electronics

MDPI

Article

Few-Shot Image Classification Based on Swin Transformer + CSAM + EMD

Huadong Sun [1,2], Pengyi Zhang [1,*], Xu Zhang [1,2] and Xiaowei Han [1,2]

[1] School of Computer and Information Engineering, Harbin University of Commerce, Harbin 150028, China; 102603@hrbcu.edu.cn (H.S.); 103044@hrbcu.edu.cn (X.Z.); hanxiaowei2017@hrbcu.edu.cn (X.H.)
[2] Heilongjiang Provincial Key Laboratory of Electronic Commerce and Information Processing, Harbin 150028, China
* Correspondence: zpy@s.hrbcu.edu.cn

Abstract: In few-shot image classification (FSIC), the feature extraction module of the traditional convolutional neural networks is often constrained by the local nature of the convolutional kernel. As a result, it becomes challenging to handle global information and long-distance dependencies effectively. In order to address this problem, an innovative FSIC method is proposed in this paper, which is the integration of Swin Transformer and CSAM and Earth Mover's Distance (EMD) technology (STCE). We utilize the Swin Transformer network for image feature extraction, and perform CSAM attention mechanism feature weighting on the output feature map, while we adopt the EMD algorithm to generate the optimal matching flow between the structural units, minimizing the matching cost. This approach allows for a more precise representation of the classification distance between images. We have conducted numerous experiments to validate the effectiveness of our algorithm. On three commonly used few-shot datasets, namely mini-ImageNet, tiered-ImageNet, and FC100, the accuracy of one-shot and five-shot has reached the state of the art (SOTA) in the FSIC; the mini-ImageNet achieves an accuracy of 98.65 $\pm$ 0.1% for one-shot and 99.6 $\pm$ 0.2% for five-shot tasks, while tiered ImageNet has an accuracy of 91.6 $\pm$ 0.1% for one-shot tasks and 96.55 $\pm$ 0.27% for five-shot tasks. For FC100, the accuracy is 64.1 $\pm$ 0.3% for one-shot tasks and 79.8 $\pm$ 0.69% for five-shot tasks. On two commonly used few-shot datasets, namely CUB, CIFAR-FS, CUB achieves an accuracy of 83.1 $\pm$ 0.4% for one-shot and 92.88 $\pm$ 0.4% for five-shot tasks, while CIFAR-FS achieves an accuracy of 86.95 $\pm$ 0.2% for one-shot and 94 $\pm$ 0.4% for five-shot tasks.

Keywords: few-shot learning; image classification; Swin Transformer; EMD

Citation: Sun, H.; Zhang, P.; Zhang, X.; Han, X. Few-Shot Image Classification Based on Swin Transformer + CSAM + EMD. *Electronics* **2024**, *13*, 2121. https://doi.org/10.3390/electronics13112121

Academic Editor: Manohar Das

Received: 10 May 2024
Revised: 27 May 2024
Accepted: 28 May 2024
Published: 29 May 2024

1. Introduction

In the age of abundant data, deep learning algorithms have demonstrated remarkable outcomes in numerous domains associated with visual computing [1,2]. However, the deep learning model's ability to achieve high accuracy is often dependent on the availability of a significant amount of training data and a lot of manual labeling. In order to save these costs, people think about how to accurately classify objects with little data. For instance, if a young child has never seen a rabbit, give him a card with a rabbit on it. When he sees the real rabbit, he will immediately recall it in his mind and recognize it. Even if the rabbit's body size, hair color, and body posture are quite different from the pictures he has seen, he can accurately identify it. Inspired by human learning views, the concept of FSL is put forward. The key problem of FSL is that there is a scarcity of labeled data, and it relies on data augmentation [3–6]. This method expands the dataset by generating data in various ways, thereby addressing the issue of data sparseness in the FSL process.

In recent years, FSL has made rapid developments. The FSL method based on metric-based learning [7–10] partially addresses the issue of data scarcity. It directly measures the distance between test images and training images rather than relying on large datasets. However, when confronted with complex situations, such as a cluttered background, high

similarity between categories, and significant differences within categories, this method may result in a significant distance between images of the same category in the embedded space after feature extraction. As a result, the accuracy of image classification is inevitably reduced. At present, the existing FSIC methods based on metric learning usually rely on direct image comparison. However, they often overlook the significance of local image features. In fact, different parts of an image may have varying levels of importance; so, we require a more adaptable metric learning approach. Specifically, Zhang et al. [11] assign less weight to the classification features that contribute less overall, while assigning greater weight to the regions that contain rich image features and advanced semantics. This method of weight distribution is more aligned with the actual situation. In this study, the problem of FSL is formalized as an optimal matching problem, and the EMD is used to measure the learning idea. A measure function is used to calculate the structural distance between test images. Finally, we utilize these distances to make predictions for image classification. This allows us to construct a classifier that can efficiently and precisely determine the category, particularly for datasets with limited examples.

The main contributions of this paper are as follows:

(1) The Swin Transformer is employed for feature extraction, enabling the acquisition of both local and global details from the image are captured and perform CSAM attention mechanism feature weighting on the output feature map.
(2) The EMD measurement module is employed to measure the distance. The main concept involves utilizing block level measurement and incorporating a cross reference weight mechanism to effectively mitigate the influence of significant variations within the same category and cluttered background.
(3) Numerous experiments were conducted on three widely utilized benchmark datasets for FSIC, and the findings demonstrate the significant enhancement achieved by the proposed model. The SOTA classification accuracy for few-shot images has been achieved.

2. Related Work

In 2017, Snell et al. [12] proposed a prototypical network. The researchers highlighted the utilization of deep neural networks to map images into feature vectors. In this approach, each category was represented as a prototype or category center point within the vector space. The feature vectors belonging to the same categories are subjected to an averaging process. In the prototype network, the training objective is to optimize the parameters of the embedding function by minimizing the loss. This enables the network to learn and identify prototypes for each category. In 2018, Sung et al. [13] proposed a model called the Relationship Network. There were two modules incorporated in this paper, namely the relationship module and the feature extraction module. Li et al. [14] proposed a Deep Nearest Neighbor Neural Network (DN4). DN4 was a neural network model designed specifically for FSL and image classification. The primary distinction of this approach was found in its feature representation and the manner in which similarity was computed. In conventional approaches, the representation of image features typically relied on image-level feature measurement. However, DN4 deviated from this practice by incorporating local descriptors of images into categories, thereby replacing image-level feature measurement. In 2021, Rizve et al. [15] proposed the complementary advantages of FSL invariance and equivariant representation. This approach aimed to achieve the necessary characteristics for input transformation and improve discrimination. It was found that features that prioritized transformation discrimination may not be ideal for class discrimination. However, these features can aid in learning the equivariant properties of data structures, leading to improved portability. In 2021, Wu et al. [16] proposed to mine parts in a task-aware manner (TPMN) by incorporating automatic part mining into FSL's metric-based model. TPMN designed a meta filter learner for a meta-learning [17,18] way to produce task-aware part filters based on task embeddings. The task aware part filter can be adapted to any individual task and automatically mines local parts relevant to the task,

even if they are invisible. Gori et al. [19] proposed a Graph Neural Network (GNN) model wherein individual nodes represented samples and the edges denoted the relationships between different samples. Compared to the conventional neural network, GNN takes into account both the inter-sample information and the intra-sample information. Kim et al. [20] proposed a method called EGNN (Edge Labeling Graph Neural Network) to incorporate edge labeling into the GNN. Traditional GNN typically focused on node characteristics and the connectivity between nodes while disregarding the label information associated with the edges. EGNN utilized edge labels to depict the association between samples and integrated them into the process of model learning. The utilization of this particular type of edge label can enhance the model's comprehension and utilization of the similarities and distinctions among samples, thereby enhancing the efficacy of FSL. Chen et al. [21] presented a novel approach in their paper, which introduced a method that integrated spatial and frequency representations to enhance FSL capabilities. By integrating information from both the spatial and frequency domains, the proposed approach extracted a multi-scale feature representation. Additionally, it leveraged transduction reasoning to effectively utilize the label information from the test set. Consequently, the learning performance of few-shot tasks was substantially enhanced.

3. Methodology

3.1. Problem Description

FSL divides the task into two parts. The training set, also called the support set, is divided into N data categories, each of which consists of K samples, referred to as the N-way K-shot problem. The test set is also called the query set, and the categories in the query set belong to the categories in the support set. To solve the N-way K-shot FSIC problem, a priori knowledge is first learned from the auxiliary dataset [13], and then the learned priori knowledge is utilized for image classification and prediction on the target dataset with limited labeling.

In an FSL task, the dataset is divided into a training set $D_{base} = \left\{(x_i, y_i)_{i=1}^{m_{base}}, y_i \in C_{base}\right\}$ and test set $D_{novel} = \left\{(x_i, y_i)_{i=1}^{m_{novel}}, y_i \in C_{novel}\right\}$, where m_{base} and m_{novel} are the sample numbers in D_{base} and D_{novel}, and C_{base} and C_{novel} are the class sets corresponding to D_{base} and D_{novel}, respectively, and $C_{base} \cap C_{novel} = \varnothing$. FSL aims to learn a model on D_{base} that generalizes well to the unseen test set D_{novel}. Scenario training is performed on D_{base} and D_{novel} A series of tasks are sampled as training samples and test samples in the FSL framework, where each classification task consists of a support set and a query set, and the support set $S = \left\{(x_i, y_i)\right\}_{i=1}^{N \times K}$ consists of N support set consists of K samples (N-way K-shot set), which serve as labeled instances. From the same N, the support set consists of B samples from the same classes that are used as unlabeled samples for the query set. $Q = \left\{(x_i, y_i)\right\}_{i=N \times K+1}^{N \times K+N \times B}$ of the query set as unlabeled samples.

$p(T)$ is set up as the task T distribution; scenario training draws a series of training tasks from $p(T)$, and a series of training tasks from $T_{train} \sim p(T)$ as training phase samples, and for a training task T_{train} in the specific task N, a portion of the data for each of the classes $D_{T_{train}}$ operating on it is used, $C_{T_{train}} = \{C_1, C_2, \ldots, C_N\} \in C_{base}$. For this N set of classes, the support set and query set in the training task are both from $D_{T_{train}}$. The data sampling in each task is shown in Figure 1.

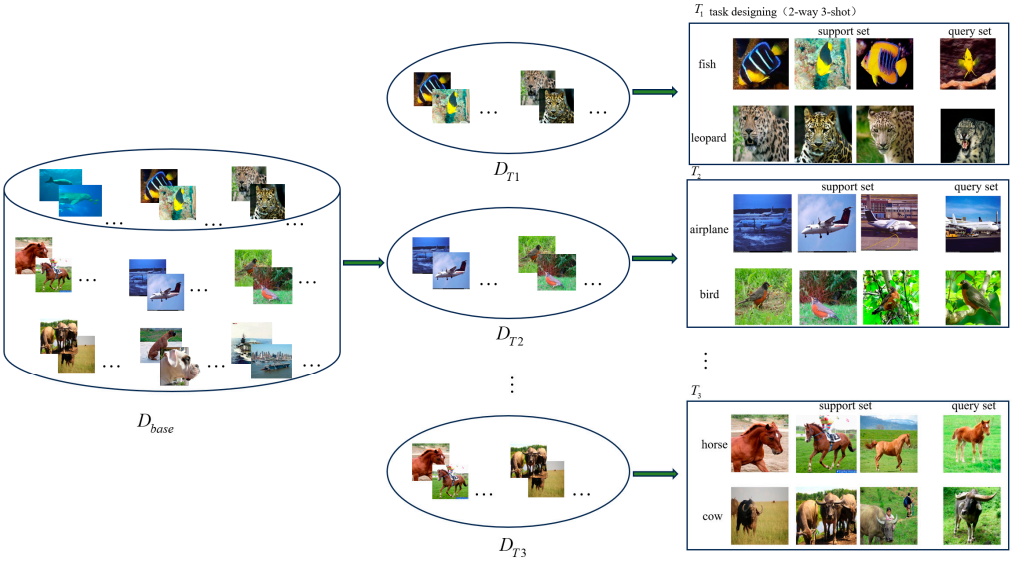


Figure 1. N-way K-shot data diagram.

3.2. Feature Extraction Module Based on Swin Transformer Network

In this study, the utilization of Swin Transformer [22] is implemented. As a module for extracting features in FSIC, the measurement of the distance between the output image features and the EMD is conducted. The architecture of the Swin Transformer is depicted in Figure 2, comprising a convolutional layer, a linear embedding layer, Patch Merging, a Block block, a global adaptive pooling layer, and a fully connected layer. The processing procedure is as follows: Initially, the input image undergoes a convolution operation to partition it into non-overlapping 4 × 4 image blocks. Then, the image blocks undergo transformation into a sequence via the linear embedding layer. In each block, the self-attention mechanism is employed to extract the image features. Subsequently, the feature map undergoes a Patch Merging operation, resulting in down sampling. This process reduces the width and height of the feature map while simultaneously increasing the number of channels. This process involves the extraction of deep features from the image by employing multiple Block and Patch Merging operations.

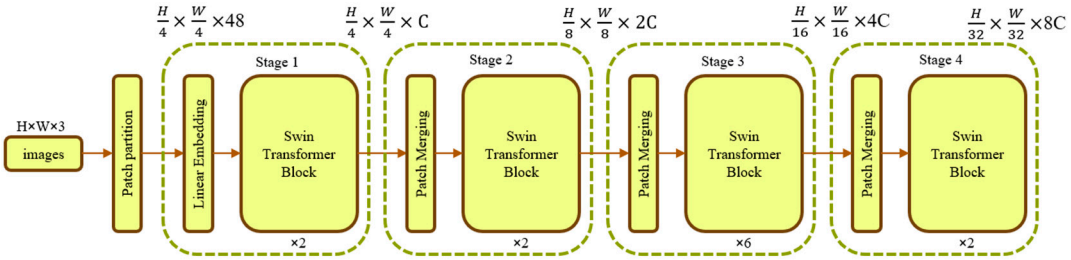


Figure 2. Swin Transformer architecture diagram.

The structure of the Swin Transformer Block is depicted in Figure 3. The LN module is employed to standardize the input features, thereby ensuring that the features across different channels exhibit a similar distribution. This aspect is beneficial in enhancing the stability of the model and expediting the convergence speed during training. The W-MSA module is utilized for conducting multi-head self-attention calculations within the window. By performing attention weight calculations, the model combines the features and facilitates the interaction and information exchange among features located at various positions. This feature facilitates the model's comprehension of the interconnections among various components. The MLP module is a fully connected feedforward network that is capable of performing complex nonlinear transformations on features. This is achieved through

the utilization of multiple fully connected layers and activation functions, which enable the capture of more comprehensive and intricate feature representations. The inclusion of higher-level feature information aids in the learning process of the model. The SW-MSA module is utilized for conducting window moving multi-head self-attention calculations. The proposed approach enhances the capture of contextual information by performing feature translation, reorganization, and integration within a local area. Simultaneously, it also maintains the relative positional relationship between windows, thereby facilitating the efficient extraction of multi scale features. Through the sequential operation of the aforementioned four modules, the process of modeling and integrating the characteristics within the window is achieved.

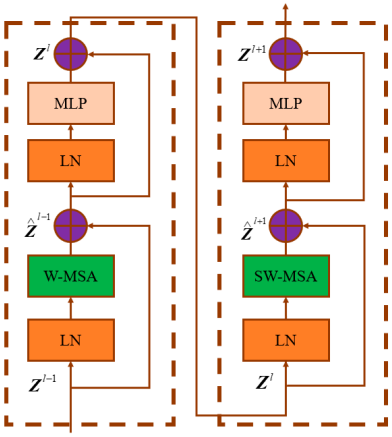


Figure 3. Swin Transformer Block structure.

3.3. CSAM MODULE

The feature maps F enter the channel attention path and the spatial attention path separately. $M_c(F) \in \mathbf{R}^c$ and $M_p(F) \in \mathbf{R}^{H \times W}$, respectively, channel attention to pay attention to the figure and space. CSAM is shown in Figure 4.

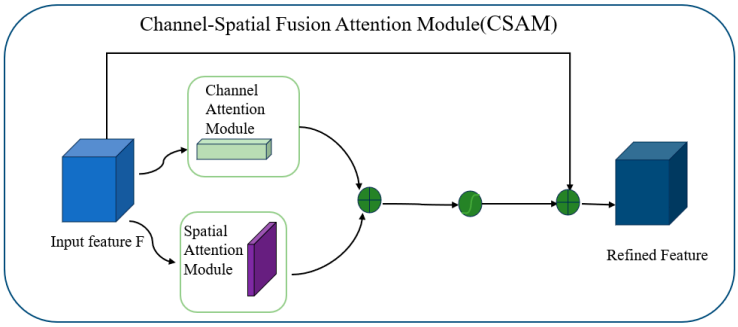


Figure 4. Schematic diagram of access and spatial attention (CSAM).

In order to effectively calculate channel attention, the spatial dimension of the input feature map needs to be compressed. The common method for spatial information aggregation is average pooling, while maximum pooling collects unique object features and can infer attention on finer channels. Therefore, the features obtained after average pooling and maximum pooling are used simultaneously. The calculation of the channel attention path $M_c(F)$ can be expressed as follows:

$$M_C(F) = \sigma(\, FC(\, \text{GlobalAvgPool}(\, F)\,) + FC(\, \text{GlobalMaxPool}(\, F)\,)\,), \tag{1}$$

where GlobalAvgPool represents global average pooling; and GlobalMaxPool indicates global maximum pooling. FC indicates the fully connected layer. σ indicates the sigmoid activation function. A channel attention diagram is shown in Figure 5.

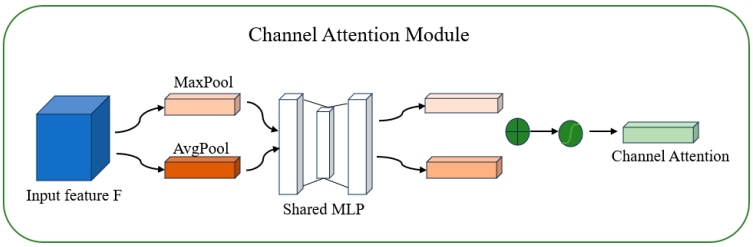


Figure 5. Channel attention diagram.

Formula (1) indicates that the feature graph F is simultaneously passed through the average pooling layer and the maximum pooling layer, and then through the fully connected layer; then, the two are added by elements, the sigmoid activation function is used for nonlinear mapping, and the output of channel dimension is finally obtained. A spatial attention diagram is shown in Figure 6.

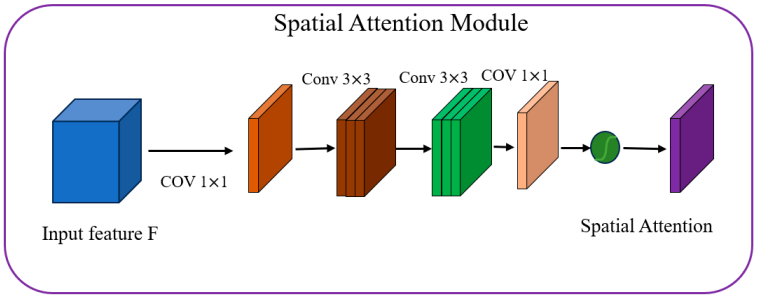


Figure 6. Spatial attention diagram.

As shown in Figure 6, the calculation of spatial attention path $M_p(F)$ can be expressed as follows:

$$M_P(F) = \sigma(\mathrm{Conv}^{1\times1}\left(\mathrm{Conv}^{3\times3}\left(\left(\mathrm{Conv}^{3\times3}\left(\mathrm{Conv}^{1\times1}(F)\right)\right)\right)\right)), \quad (2)$$

where Conv represents the convolution operation, 1×1 and 3×3 represent the convolution kernel size, and σ indicates the sigmoid activation function. The feature graph F is extracted by a series of convolution operations to obtain $M_c(F)$, which is then added to $M_p(F)$ by elements. Then, through the sigmoid activation function, in order to facilitate the forward and backward propagation of information, the low-level features can be directly transmitted to the high-level network, and the original feature diagram F is added by elements to obtain the final module output F_{CSAM}. The calculation process can be expressed as follows:

$$F_{\mathrm{CSAM}} = F + \sigma\left(M_C(F) + M_P(F)\right). \quad (3)$$

Through the analysis of the structure of the channel space fusion attention module, it can be seen that feature extraction of the channel dimension alone can better learn color information into the model, while the learning space dimension can make the network focus more on learning texture features separately; finally, the two are fused to obtain a shape consistent with the original feature map. Then, the processed image output is measured by EMD.

3.4. Measurement Module

3.4.1. Problem Description

EMD is used to solve the optimal solution problem of transportation in linear programming. A certain number of mountain piles are piled up in two different ways, and EMD calculates the sum of the minimum distances required to move one pile into another. Suppose a group of suppliers $A = \{a_i | i = 1, 2, \ldots m\}$ need to transport goods to a specified

group of destinations $B = \{b_j | j = 1, 2, \ldots k\}$, where a_i represents the i-th supplier, and b_j represents the j-th destination. The unit cost from supplier to destination is c_{ij}, and the unit quantity of transportation is x_{ij}. The goal of the transportation problem is to find the cheapest goods flow $\tilde{\mathcal{X}} = \{\tilde{x}_{ij} \mid i = 1, \ldots m, j = 1, \ldots k\}$ from the suppliers to the demanders:

$$\begin{aligned} &\underset{x_{ij}}{minimize} \ \sum_{i=1}^{m} \sum_{j=1}^{k} c_{ij} x_{ij} \\ &subject\ to\ x_{ij} \geq 0, i = 1, \ldots, m, j = 1, \ldots, k \\ &\qquad \sum_{j=1}^{k} x_{ij} = a_i, i = 1, \ldots m \\ &\qquad \sum_{i=1}^{m} x_{ij} = b_j, j = 1, \ldots k \end{aligned} \tag{4}$$

3.4.2. Application of EMD in FSIC

Zhang et al. [11] put forward DeepEMD to address the few-shot challenge with EMD, which is a dynamic round training process. Specifically, the Swin Transformer is utilized in order to generate image embeddings denoted as $U \in R^{H \times W \times C}$, where H and W correspond to the spatial dimensions of the feature map, and C represents the dimensionality of the features. Each image representation comprises a collection of local feature vectors [u_1, u_2, …u_{HW}], and each vector u_i represents a node within the set. v_j is the same, u_i is the query set and v_j is the support set. The overall framework of Swin Transformer + CSAM + EMD is shown in Figure 7. Therefore, the similarity of two images can be quantified by determining the optimal matching cost between two sets of vectors. According to the original EMD formula in Formula (4), the unit cost is determined by computing the paired distance between embedded nodes u_i, v_j derived from two image features:

$$c_{ij} = 1 - \frac{\mathbf{u}_i^T \mathbf{v}_j}{\| \mathbf{u}_i \| \| \mathbf{v}_j \|}, \tag{5}$$

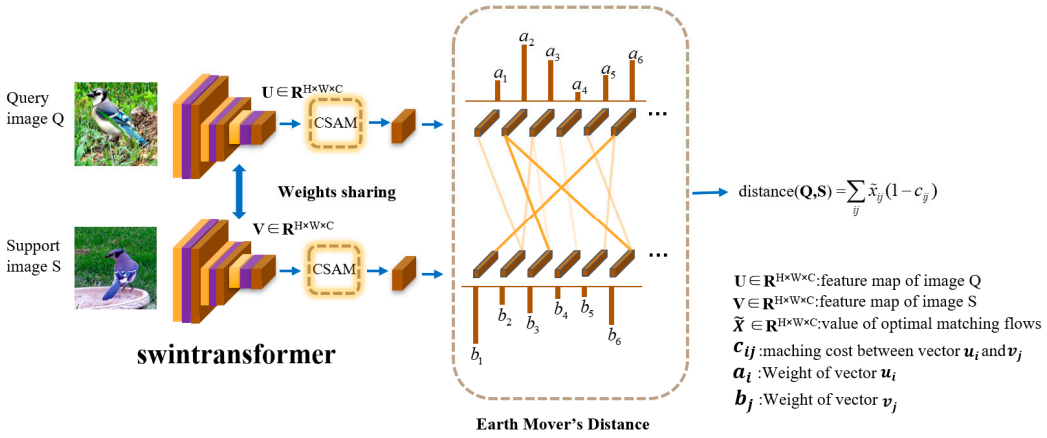


Figure 7. Overall framework of Swin Transformer + CSAM + EMD (STCE).

Nodes exhibiting similar representations tend to produce reduced matching costs among one another. As for the types of weights a_i and b_j, we will elaborate on them in Section 3.4.4. Once the optimal matching stream $\tilde{\mathcal{X}}$ has been obtained, the similarity score between image representations can be expressed as follows:

$$a(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{HW} \sum_{j=1}^{HW} (1 - c_{ij}) \tilde{x}_{ij}. \tag{6}$$

3.4.3. End-to-End Training

Applying Implicit Function Theorem [23–25] to the Optimality (KKT) Condition, the Jacobian theorem can be obtained. For the sake of comprehensiveness, the equation can be expressed in a condensed matrix form, starting from Equation (4).

$$\begin{array}{ll} \text{minimize} & \mathrm{c}(\theta)^{\mathrm{T}}\mathrm{x} \\ \text{subject to} & \mathrm{G}(\theta)\mathrm{x} \leq \mathrm{I}(\theta), \\ & \mathrm{E}(\theta)\mathrm{x} = \mathrm{f}(\theta). \end{array} \tag{7}$$

θ represents the problem parameter associated with the initial layer in a differentiable manner. Ex = f represents the equality constraint and Gx ≤ I the inequality constraint in Formula (4). Therefore, the Lagrangian function for the linear programming problem described in Equation (7) can be represented by the following mathematical expression:

$$\mathrm{L}(\theta, \mathrm{x}, \nu, \lambda) = \mathrm{c}^{\mathrm{T}}\mathrm{x} + \lambda^{\mathrm{T}}(\mathrm{Gx} - \mathrm{I}) + \nu^{\mathrm{T}}(\mathrm{Ex} - \mathrm{f}), \tag{8}$$

where ν is the dual variable with equal constraints, and $\lambda > 0$ is the dual variable with unequal strain. Following the KKT condition with notational convenience, the original dual interior point method $g(\theta, \tilde{x}, \tilde{v}, \tilde{\lambda}) = 0$ is used to solve the problem to obtain the best objective function $(\tilde{x}, \tilde{v}, \tilde{\lambda})$, as follows:

$$g(\theta, x, \nu, \lambda) = \begin{bmatrix} \nabla_\theta L(\theta, x, \nu, \lambda) \\ \boldsymbol{diag}(\lambda)(G(\theta)x - I(\theta)) \\ E(\theta)x - f(\theta) \end{bmatrix} \tag{9}$$

The partial Jacobian of x with respect to θ at the optimal solution $(\tilde{x}, \tilde{v}, \tilde{\lambda})$, denoted as $J_\theta \tilde{x}$, can be obtained by ensuring:

$$J_\theta \tilde{x} = -J_x g(\theta, \tilde{\lambda}, \tilde{v}, \tilde{x})^{-1} J_\theta g(\theta, \tilde{x}, \tilde{v}, \tilde{\lambda}). \tag{10}$$

The (partial) Jacobian with respect to θ can be defined as follows:

$$J_\theta g(\theta, \tilde{\lambda}, \tilde{v}, \tilde{x}) = \begin{bmatrix} J_\theta \nabla_x L(\theta, \tilde{x}, \tilde{v}, \tilde{\lambda}) \\ \boldsymbol{diag}(\tilde{\lambda}) J_\theta (G(\theta)x - h(\theta)) \\ J_\theta (E(\theta)\tilde{x} - f(\theta)) \end{bmatrix}. \tag{11}$$

By solving the optimal solution $\tilde{x}$, we can find the relationship between x and θ. This facilitates efficient backpropagation.

3.4.4. Weight Generation

In the task of FSIC, it is difficult for a single image to assess the significance of local feature representation. In order to reduce the weight of high-variance background areas in two images and increase the weight of co-current object areas, the cross-reference mechanism is used to generate correlation scores as weight values through the dot product between node features and average node features in another structure.

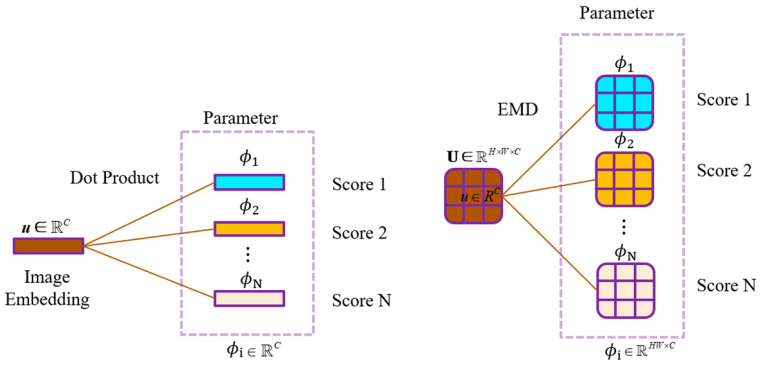
$$a_i = max\{u_i^T \cdot \frac{\sum_{j=1}^{HW} v_j}{HW}, 0\}, \tag{12}$$

b_j can be obtained in the same way; finally, all the weights in the structure are normalized.

$$\hat{a}_i = a_i \frac{HW}{\sum_{j=1}^{HW} a_i}. \tag{13}$$

3.4.5. How to Set K-Shot?

We discussed the situation of the 1-shot before, and we aim to know how to set the K-shot. FC is used to find a prototype vector for each category and uses distance measurement to classify images. Similarly, the K-shot learns a structured fully connected layer (SFCL), in which each category is embedded into a set of vectors instead of a vector. The trained 1-shot model is utilized as a fixed feature extractor, while the parameters within the SFCL are learned from the support set. As shown in Figure 8.



(a) Standard Fully Connected Layer (b) Stuctured Fully Connected Layer

Figure 8. (**a**) illustrates the FC, while (**b**) depicts the SFCL utilized by K-shot. The fully connected layer is responsible for learning a collection of vectors that serve as prototypes for each class. These vectors are then utilized in conjunction with EMD to generate category scores.

4. Experimental Section

We initially present the details of the dataset used and highlight key aspects incorporated in our network design. Finally, a comparison is made between our model and the SOTA method on widely recognized datasets.

4.1. Dataset Description

The algorithm's accuracy was confirmed using these five datasets.

Mini-ImageNet [26]: The mini-ImageNet dataset, which is commonly utilized in the field of FSL, is considered a coarse-grained dataset. In 2016, the Google DeepMind team transitioned from ImageNet to the mini-ImageNet dataset [27]. The dataset used in this study consists of 60,000 color pictures, which have been carefully chosen to represent 100 distinct categories. Each category contains 600 images, which are further divided into subsets for different purposes (specifically, 64 for meta training, 16 for meta verification and 20 for meta testing).

CIFAR-FS [28]: It is a modified version of CIFAR-100 [29], which consists of 100 categories and 600 images per category. It is common practice to divide the dataset into 64 classes for training purposes. while 16 classes are allocated for validation and another set of 20 classes are designated for testing.

CUB-200 [30]: The CUB dataset was initially introduced for the purpose of classifying birds at a fine-grained level. It is divided into three subsets for the purposes of meta training, meta validation, and meta testing, with each subset containing 100, 50, and 50 classes, respectively. One of the main difficulties in this dataset lies in the subtle distinctions between the different bird species.

Tiered-ImageNet [31]: It is a dataset that shares similarities with mini-ImageNet and is considered to be coarse-grained. It consists of 608 classes, with the training set comprising 351 classes, the verification set comprising 97 classes, and the test set consisting of 160 classes. Additionally, this dataset offers a larger number of images for both training and evaluation purposes, with a total of 779,165 images available.

FC100 [32]: The FC100 dataset is a classification dataset that is designed for FSL, derived from the CIFAR100 dataset. The dataset follows a specific split division proposed

in this study [32]. In this split, the 36 original super classes have been reorganized into 12 super classes, consisting of 60 classes, for meta training. Additionally, four super classes, comprising 20 classes, have been allocated for meta validation, and another four super classes, containing 20 classes, have been designated for meta testing.

4.2. Experimental Environment

The experimental environment is an Ubuntu20.00 system, the CPU processor is Xeon(R) Platinum 8352V, and the GPU uses RTX4090 two graphics cards, with a single-video memory of 24 G and two-video memories of 48 G. The model training platform adopts the PyTorch deep learning framework, and the specifically related versions are PyTorch 2.0.0, Python 3.8, and Cuda 11.8.

4.3. Implementation Details

In the experiment, the size of all images in the mini-ImageNet dataset was set to 224 × 224. Simultaneously, common data enhancement strategies were adopted to enhance the data, such as random horizontal flipping and color dithering, and some randomness was introduced into brightness, contrast and saturation. Each training sample made slight changes in these attributes with a certain probability, thus increasing the diversity of data, which was helpful for the model to better adapt to different brightness, contrast and color changes. The tensor was normalized after image conversion, the image pixel values to were normalized mean and standard deviation, the image size of the other four datasets was set to 84 × 84, without using color dithering, and the other steps were the same.

We initialized the pretraining phase using the Swin-T pretrained model with 28,288,354 parameters and 4.5 G FLOPs. The Swin Transformer model's parameters were adjusted through training. The final trained model produced a feature map of size 7 × 7 × 768, which was then average-pooled to a size of 5 × 5 × 768 and performed CSAM attention mechanism feature weighting on the output feature map for ease of downstream EMD fine-tuning.

4.4. Analysis of Ablation Experiment

The module that can be ablated in this experiment is CSAM, and Swin + EMD has excellent performance. The addition of this CSAM module improves the accuracy of the five datasets by about 0.2–0.3%.

Regarding the choice of learning rate, this study attempted values of 0.1, 0.01, 0.0001, 0.00001, and 0.00005 during pretraining. It was found that a learning rate of 0.00005 achieved higher accuracy during pretraining, while a learning rate of 0.0005 achieved higher accuracy during meta training. The performance of different learning rates varied across the five datasets. For example, increasing the learning rate to 0.0005 resulted in a 1% improvement in accuracy for the CUB dataset, but in a slight decrease of around 0.5% for the mini-ImageNet dataset. Fine-tuning the parameters may lead to slight improvements on other datasets, but to determine the optimal learning rate, the mini-ImageNet dataset was chosen as the reference.

SWIN-T pre-training model parameters used in this experiment; the training speed of this method is relatively fast, with pre-training taking about 2 h and fine-tuning about 1 h. In the same configuration environment and the same dataset, with mini-ImageNet, the P > M > F [33] training time is about 56 h. This experiment is expected to reach the highest accuracy rate in 2 h.

Momentum is 0.9, weight decay is 0.05. Epoch is 100, and, typically, 20 epochs in these 5 datasets can have good performance.

The settings of hyperparameters in this experiment refer to those of Hu et al. [33] and Zhang et al. [11] in their experiments; the authors of this paper have selected relatively good parameters after a large number of experiments, and the accuracy rate is shown to be very high.

4.5. Analysis of Experimental Results

All ResNets in the table are represented by R, and compared with the previous SOTA method, we call our pipeline STCE.

From Table 1, we can see that the STCE method proposed in this paper exceeded the accuracy of the highest P > M > F method on the few-shot dataset mini-ImageNet. Specifically, the accuracy was 3.3% higher on one-shot and 1.2% higher on five-shot. On the tiered-ImageNet dataset, it outperformed the TRIDENT method with the highest accuracy by 4.6% on one-shot, reaching the SOTA in the few-shot field on both datasets. From Table 2, we can see that the STCE method proposed in this study achieved higher accuracy than the SOTA BAVARDAGE method on the few-shot dataset FC100. Specifically, it outperformed BAVARDAGE by 6.8% in the one-shot scenario and by 9% in the five-shot scenario. These results demonstrate that the STCE method achieves the SOTA accuracy on the FC100 dataset. From Table 3, we can see that the STCE method proposed exhibited outstanding performance on the few-shot dataset CIFAR-FS. It attained an impressive accuracy of 86.95% in the one-shot scenario, which was marginally lower than the PT + MAP + SF + SOT method by 3%. However, in the five-shot scenario, it demonstrated superior performance compared to PT + MAP + SF + SOT by 1.2%, thereby surpassing it and establishing a SOTA performance. From Table 4, we can see that the STCE method achieved a performance of 83.1% on the CUB dataset in the one-shot scenario and a performance of 92.88% in the five-shot scenario.

Table 1. Results on mini-ImageNet and tiered-ImageNet datasets.

FSCbenchmark		**Mini-Imagenet**		**Tiered-Imagenet**	
Methods	**Backbone**	**Five-Way One-Shot**	**Five-Way Five-Shot**	**Five-Way One-Shot**	**Five-Way Five-Shot**
DPGN [34]	R12	67.77 ± 0.32%	84.6 ± 0.43%	72.45 ± 0.51%	87.24 ± 0.39%
S2M2R [9]	WRN	64.93 ± 0.18%	83.18 ± 0.11%	73.71 ± 0.22%	88.59 ± 0.14%
ProtoCompletion [35]	R12	73.13 ± 0.85%	82.06 ± 0.54%	81.04 ± 0.89%	87.42 ± 0.57%
Baseline++ [36]	R18	51.87 ± 0.77%	75.68 ± 0.63%	51.87 ± 0.77%	75.68 ± 0.63%
SimpleShot [37]	WRN	63.5 ± 0.20%	80.10 ± 0.15%	69.75 ± 0.2%	85.31 ± 0.15%
ConstellationNet [38]	R12	64.89 ± 0.23%	79.95 ± 0.17%	-	-
PAL [39]	R12	69.37 ± 0.64%	84.40 ± 0.44%	72.25 ± 0.72%	86.95 ± 0.47%
TIM-GD [40]	WRN	77.8%	87.4%	82.1%	89.8%
LaplacianShot [41]	WRN	70.27 ± 0.19%	84.07%	79.13 ± 0.21%	86.75 ± 0.15%
BD-CSPN [42]	WRN-28-10	70.31 ± 0.93%	81.89 ± 0.60%	78.74 ± 0.95	86.92 ± 0.63
PT + MAP [43]	WRN	82.92 ± 0.26%	88.82 ± 0.13%	85.67 ± 0.26%	90.45 ± 0.14%
PEMnE-BMS [44]	WRN	83.35 ± 0.25%	89.53 ± 0.13%	86.07 ± 0.25%	91.09 ± 0.14%
TransCNAPS + FETI [45]	R18	79.9 ± 0.8%	91.50 ± 0.4%	73.8 ± 1.0%	87.7 ± 0.6%
TRIDENT [46]	Conv4	86.11 ± 0.59%	95.95 ± 0.28%	86.97 ± 0.50%	96.57 ± 0.17%
P > M > F [33]	ViT-base	95.3%	98.4%	-	-
STCE (Ours)	Swin Transformer	98.65 ± 0.1%	99.6 ± 0.2%	91.6 ± 0.1%	96.55 ± 0.27%

Table 2. Results on FC100 datasets.

FSCbenchmark		**FC100**	
Methods	**Backbone**	**Five-Way One-Shot**	**Five-Way Five-Shot**
BML [47]	R12	45.00 ± 0.41%	63.03 ± 0.41%
IE [15]	R12	47.76 ± 0.77%	65.30 ± 0.76%
DeepEMD [11]	R12	46.47 ± 0.78%	63.22 ± 0.71%
TPMN [16]	R12	46.93 ± 0.71%	63.26 ± 0.74%
PAL [39]	R12	47.20 ± 0.60%	64.00 ± 0.60%
ConstellationNet [38]	R12	43.80 ± 0.20%	59.70 ± 0.20%
BAVARDAGE [48]	R12	57.27 ± 0.29%	70.60 ± 0.21%
STCE (Ours)	Swin Transformer	64.1 ± 0.3%	79.8 ± 0.69%

Table 3. Results on CIFAR-FS datasets.

FSCbenchmark		CIFAR-FS	
Methods	**Backbone**	**Five-Way One-Shot**	**Five-Way Five-Shot**
ProtoNet [12]	ConvNet-64	55.50 ± 0.70%	72.00 ± 0.60%
DPGN [34]	R12	77.9 ± 0.5%	90.2 ± 0.4%
IE [15]	R12	77.87 ± 0.85%	89.74 ± 0.57%
MAML [49]	ConvNet-32	58.90 ± 1.90%	71.50 ± 1.00%
PAL [39]	R12	77.1 ± 0.70%	88.00 ± 0.50%
RENet [50]	R12	74.51 ± 0.46%	86.60 ± 0.32%
ConstellationNet [38]	R12	75.4 ± 0.20%	86.8 ± 0.20%
BD-CSPN [42]	WRN-28-10	72.13 ± 1.01%	82.28 ± 0.69%
S2M2R [9]	WRN	74.81 ± 0.19%	87.47 ± 0.13%
BML [47]	R12	73.45 ± 0.47%	88.04 ± 0.33%
PEMbE-NCM [44]	WRN	74.84 ± 0.21%	87.73 ± 0.15%
PT + MAP + SF + SOT [51]	WRN-28-10	89.94 ± 0.6%	92.83 ± 0.8%
STCE (Ours)	Swin Transformer	86.95 ± 0.2%	94 ± 0.4%

Table 4. Results on CUB datasets.

FSCbenchmark		CUB	
Methods	**Backbone**	**Five-Way One-Shot**	**Five-Way Five-Shot**
MAML [49]	R10	70.32 ± 0.99%	80.93 ± 0.71%
BD-CSPN + ESFR [52]	R18	82.68%	88.65%
ProtoNet [12]	R18	72.99 ± 0.88%	86.64 ± 0.51%
AmdimNet [24]	AmdimNet	77.09 ± 0.21%	89.18 ± 0.13%
MatchingNetwork [53]	R18	73.49 ± 0.89%	84.45 ± 0.58%
S2M2R [9]	WRN	80.68 ± 0.81%	90.85 ± 0.44%
FEAT [54]	R12	73.27 ± 0.22%	85.77 ± 0.14%
BML [47]	R12	76.21 ± 0.63%	90.45 ± 0.36%
DPGN [34]	R12	75.71 ± 0.47%	91.48 ± 0.33%
PT + NCM [43]	WRN	80.57 ± 0.20%	91.15 ± 0.10%
DeepEMD [11]	R12	75.65 ± 0.83%	88.69 ± 0.50%
RENet [50]	R12	79.49 ± 0.44%	91.11 ± 0.24%
Delta-encoder [55]	VGG16	69.8%	82.6%
LaplacianShot [41]	R18	80.96%	88.68%
PEMbE-NCM [44]	WRN	80.82 ± 0.19%	91.46 ± 0.10%
STCE (Ours)	Swin Transformer	83.1 ± 0.4%	92.88 ± 0.4%

During the experiment, we divided it into three stages: pretraining, meta training, and testing. The pretraining models trained on ImageNet-1K include Swin-T, Swin-S, and Swin-B, with model parameters and layer numbers as follows: 28,288,354 parameters and [2,2,6,2] layers for Swin-T, 49,606,258 parameters and [2,2,18,2] layers for Swin-S, and 87,768,224 parameters and [2,2,18,2] layers for Swin-B. We tried various model architectures, including Swin-T, Swin-B, Swin-L, and the upgraded version of Swin Transformer V2 [56]. We found that the pretraining parameters of the Swin-T model are the most suitable. Through an extensive series of experiments, we find that the bigger the model is, the better the parameters are, and they may lead to a series of problems such as over-fitting, which leads to low accuracy in FSIC. To sum up, the experimental findings demonstrate that the utilization of pretraining models for transfer learning yields significantly higher accuracy in FSL compared to more intricate methodologies.

5. Conclusions

This study introduces a novel approach by combining Swin Transformer and CSAM and EMD for the classification of few-shot images. By leveraging the strong feature representation capabilities of Swin Transformer, the model is able to capture more comprehensive

and global image information. To address accuracy issues caused by variations within the same category and cluttered backgrounds, an EMD measurement module is incorporated. The proposed model, STCE, is evaluated on five datasets. The results of the experiment suggest that the STCE algorithm demonstrates better performance in comparison to other methods, achieving SOTA performance in the field of FSL for mini-ImageNet, tiered-ImageNet, and FC100 datasets. Additionally, the model also achieves impressive results on the other two datasets, improving the accuracy of FSIC.

The limitations of the current approach do not perform particularly well at fine granularity; however, we believe that transfer learning can greatly improve accuracy in few-shot domains, and that pre-training and fine-tuning form a type of method with a shorter training time, which has very good prospects for future applications in few-shot domains.

Author Contributions: Conceptualization, P.Z., H.S. and X.H.; methodology, H.S. and P.Z.; software, P.Z. and H.S.; validation, X.Z. and X.H.; formal analysis, P.Z. and H.S.; investigation, P.Z.; resources, H.S.; data curation, P.Z.; writing—original draft preparation, P.Z. and H.S.; writing—review and editing, H.S. and P.Z.; visualization, X.Z.; supervision, X.Z.; project administration, P.Z.; funding acquisition, H.S. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Harbin City Science and Technology Plan Projects grant number 2022ZCZJCG006, Basic Research Support Program for Excellent Young Teachers in Provincial Undergraduate Universities in Heilongjiang Province grant number YQJH2023240, and Collaborative Innovation Achievement Program of Double First-class Disciplines in Heilongjiang Province grant number LJGXCG2023-106.

Data Availability Statement: The data presented in this study are openly available at https://blog.csdn.net/qq_36104364/article/details/107508592.

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

References

1. Tao, H. Smoke Recognition in Satellite Imagery via an Attention Pyramid Network With Bidirectional Multilevel Multigranularity Feature Aggregation and Gated Fusion. *IEEE Internet Things J.* **2024**, *11*, 14047–14057. [CrossRef]
2. Tao, H.; Duan, Q. Learning Discriminative Feature Representation for Estimating Smoke Density of Smoky Vehicle Rear. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 23136–23147. [CrossRef]
3. Liu, M.; Huang, X.; Mallya, A.; Karras, T.; Aila, T.; Lehtinen, J.; Kautz, J. Few-shot unsupervised image-to-image translation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 10551–10560.
4. Zhang, H.; Zhang, J.; Koniusz, P. Few-shot learning via saliency-guided hallucination of samples. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2770–2779.
5. Chen, Z.; Fu, Y.; Wang, Y.; Ma, L.; Liu, W.; Hebert, M. Image deformation meta-networks for one-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8680–8689.
6. Wang, Y.; Girshick, R.; Hebert, M.; Hariharan, B. Low-shot learning from imaginary data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7278–7286.
7. Li, A.; Luo, T.; Lu, Z.; Xiang, T.; Wang, L. Large-scale few-shot learning: Knowledge transfer with class hierarchy. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7212–7220.
8. Peng, Z.; Li, Z.; Zhang, J.; Li, Y.; Qi, G.; Tang, J. Few-shot image recognition with knowledge transfer. In Proceedings of the IEEE/CVF international Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 441–449.
9. Mangla, P.; Singh, M.; Sinha, A.; Kumari, N.; Balasubramanian, V.; Krishnamurthy, B. Charting the right manifold: Manifold mixup for few-shot learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020; pp. 2218–2227.
10. Luo, T.; Li, A.; Xiang, T.; Huang, W.; Wang, L. Few-shot learning with global class representations. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9715–9724.
11. 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.
12. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 4077–4087.

13. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.; Hospedales, T. Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1199–1208.
14. 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.
15. Rizve, M.; Khan, S.; Khan, F.; Shah, M. Exploring complementary strengths of invariant and equivariant representations for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10836–10846.
16. Wu, J.; Zhang, T.; Zhang, Y.; Wu, F. Task-aware part mining network for few-shot learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 8433–8442.
17. Thrun, S.; Pratt, L. *Learning to Learn*; Springer Science & Business Media: New York, NY, USA, 2012; pp. 3–17.
18. Ravi, S.; Larochelle, H. Optimization as a model for few-shot learning. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
19. Gori, M.; Monfardini, G.; Scarselli, F. A new model for learning in graph domains. In Proceedings of the IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005; pp. 729–734.
20. Kim, J.; Kim, T.; Kim, S.; Yoo, C. Edge-labeling graph neural network 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. 11–20.
21. Chen, X.; Wang, G. Few-shot learning by integrating spatial and frequency representation. In Proceedings of the 18th Conference on Robots and Vision (CRV), Burnaby, BC, Canada, 26–28 May 2021; pp. 49–56.
22. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
23. Barratt, S. On the differentiability of the solution to convex optimization problems. *arXiv* **2018**, arXiv:1804.05098.
24. Dontchev, A.; Tyrrell Rockafellar, R. Implicit functions and solution mappings. *IEEE Control. Syst. Mag.* **2011**, *31*, 74–77.
25. Krantz, S.; Parks, H. *The Implicit Function Theorem: History, Theory, and Applications*; Springer Science & Business Media: New York, NY, USA, 2012; pp. 1–12.
26. Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; Wierstra, D. Matching networks for one shot learning. *arXiv* **2016**, arXiv:1606.04080.
27. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Andrej, K.; Aditya, K.; Bernstein Alexander, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
28. Bertinetto, L.; Henriques, J.; Torr, P.; Vedaldi, A. Meta-learning with differentiable closed-form solvers. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
29. Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*; University of Toronto: Toronto, ON, Canada, 2009.
30. Wah, C.; Branson, S.; Welinder, P.; Pietro, P.; Belongie, S. *The Caltech-ucsd Birds-200–2011 Dataset*; California Institute of Technology: Pasadena, CA, USA, 2011.
31. Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J.; Larochelle, H.; Zemel, R. Meta-learning for semi-supervised few-shot classification. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
32. Oreshkin, B.; Rodriguez, P.; Lacoste, A. TADAM: Task dependent adaptive metric for improved few-shot learning. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montréal, QC, Canada, 3–8 December 2018.
33. Hu, S.; Li, D.; Stühmer, J.; Kim, M.; Hospedales, T. Pushing the Limits of Simple Pipelines for Few-Shot Learning: External Data and Fine-Tuning Make a Difference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
34. Yang, L.; Li, L.; Zhang, Z.; Zhou, X.; Zhou, E.; Liu, Y. DPGN: Distribution Propagation Graph Network for Few-Shot Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13387–13396.
35. Zhang, B.; Li, X.; Ye, Y.; Huang, Z.; Zhang, L. Prototype Completion With Primitive Knowledge for Few-Shot Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3754–3762.
36. Chen, W.; Liu, Y.; Kira, Z.; Wang, Y.; Huang, J. A Closer Look at Few-shot Classification. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
37. Wang, Y.; Chao, W.; Weinberger, K.; Maaten, L. SimpleShot: Revisiting Nearest-Neighbor Classification for Few-Shot Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
38. Xu, W.; Xu, Y.; Wang, H.; Tu, Z. Attentional constellation nets for few-shot learning. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
39. Ma, J.; Xie, H.; Han, G.; Chang, S.; Galstyan, A.; Wael, A. Partner-assisted learning for few-shot image classification. In Proceedings of the IEEE/CVF international conference on computer vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10573–10582.

40. Boudiaf, M.; Masud, Z.; Rony, J.; Dolz, J.; Piantanida, P.; Ayed, I. Information Maximization for Few-Shot Learning. In Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, Virtual, 6–12 December 2020.
41. Ziko, I.; Dolz, J.; Granger, E.; Ayed, I. Laplacian Regularized Few-Shot Learning. In Proceedings of the 37th International Conference on Machine Learning, Virtual, 13–18 July 2020.
42. Liu, J.; Song, L.; Qin, Y. Prototype rectification for few-shot learning. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
43. Hu, Y.; Gripon, V.; Pateux, S. Leveraging the feature distribution in transfer-based few-shot learning. In Proceedings of the 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, 14–17 September 2021; pp. 487–499.
44. Hu, Y.; Gripon, V.; Pateux, S. Squeezing backbone feature distributions to the max for efficient few-shot learning. *arXiv* **2021**, arXiv:2110.09446. [CrossRef]
45. Bateni, P.; Barber, J.; Meent, J.; Wood, F. Enhancing Few-Shot Image Classification With Unlabelled Examples. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2021; pp. 2796–2805.
46. Singh, A.; Hadi, J. Transductive decoupled variational inference for few-shot classification. *arXiv* **2022**, arXiv:2208.10559.
47. Zhou, Z.; Qiu, X.; Xie, J.; Wu, J.; Zhang, C. Binocular mutual learning for improving few-shot classification. In Proceedings of the IEEE/CVF international conference on computer vision, Montreal, QC, Canada, 10–17 October 2021; pp. 8402–8411.
48. Hu, Y.; Pateux, S.; Gripon, V. Adaptive dimension reduction and variational inference for transductive few-shot classification. *arXiv* **2022**, arXiv:2209.08527.
49. Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
50. Kang, D.; Kwon, H.; Min, J.; Cho, M. Relational embedding for few-shot classification. In Proceedings of the IEEE/CVF international Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
51. Shalam, D.; Korman, S. The self-optimal- transport feature transform. *arXiv* **2022**, arXiv:2204.03065.
52. Lee, D.; Chung, S. Unsupervised embedding adaptation via early-stage feature reconstruction for few-shot classification. In Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021.
53. Chen, D.; Chen, Y.; Li, Y.; Mao, F.; He, Y.; Xue, H. Self-supervised learning for few-shot image classification. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021.
54. 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. 8808–8817.
55. Schwartz, E.; Karlinsky, L.; Shtok, J.; Harary, S.; Marder, M.; Feris, R.; Kumar, A.; Giryes, R.; Bronstein, A. Delta-encoder: An effective sample synthesis method for few-shot object recognition. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montréal, QC, Canada, 3–8 December 2018.
56. Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. Swin transformer v2: Scaling up capacity and resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.