
×	-	65.16 ± 0.18	82.81 ± 0.41
✓	Concat	65.39 ± 0.24	83.01 ± 0.27
✓	CMA	68.62 ± 0.35	85.03 ± 0.28

Visualization analysis. We provide visualizations to validate the distinctiveness of the representations generated via our framework for novel classes. Specifically, we randomly select five classes, each containing 100 images, from the meta-test set of miniImageNet. We then generate embeddings for all images using both the baseline and our framework (baseline + our method). We then use t-SNE [37] to project these embeddings into 2D space, as shown in Figure 5. The embedding distribution obtained through the baseline + our framework is more separable, with more precise and compact class boundaries between classes (see Figure 5a vs. Figure 5b). Visualization results demonstrate that our proposed framework produces more distinct and separable representations for new categories. This improved discriminability allows the model to better differentiate between classes, even with limited training data, leading to more accurate classifications of previously unseen categories. These results highlight the effectiveness of our framework in enhancing the ability of the model to generalize to new, unseen classes.

**Figure 5.** t-SNE visualization of 100 randomly sampled images for each of the five meta-test classes in miniImageNet.

6. Conclusions

In this work, we have proposed an effective feature extraction and fusion scheme to enhance robustness and discriminability features in new classes with few samples. Based on the above work, we construct an FSCA framework that utilizes a multi-head attention mechanism to fuse corner information (CMA) and adaptive feature embedding (EA). Three data sets are empirically analyzed and validated under our framework. In the future, we plan to extend our approach to tasks beyond classification, including object detection and semantic segmentation.

Author Contributions: Methodology, Y.L. and D.D.; Formal analysis, Z.T.; Investigation, S.L.; Writing—original draft, R.X.; Writing—review & editing, J.H.; Visualization, M.S.; Supervision, Z.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 62076035 and supported by the Project of China United Network Communications Co., Ltd. Beijing Branch.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Author Rui Xu was employed by the company China United Network Communications Co., Ltd., Beijing Branch. Author Jitao Huang, Yuqi Li and Dianxin Dong were employed by the company China United Network Communications Co., Ltd., Beijing Branch. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

Abbreviations	Full Form
FSCA	Few-Shot Learning via Embedding Adaptation with Corner Multi-Head Attention
FSL	Few-Shot Learning
CMA	Corner Multi-Head Attention
CFM	Corner Feature Map
EA	Embedding Adaptation
FC	Fully Connected

References

1. Xie, X.; Fu, L.; Zhang, Z.; Wang, Z.; Bai, X. Toward Understanding WordArt: Corner-Guided Transformer for Scene Text Recognition. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 303–321.
2. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
3. Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv* **2017**, arXiv:1703.03400.
4. Nichol, A.; John, S. Reptile: A scalable meta-learning algorithm. *arXiv* **2018**, arXiv:1803.02999.
5. Elsken, T.; Staffler, B.; Metzen, J.H.; Hutter, F. Meta-Learning of Neural Architectures for Few-Shot Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 13–19 June 2020; pp. 12362–12372.
6. Lee, K.; Maji, S.; Ravichandran, A.; Soatto, S. Meta-Learning with Differentiable Convex Optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 15–19 June 2019; pp. 10657–10665.
7. Ravi, S.; Larochelle, H. Optimization as a Model for Few-Shot Learning. In *Proceedings of the International Conference on Learning Representations*, Toulon, France, 24–26 April 2017.
8. Yoon, J.; Kim, T.; Dia, O.; Kim, S.; Bengio, Y.; Ahn, S. Bayesian Model-Agnostic Meta-Learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, Montréal, QC, Canada, 3–8 December 2018; pp. 7343–7353.
9. Rusu, A.A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; Hadsell, R. Meta-Learning with Latent Embedding Optimization. In *Proceedings of the International Conference on Learning Representations*, New Orleans, LA, USA, 6–9 May 2019.
10. Sun, Q.; Liu, Y.; Chua, T.; Schiele, B. Meta-Transfer Learning for Few-Shot Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 15–20 June 2019; pp. 403–412.
11. Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; Wierstra, D. Matching Networks for One Shot Learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, Barcelona, Spain, 5–10 December 2016; pp. 3630–3638.
12. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical Networks for Few-shot Learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 4–9 December 2017; pp. 4077–4087.
13. 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 Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 13–19 June 2020.
14. Ye, H.; Hu, H.; Zhan, D.; Sha, F. Few-Shot Learning via Embedding Adaptation with Set-to-Set Functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 13–19 June 2020; pp. 8805–8814.
15. Khrulkov, V.; Mirvakhabova, L.; Ustinova, E.; Oseledets, I.V.; Lempitsky, V.S. Hyperbolic Image Embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 13–19 June 2020; pp. 6417–6427.
16. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.S.; 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, 18–22 June 2018; pp. 1199–1208.

17. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the International Conference on Machine Learning Deep Learning Workshop, Lille, France, 6–11 July 2015; pp. 1–30.
18. Hou, R.; Chang, H.; Ma, B.; Shan, S.; Chen, X. Cross Attention Network for Few-shot Classification. In Proceedings of the Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 4005–4016.
19. 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 Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7260–7268.
20. Satorras, V.G.; Estrach, J.B. Few-Shot Learning with Graph Neural Networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
21. Chen, Y.; Liu, Z.; Xu, H.; Darrell, T.; Wang, X. Meta-Baseline: Exploring Simple Meta-Learning for Few-Shot Learning. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 9042–9051.
22. Liu, C.; Fu, Y.; Xu, C.; Yang, S.; Li, J.; Wang, C.; Zhang, L. Learning a Few-shot Embedding Model with Contrastive Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; pp. 8635–8643.
23. Shi, J.; Tomasi, C. Good features to track. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600.
24. Xiong, R.; Yang, Y.; He, D.; Zheng, K.; Zheng, S.; Xing, C.; Zhang, H.; Lan, Y.; Wang, L.; Liu, T. On layer normalization in the transformer architecture. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 10524–10533.
25. Bahdanau, D. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
26. Lin, Z.; Feng, M.; Santos, C.N.d.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A structured self-attentive sentence embedding. *arXiv* **2017**, arXiv:1703.03130.
27. Qiao, S.; Liu, C.; Shen, W.; Yuille, A.L. Few-Shot Image Recognition by Predicting Parameters From Activations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7229–7238.
28. Oreshkin, B.N.; López, P.R.; Lacoste, A. TADAM: Task dependent adaptive metric for improved few-shot learning. In Proceedings of the Annual Conference on Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; pp. 719–729.
29. Wang, X.; Wang, X.; Jiang, B.; Luo, B. Few-Shot Learning Meets Transformer: Unified Query-Support Transformers for Few-Shot Classification. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 7789–7802. [[CrossRef](#)]
30. Zhang, M.; Zhang, J.; Lu, Z.; Xiang, T.; Ding, M.; Huang, S. IEPT: Instance-Level and Episode-Level Pretext Tasks for Few-Shot Learning. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 3–7 May 2021.
31. 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.
32. Kim, J.; Kim, H.; Kim, G. Model-Agnostic Boundary-Adversarial Sampling for Test-Time Generalization in Few-Shot Learning. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 599–617.
33. Xing, C.; Rostamzadeh, N.; Oreshkin, B.N.; Pinheiro, P.O. Adaptive Cross-Modal Few-shot Learning. In Proceedings of the Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 4848–4858.
34. Tian, Y.; Wang, Y.; Krishnan, D.; Tenenbaum, J.B.; Isola, P. Rethinking Few-Shot Image Classification: A Good Embedding is All You Need? In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 266–282.
35. Chen, W.; Liu, Y.; Kira, Z.; Wang, Y.F.; Huang, J. A Closer Look at Few-shot Classification. In Proceedings of the ICLR, New Orleans, LA, USA, 6–9 May 2019.
36. Mangla, P.; Singh, M.; Sinha, A.; Kumari, N.; Balasubramanian, V.N.; Krishnamurthy, B. Charting the Right Manifold: Manifold Mixup for Few-shot Learning. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 2207–2216.
37. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.