International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-023-01892-9

ORIGINAL ARTICLE

Self-similarity feature based few-shot learning via hierarchical relation network

Yangqing Zhong[1,2] · Yuling Su[1,2] · Hong Zhao[1,2]

Received: 18 February 2023 / Accepted: 28 May 2023

Abstract
Few-shot learning aims to recognize new visual concepts with a small number of labeled samples. The hierarchical structure based on inter-class labels performs well in many few-shot learning models. However, intra-class features are similar and difficult to distinguish, which is important for mining the correlation and independence between intra-class features in the scene of sparse data. In this paper, we propose a few-shot learning model with a self-similarity feature representation by a hierarchical relation network, which considers inter-class labels and intra-class features to guide few-shot learning. First, we introduce a self-similarity feature representation module as the intermediate feature transform in the neural network. Unlike the traditional model, it extracts specific feature information from intra-class features. Second, we leverage the inter-class label hierarchical structure as important auxiliary information to establish a hierarchical relation network metric module. The module uses coarse-grained information to guide fine-grained classification, which effectively alleviates the problem of insufficient data. Experimental results show that our model improves the classification accuracy, reaching 58.68% on the tieredImageNet dataset.

Keywords Few-shot learning · Self-similarity · Hierarchical classification · Relation network · Multi-granularity

1 Introduction

Few-shot learning is not only a hot research field of machine learning but also a major challenge in applying deep learning [36]. However, the deep learning model heavily relies on many labeled data. In most real-world applications, manually collecting sufficient data and labels is expensive and time-consuming, which limits the universality of the deep learning model. In contrast, humans can easily identify new classes from several labeled examples through a priori knowledge. Inspired by human learning, few-shot learning appears, aiming to learn knowledge from scarce labeled training samples [41]. Many methods alleviate the problem of limited labeled data through few-shot learning [24, 25]. Recently, it has attracted the increasing attention of researchers [3, 19].

The existing few-shot learning approaches can be generally categorized into three classes: data augmentation based, meta-learning optimization based, and matching networks based approaches [2]. First, the data augmentation approaches [4, 46] improve the classification performance of few-shot by increasing the number of training samples or transferring the learned knowledge from large-scale datasets. They have been widely used when the samples with labels are extremely limited. Second, meta-learning optimization based approaches [10, 26, 31] use the task assigned to learn in each task and accumulate knowledge between tasks, rather than relying on standardization to make up for the lack of data. Finally, matching network based approaches [16, 29, 32] usually use the image-level feature metric, which relies on the idea that samples from the same class are more similar than samples from different classes. Nevertheless, these approaches are designed for flat data with fixed samples and simple features. They are unsuitable for hierarchical classification since they treat each class label as a simple symbol without correlations and assume that the classes are independent.

✉ Hong Zhao
hongzhaocn@163.com

[1] School of Computer Science, Minnan Normal University, Zhangzhou, Fujian 363000, China

[2] Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou, Fujian 363000, China

Published online: 20 June 2023

Human beings analyze problems from abstract to concrete with different granularities, and granular computing simulates structured human thinking [37]. One of the critical ideas of granular computing is to summarize data into high-level abstract structures [34, 39]. Hierarchical structures with different granularities can be divided into coarse- and fine-grained. Coarse-grained is a high level of abstract knowledge formed by fine-grained, which can reveal the relationship between fine-grained classes [8, 46]. Figure 1 illustrates an example of class label hierarchical structure. Specifically, classes *Animal*, *Plant*, and *Vehicle* are coarse-grained labels. Among them, classes *Bird*, *Dog*, and *Fish* are fine-grained labels of class *Animal*. The coarse- and fine-grained features are different, and the hierarchical structure of different levels can highlight the relevant features of different granularities [40]. Labels with different granularity classes have different semantic descriptions, which are important auxiliary information for classification.

Many existing few-shot learning methods leverage the hierarchical structure information of inter-class labels to improve the performance of models. According to the different types of inter-class hierarchical structure, it can be categorized into graph structure [3, 46] and tree structure [15, 18, 30]. For instance, Li et al. [15] learned a transferable visual feature model through the tree structure and encoded the semantic relationship between the source and the target class. Su et al. [30] considered the relationship between classes and proposed a few-shot hierarchical classification model based on the tree structure multi-granularity relation network. The above few-shot learning models based on the hierarchical classification of inter-class labels show great potential. However, these models only consider the correlation of inter-class labels and do not fully exploit the crucial information in the intra-class features in the case of few-shot model data scarcity, thus affecting the generalization of the model.

In this paper, we establish a Self-Similarity Feature based Hierarchical Relation Network (SSF-HRNet) for few-shot learning, which realizes few-shot classification from the perspective of intra-class features and inter-class labels. It makes full advantage of the semantic information in sample intra-class features and inter-class label hierarchical structure, which is very important for the scarcity of samples in few-shot learning. SSF-HRNet is divided into two consecutive stages: self-similarity feature representation (SSFR) and hierarchical relation network construction (HRNC). The SSFR module in the first stage first extracts base features through the backbone network. Then, a self-similarity module is used to measure the similarity of feature neighborhood regions and obtain a feature representation of samples with specific feature information. The second stage considers the hierarchical structure of inter-class labels to calculate the multi-granularity features. Then, we establish HRNC to calculate different granularity classes and obtain hierarchical relation scores. Furthermore, we design a hierarchical loss function to calculate the loss of coarse- and fine-grained classes and use coarse-grained class labels to guide the few-shot classification of fine-grained class labels. In this way, we take full advantage of the inter-class label and intra-class feature data of the sample to effectively alleviate the problem of insufficient samples.

The main contributions of our paper can be summarized as follows. First, we learn the self-similarity representation of few-shot learning and implement the self-similarity representation as a module called SSFR, which can obtain the feature representation of samples with specific feature information. Second, we propose a few-shot learning model that takes into account the similarity of intra-class features and the hierarchical structure of inter-class labels and makes full use of the existing sample data information to alleviate the problem of insufficient data. Third, we establish a hierarchical loss function to jointly optimize the loss functions with different granularities to improve the classification ability of the model.

We conduct several experiments on two benchmark datasets: Omniglot [38] and tiredImageNet [44]. Ablation studies validate that both SSFR and HRNC can significantly boost performance. Comparative experiments demonstrate that our model is superior to several advanced models. Moreover, compared with the most advanced relation network [32] few-shot learning model, SSF-HRNet performance is improved by 4.42% under the 5-way 1-shot setting on tiredImageNet. The basic data and source code are available at https://github.com/fhqxa/SSF-HRNet.

The remainder of this paper is organized as follows. In Sect. 2, we review the main related work. In Sect. 3, the details of the proposed model are described. In Sect. 4, the experimental setup is introduced, including datasets, comparison models, and implementation details. In Sect. 5, we

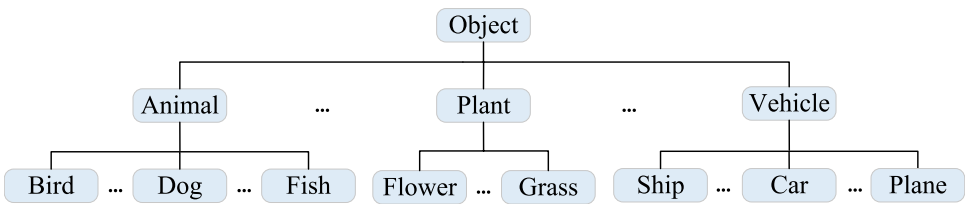

Fig. 1 An example of a class hierarchical tree structure

present the experimental results and analysis. In Sect. 6, we provide the main conclusions and future work.

2 Related work

We briefly review the methods of few-shot learning, hierarchical classification, and self-similarity representation.

2.1 Few-shot learning

Few-shot learning tasks can be grouped into data augmentation based, meta-learning optimization based, and matching network based approaches. Data augmentation based approaches attempt to increase the number of training sets to improve the performance of few-shot learning. Chu et al. [4] considered an image decorrelation sampling method based on maximum entropy reinforcement learning. Zhang et al. [46] presented a data hallucination model for few-shot learning using saliency mapping. Meta-learning based approaches learn within each task and accumulate knowledge between tasks. For instance, Santoro et al. [26] suggested a memory-augmented neural network, which is trained to learn how to store and retrieve memory for each classification task. Jamal et al. [10] presented an entropy-based method to meta-learned the unbiased initial model with the largest output label uncertainty. In addition, Sun [31] proposed a meta-transfer learning method, which is realized by learning the weight and shift function of each task.

Additionally, matching network based approaches usually use metrics to solve few-shot learning problems [22]. For example, Snell et al. [29] represented each class by convolution embedding support examples and classified it by calculating the Euclidean distances to each class prototype. For better metric learning ability, Sung et al. [32] proposed a distance measure on the concatenated feature maps of query and support with the network for classification. Additionally, Li et al. [16] utilized a metric based on local descriptors, which is performed online by k-nearest neighbor search for deep local descriptors. Among them, our work belongs to the matching network based approach. We evaluate the similarity between samples based on the relation network. This method can be better extended to unseen classes through learning and improving the transferability of embedding.

2.2 Hierarchical classification

Many researchers exploit label hierarchical structure information for few-shot learning to improve classification efficiency. The complex label hierarchical structure can be divided into graph and tree structures. From the graph structure perspective, Zhong et al. [46] reorganized all class nodes in the graph into a multi-level super graph and proposed a hierarchical message passing graph neural network framework. Chen et al. [3] explored the relationship between the query and support nodes from the level and realized the effective learning of multi-level relationships on graph neural networks.

From the tree structure perspective, Liu et al. [18] utilized the tree structure as prior knowledge to train a coarse-to-fine classifier, which can accurately predict the many-class few-shot problem. Similarly, Li et al. [15] learned the transferable visual feature according to the hierarchical structure by encoding the semantic relationship between the source class and the target class. Additionally, Su et al. [30] proposed a hierarchical classification model based on a multi-granularity relation network, which constructed a hierarchical tree structure to assist classification. These methods are consistent with our method in terms of objectives. We use the structure information of the inter-class label hierarchical tree to improve the efficiency of the few-shot learning and explore the complementarity of feature knowledge at different granularity hierarchies. Unlike the above methods, we also explore the correlation and independence between intra-class features while using inter-class labels.

2.3 Self-similarity representation

Self-similarity measures the similarity of local areas in the neighborhood to reveal a structural layout of the image [28]. Early studies use self-similarity in the fields of action recognition [11], visual correspondence [35], and target detection [5, 33]. In recent work, self-similarity is used as the intermediate feature transformation of a deep neural network. For example, Kim et al. [13] represented a complete convolution self-similarity descriptor for dense semantic correspondence to make robust matching between different instances in the same object class. Kwon et al. [14] explored a robust motion representation method based on spatiotemporal self-similarity, which can be easily inserted into neural structures and trained end-to-end. Furthermore, Zheng et al. [46] utilized the self-similarity spatial pattern to define the scene structure and maintain consistency while supporting large appearance changes. Inspired by these works, we introduce a self-similarity feature representation (SSFR) module for few-shot learning. Unlike the existing self-similarity models, our SSFR stores the semantic information with intra-class features through intra-class feature transformation and channel network self-similarity, while mining the correlation and independence between intra-class features for few-shot image classification.

3 SSF-HRNet model

In this section, we first introduce the main framework of the Self-Similarity Feature based few-shot learning via a Hierarchical Relation Network (SSF-HRNet). Then, we detail

the two modules involved in this framework: self-similarity feature representation and hierarchical relation network construction.

3.1 Framework overview

Self-similarity reveals a structural layout of an image by measuring similarities of a local patch within its neighborhood [28]. Recent work of [12-14] adopt self-similarity as an intermediate feature transformation for a deep neural network. It has been proved that self-similarity contributes to the effective representation of semantic correspondence in network learning. We introduce intra-class self-similarity feature representation and inter-class label hierarchical tree structure for few-shot learning. The basic framework of SSF-HRNet is illustrated in Fig. 2.

The SSF-HRNet model is composed of two stages:

(1) Self-similarity feature representation (SSFR): we extract the intra-class self-similarity features of the support and query sets according to the self-similarity feature transformation and calculation in the SSFR module.
(2) Hierarchical relation network construction (HRNC): we construct HRNC based on the inter-class label hierarchical tree structure to obtain the relation scores and establish a hierarchical loss function for few-shot learning.

3.2 Self-similarity feature representation

Feature extraction is an indispensable preprocessing step in the process of few-shot classification [9, 46]. We take ResNet12 as the backbone of the network for feature extraction refer to [20]. We use the N-way K-shot strategy in each training iteration to build the few-shot learning model. Each training set consists of support and query sets. Specifically, we randomly select N class samples from the training set and then randomly select K labeled samples from each class as the support set, where $n_s = N \times K$ is the number of support set samples. Similarly, we select the remaining K' samples from the N class as the query set, where $n_q = N \times K'$ is the number of the query set samples.

Let $D_{train} = \{S, Q\}$ be the training dataset and D_{test} be the test dataset, where $D_{train} \cap D_{test} = \emptyset$. Let the support set be $S = \{(x_1, y_1), ..., (x_i, y_i), ..., (x_{n_s}, y_{n_s})\}$ $(i = 1, ..., n_s)$ and the query set be $Q = \{(\tilde{x}_1, \tilde{y}_1), ..., (\tilde{x}_j, \tilde{y}_j), ..., (\tilde{x}_{n_q}, \tilde{y}_{n_q})\}$ $(j = 1, ..., n_q)$. Given the input sample, we first extract the base feature **B** from the backbone feature extraction network ResNet12. Then the SSFR module represents the base features as self-similarity feature **C**, mining the correlation and independence between intra-class features and paying more attention to the relevant feature areas in the image. Figure 3 illustrates the SSFR architecture, consisting of two main learnable modules: the self-similarity feature transformation module and the self-similarity feature calculation module.

Self-similarity feature transformation (SSFT). The SSFT module transforms the base features and comprehensively learns the feature neighborhood information to obtain the self-similarity transformation features. The structure of SSFT is shown in Fig. 3 (a). Given a base feature $\mathbf{B} \in \mathbb{R}^{C \times H \times W}$, where C is the feature dimension, H and W are the height and width of the feature. Base feature **B** is transformed into a C-dimensional self-similarity transformation feature $\mathbf{T} \in \mathbb{R}^{C \times H \times W \times U \times V}$ under the action of SSFT module $\mathcal{F}_t$, whose elements are defined as

$$\mathbf{T}_{C,H,W,U,V} = \mathcal{F}_t(\mathbf{B}_{C,H,W}; \theta_t), \tag{1}$$

where parameter θ_t is the weight of the SSFT module, U and V are the sizes of the extracted sliding local area block. We denote (U, V) with sliding local area block P and $P \in [-d_U, d_U] \times [-d_V, d_V]$. Each position of base feature **B** is set to $x \in [1, H] \times [1, W]$, and we calculate the Hadamard product $\odot$ of x and its neighborhood. Finally, the base feature is transformed into a self-similarity transformation feature

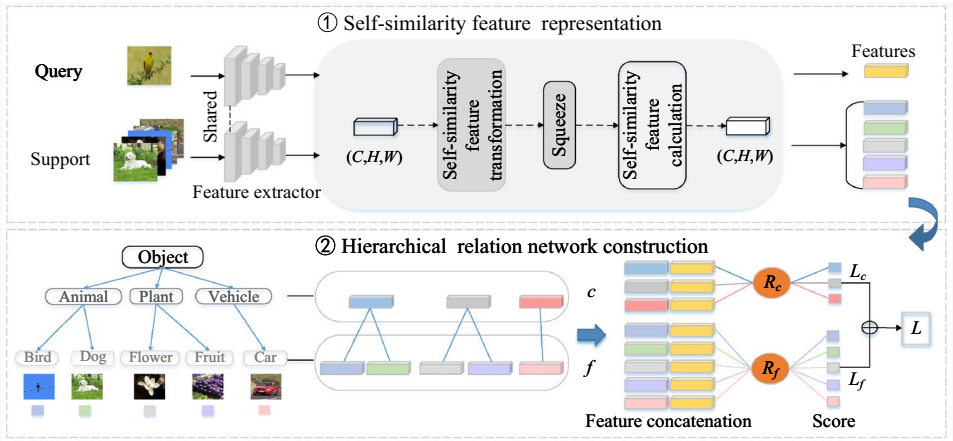

Fig. 2 Framework of the SSF-HRNet under the 5-way 1-shot setting. We use f to represent the fine-grained feature, and c represents the coarse-grained feature. R_c and R_f represent the coarse- and fine-grained relation networks and generate the corresponding coarse-grained loss L_c and fine-grained loss L_f. $\oplus$ means adding up the two losses

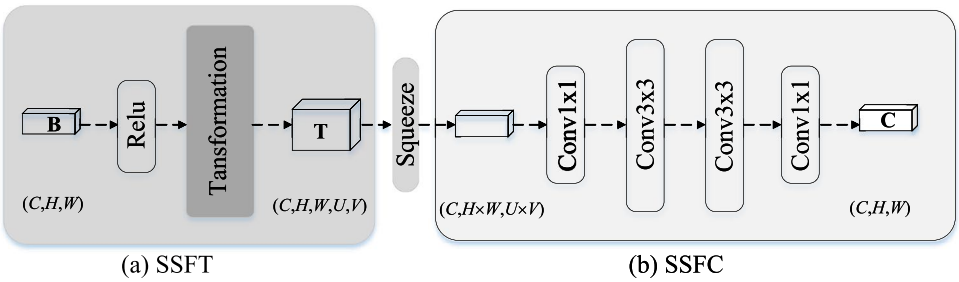

Fig. 3 Architecture of SSFT and SSFC modules. We use **B**, **T**, and **C** to represent the base feature, self-similarity transformation feature, and self-similarity feature, respectively

matrix $\mathbf{T} \in \mathbb{R}^{C\times H\times W\times U\times V}$. The self-similarity transformation feature **T** can be expressed as:

$$\mathbf{T}(x,p) = \frac{\mathbf{B}(x)}{\| \mathbf{B}(x) \|} \odot \frac{\mathbf{B}(x+P)}{\| \mathbf{B}(x+P) \|}, \tag{2}$$

where notation $\|\cdot\|$ refers to the length of the feature tensor.

Self-similarity transformation is closely related to conventional cross-similarity between two different features [5, 11]. We preserve the key semantic information of the feature vector by SSFT of the base features.

Self-similarity feature calculation (SSFC). The SSFC module captures the relationship pattern of self-similarity transformation features through convolution calculation and obtains self-similarity features. Figure 3 (b) illustrates the SSFC module architecture. We first convert the size of the self-similarity transformation feature **T** into the same size as the base feature for the convenience of calculation, i.e., $\mathbb{R}^{C\times H\times W\times U\times V} \rightarrow \mathbb{R}^{C\times(H\times W)\times(U\times V)}$. Then we perform a series of convolution operations in which the convolution block follows the bottleneck structure in terms of computational efficiency. The bottleneck structure can reduce the number of parameters and deepen the number of network layers. We utilize two 1×1 convolution layers in SSFC to reduce and increase the channel size. We adjust the channel size to fit the original input feature $\mathbf{B} \in \mathbb{R}^{C\times H\times W}$. At the same time, we leverage two 3×3 convolution layers in the middle of SSFC for feature calculation. Among them, we insert batch normalization and relu activation functions between convolutions to increase the learning ability of the model. We define the SSFC module as $\mathcal{F}_c$. The final output self-similarity feature $\mathbf{C} \in \mathbb{R}^{C\times H\times W}$ of $\mathcal{F}_c$ is expressed as:

$$\mathbf{C}_{C,H,W} = \mathcal{F}_c(\mathbf{T}_{C,H,W,U,V}; \theta_c), \tag{3}$$

where parameter θ_c is the weight of the SSFC module. The convolution layer gradually collects local correlations without filling. This structural pattern analysis process supplements the appearance patterns in the base feature.

Self-similarity describes the relationship structure of image features by calculating the similarity between image features [12, 14, 28]. We simplify and integrate the SSFR module in Eq. (4). The self-similarity features **X** and $\tilde{\mathbf{X}}$ of the support and query sets extracted by the self-similarity feature extraction module can be expressed as:

$$\begin{cases} \mathbf{X} = \mathbb{F}(\mathbf{B}; \theta_t, \theta_c) \\ \tilde{\mathbf{X}} = \mathbb{F}(\tilde{\mathbf{B}}; \tilde{\theta}_t, \tilde{\theta}_c), \end{cases} \tag{4}$$

where $\mathbb{F}$ represents the whole module of SSFR, **B** and $\tilde{\mathbf{B}}$ are the base features of support and query sets. Parameters $\tilde{\theta}_t$ and $\tilde{\theta}_c$ are the weights of the query set under the SSFR module.

3.3 Hierarchical relation network construction

In the age of big data, human beings find common rules by distinguishing specific fine-grained classes through coarse-grained classes, which are more abstract concepts of fine-grained classes [43]. Different granularities correspond to different requirements, and we build a hierarchical relation network to calculate different granularity classes. The HRNC module first calculates multi-granularity features and then builds a hierarchical relation network. We calculate the hierarchical relation score between support and query samples with different granularities and leverage the relation score to measure the similarity between the two samples.

Multi-granularity feature calculation. We extract multi-granularity features according to the relationship between different granularity classes in the inter-class label hierarchical structure. Combined with the hierarchical and multi-granularity cognitive mechanism of humans processing complex problems, we extract the feature information of samples at different granularity hierarchies to increase the interpretability of the model and improve the learning ability of the model. We take the average value of each five sample features like the support set fine-grained class features on the 5-way 5-shot, maintaining the same number of the support set fine-grained classes as the 5-way 1-shot. Let $\mathbf{X}_i$ be the self-similarity feature matrix of the i-th fine-grained class of the support set, where $i = 1, ..., N$. Let $\tilde{\mathbf{X}}_j$ be the self-similarity feature matrix of the j-th fine-grained class of the query set, where $j = 1, ..., n_q$. The weighted average method is already successfully applied in multi-task learning [46], multi-granularity few-shot learning [30], and hierarchical

classification [46]. We use the weighted average method in the coarse-grained feature calculation, which takes full advantage of the inherent information of data to guide few-shot classification. The features of coarse-grained classes are the weighted average of the nearest features of fine-grained classes. The feature calculation of the k-th coarse-grained class is as follows

$$\mathbf{X}^{(k)} = \frac{1}{|C_k|} \sum_{i=1}^{|C_k|} \mathbf{X}_i, \tag{5}$$

where $k = 1, ..., m_s$, m_s is the number of the coarse-grained class in the support set, C_k is the set of all fine-grained class of the k-th coarse-grained class, and $|C_k|$ is the number of the fine-grained class of the k-th coarse-grained class. Figure 4 illustrates the relationship between coarse- and fine-grained classes. For instance, $\mathbf{X}^{(1)}$ is the coarse-grained class feature, $\mathbf{X}_1$ and $\mathbf{X}_2$ are all fine-grained class features of the 1-st coarse-grained class.

Design of HRNC. We consider the hierarchical structure and design HRNC based on relation network [32]. HRNC is utilized to calculate coarse- and fine-grained class relation scores. We use the relation scores to measure the similarity between support and query samples. The support and query set self-similarity feature with different granularities are adjusted to the same number of channels and then spliced into relationship pairs $\mathcal{C}(\cdot, \cdot)$. For example, a relationship pair of fine-grained classes can be represented as $\mathcal{C}(\mathbf{X}_i, \tilde{\mathbf{X}}_j)$. HRNC includes coarse-grained relation network R_c and fine-grained relation network R_f, which calculate coarse- and fine-grained class relation pairs. The coarse-grained class relation score r_c and fine-grained class relation score r_f are calculated as follows

$$\begin{cases} r_f = R_f(\mathcal{C}(\mathbf{X}_i, \tilde{\mathbf{X}}_j); \phi_f) \\ r_c = R_c(\mathcal{C}(\mathbf{X}^{(k)}, \tilde{\mathbf{X}}_j); \phi_c), \end{cases} \tag{6}$$

where ϕ_c and ϕ_f represent the parameters of coarse- and fine-grained relation networks; Next, we map the relation score into the Softmax function to generate a relation score within a reasonable range. Therefore, the value range of the relation score is 0 to 1. The relation score indicates the similarity between the support and query samples. The higher the relation score, the greater the similarity between the two samples.

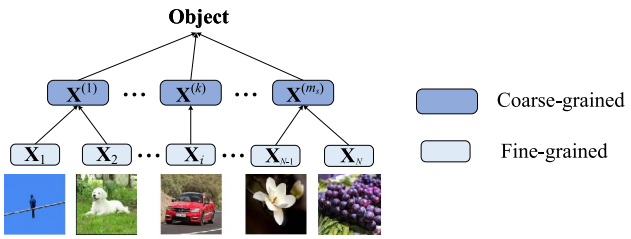

Fig. 4 An example of the relationship between coarse- and fine-grained classes

We combine the coarse- and fine-grained class relation scores to construct the hierarchical relation score r as follows

$$r = r_c + \beta r_f, \tag{7}$$

where parameter β is a weight factor to tradeoff coarse- and fine-grained class relation scores. Finally, we predict the query samples according to the hierarchical relation score. The support sample with the highest relation score is the prediction class of the query samples.

Hierarchical loss function learning. We combine the losses of coarse- and fine-grained classes to construct a hierarchical loss function. First, we calculate the losses of coarse- and fine-grained classes based on the mean square error loss and train our model in Eq. (8). Then, the coarse-grained relation score r_c and fine-grained relation score r_f calculated by Eq. (6) are regressed to the ground truth. In detail, the correct prediction result is one, and the error one is zero. The calculation process of coarse-grained classification loss L_c and fine-grained classification loss L_f is as follows.

$$\begin{cases} L_c = \sum_{l=1}^{m_q} \sum_{k=1}^{m_s} (r_c - \mathbf{1}(y^{(k)} == \tilde{y}^{(l)}))^2 \\ L_f = \sum_{j=1}^{n_q} \sum_{i=1}^{n_s} (r_f - \mathbf{1}(y_i == \tilde{y}_j))^2, \end{cases} \tag{8}$$

where $y^{(k)}$ and y_i are coarse- and fine-grained class labels; $\tilde{y}^{(l)}$ and $\tilde{y}_j$ are coarse- and fine-grained class prediction labels; m_s and m_q are the numbers of coarse-grained class support and query sets; n_s and n_q are the numbers of fine-grained class support and query sets.

Additionally, each granularity class level is not independent but has a close subordinate relationship. Coarse-grained classes can reveal the relationship between fine-grained classes and narrow the scope of fine-grained comparison. The relation scores of different granularity levels affect the final prediction results. Therefore, we synthesize coarse- and fine-grained classification losses to form our hierarchical loss function:

$$\mathcal{L} = L_f + \lambda L_c, \tag{9}$$

where parameter λ is a weight factor to tradeoff coarse-grained classification loss L_c and fine-grained classification loss L_f. Loss $\mathcal{L}$ is the ultimate hierarchical loss function.

Algorithm 1 lists the detailed process of self-similarity feature few-shot learning based on a hierarchical relation network, which provides pseudo-code for the model training

process. We first construct the support and query sets and then calculate the coarse- and fine-grained features on lines 3–6.

Algorithm 1 Self-similarity Feature Based Few-shot Learning via Hierarchical Relation Network (SSF-HRNet)

Input: The training set is D_{train}; $|C_k|$ is the number of the fine-grained class of the k-th coarse-grained class; L_c and L_f are coarse- and fine-grained classification losses.

Output: The parameters $\theta_c, \theta_t, \tilde{\theta}_c, \tilde{\theta}_t, \phi_c, \phi_f$.

Iteration:

1: Initialize parameters $\theta_c, \theta_t, \tilde{\theta}_c, \tilde{\theta}_t, \phi_c, \phi_f$;
2: **for** each episode **do**
3: Randomly select N classes from D_{train} and select K samples from each class to build support set S and query set Q;
4: Obtain base feature $\mathbf{B}$ by ResNet12;
5: Obtain the self-similarity features $\mathbf{X}$ by Eq. (4);
6: Compute the coarse-grained features by $\mathbf{X}^{(k)} = \frac{1}{|C_k|}\sum_{i=1}^{|C_k|}\mathbf{X}_i$;
7: Obtain the hierarchical relation scores by Eq. (6);
8: Update coarse- and fine-grained classification loss based on Eq. (8);
9: Update the hierarchical loss $\mathcal{L} = L_f + \lambda L_c$;
10: Update parameters $\theta_c, \theta_t, \tilde{\theta}_c, \tilde{\theta}_t, \phi_c, \phi_f$;
11: **end for**

We further explore the relationship between samples and obtain the relation score on line 7. Finally, we update the parameters and the hierarchical loss on lines 8–10.

3.4 SSF-HRNet discussion

Combining the two components of self-similarity feature representation in Sect. 3.2 and hierarchical relation network construction in Sect. 3.3, we have obtained a hierarchical relation network few-shot learning method based on the self-similarity feature. Specifically, SSF-HRNet fully utilizes semantic information in sample intra-class features and inter-class label hierarchies, which is extremely important for few-shot learning with scarce samples.

Compared with traditional few-shot learning methods, the SSF-HRNet algorithm has the following merits. First, it can use the neighborhood information of self-similarity representation learning features to obtain key features of the sample. Second, it can use hierarchical information to measure features from multi-granularity relationships. Thirdly, it establishes a hierarchical loss function to jointly optimize loss functions with different granularities to improve the classification ability of the model. SSF-HRNet degenerates to a traditional few-shot learning method if there is no self-similarity feature representation module and hierarchical relation network. When $\lambda = 0$, the optimization problem degenerates to $\min L_f = \sum_{j=1}^{n_q}\sum_{i=1}^{n_s}(r_f - \mathbf{1}(y_i == \tilde{y}_j))^2$. Therefore, the SSF-HRNet is a generalization of traditional few-shot learning.

4 Experimental settings

In this section, we first describe the Omniglot and tieredImageNet datasets used in the experiment. Second, we introduce comparison methods used in the experiment. Our experiments are implemented on an Ubuntu20.04 desktop computer with NVIDIA GeForce RTX 2080 Ti GPU card. The standard SGD optimization algorithm is used, momentum is kept at 0.9, the learning rate is 0.1, and the learning rate is reduced by half for every 100, 000 episode. In our SSF-HRNet, the N-way K-shot strategy is used in training and test stages, including 5-way 1-shot and 5-shot. We follow the evaluation settings for the N-way K-shot evaluation and test 10 query samples for each class in an episode. In the setup of the test phase, we leverage 1000 random episode samples from the test set for a few-shot classification and calculate the average accuracy with the 95% confidence interval.

Table 1 Dataset description. tieredImageNet is given according to the training / val / test division, and Omniglot is given according to training/test splits

Dataset	TieredImageNet	Omniglot
Coarse-grained classes	20 / 6 / 8	38 / 12
Fine-grained classes	351 / 97 / 160	1263 / 360
Images	779,165	32,460
Size	$84 \times 84 \times 3$	28×28

4.1 Datasets

Our experiments are evaluated on two widely utilized few-shot learning datasets: Omniglot [38] and tieredImageNet [44]. Table 1 lists basic statistics for the two datasets, which contain coarse- and fine-grained classes.

The Omniglot [38] dataset includes 1623 different handwritten characters from 50 different alphabets. Each handwritten character contains 20 samples taken by different people. We define the class hierarchy of Omniglot, in which 50 alphabets are defined as coarse-grained classes and 1,623 handwritten characters are defined as fine-grained classes. We randomly selected 38 coarse-grained classes, including 1,263 classes as the training set and the rest as the test set. Additionally, we use the same data preprocessing as in [30]: the size of each image is adjusted to 28×28 and rotated the existing data 90°, 180° and 270° as data enhancement.

The tieredImageNet [44] dataset is a subset of the ImageNet dataset, which has a class hierarchy structure. It contains 34 coarse-grained classes, which can be further divided into the training set, valid set, and test set containing 20, 6, and 8 coarse-grained classes, respectively. Each class has several different image samples, a total of 608 fine-grained classes, including 779,165 images. The size of the image is $84 \times 84 \times 3$.

4.2 Comparison models

We compare SSF-HRNet with several classical and advanced few-shot models:

(1) PN [29] was one of the classical few-shot learning models. It generated prototypes of each class in the support set and then used Euclidean distance to measure the distance between query samples and these prototypes.
(2) RN [32] constructed a neural network to calculate the distance between support and query samples to analyze the matching degree between samples.
(3) MAML [6] aimed to train the initial parameters of the model so that the model can quickly adapt to neural networks with different depths and perform tasks well.
(4) MN [36] learned based on the attention mechanism and memory mechanism and used cosine distance to measure the similarity between support and query samples.
(5) CovaMNet [17] designed a covariance metric network to complete few-shot classification tasks by the covariance representation and covariance metric based on distribution consistency.
(6) LwoF [7] was trained directly with existing large datasets, and the core is how to deal with the output of new tasks and classes.
(7) REPTILE [21] considered the meta-learning problem with task distribution and obtained a previously unseen task sampled from the distribution by an agent that performs well in providing services.
(8) TPN [20] learned feature embedding parameters and graph construction parameters jointly in an end-to-end manner and propagated labels from labeled to unlabeled test instances.
(9) HMRN [30] utilized the hierarchical structure of classes and proposed a few-shot hierarchical classification model based on a multi-granularity relation network.
(10) C2F w/ BDE-MetaBL [42] constructed pseudo-tasks from coarsely-labeled data and grouped each coarse-grained class into pseudo-fine-grained classes through similarity matching to develop a coarse-to-fine pseudo-labeling process.
(11) APL [23] was an algorithm that approximates the probability distribution by remembering the most surprising observations.
(12) IMP [1] presented an infinite mixture prototype for adaptively representing simple and complex data distributions to achieve few-shot learning.
(13) PRN [19] used the principle of minimizing intra-class distance and maximizing inter-class distance to classify.

5 Experimental results and analysis

In this section, we present the experimental results and discussion from four perspectives: (1) we analyze the parameter sensitivity to hierarchical loss function $\mathcal{L}$; (2) we evaluate the effectiveness of the hierarchical loss function; (3) we exploit the effectiveness of SSFR and HRNC modules; (4)

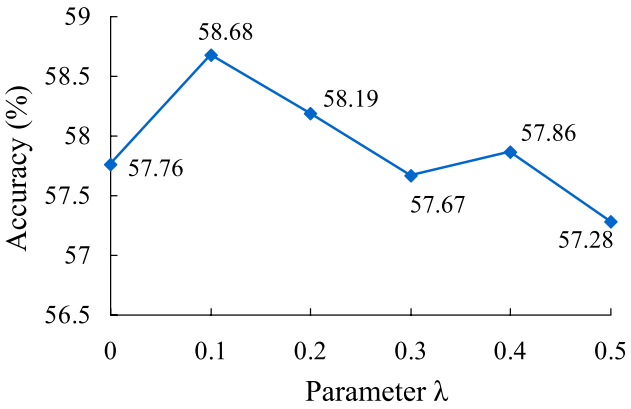

Fig. 5 Performance of parameter λ in hierarchical loss function on the tieredImageNet dataset (5-way 1-shot)

we visualize the performance of the SSF-HRNet model; and (5) we compare the SSF-HRNet with several models.

5.1 Performance of parameter on hierarchical loss function

We analyze the impact of the parameter in the hierarchical loss function on the model. We complete our experiment on tieredImageNet using the 5-way 1-shot strategy. The parameter λ is selected from the candidates $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. The experimental results are shown in Fig. 5, and we can obtain the following observations:

(1) The performance of the model is optimal when the parameter λ is 0.1. The accuracy of the model reaches 58.68%. Therefore, we choose parameter $\lambda = 0.1$ as the optimal coefficient. Parameter $\lambda = 0$ indicates that only fine-grained loss is involved in the hierarchical loss function, and the model accuracy is 57.76%. After adding coarse-grained class loss, the performance of the model is improved, demonstrating that coarse-grained information is helpful for fine-grained classification.
(2) With the increasing proportion of coarse-grained information, the accuracy of the model illustrates a downward trend. It demonstrates that an appropriate amount of coarse-grained information promotes model classification, and excessive coarse-grained information inhibits model classification.

5.2 Performance comparison for the hierarchical loss function

We conduct experiments with/without the strategy of hierarchical loss function on the tieredImageNet and Omniglot datasets to investigate the impact of our hierarchical loss function proposed in Eq. (9). The model replaces the loss with fine-grained classification loss without a hierarchical loss function. The comparison results are listed in Table 2.

From Table 2, we can obtain the following conclusions:

(1) The hierarchical loss function combines coarse- and fine-grained classification losses. We use the hierarchical loss function on the tieredImageNet dataset, and the accuracy of SSF-HRNet using 5-way 1-shot and 5-shot strategies reach 58.68% and 74.47%. Compared with the model without hierarchical loss function, the performance is improved by 0.92% and 0.72% on 5-way 1-shot and 5-shot, respectively. It illustrates that the coarse-grained class information in the hierarchical loss function can be used as higher-level abstract knowledge of fine-grained classes to guide the classification.
(2) On Omniglot, the performance of SSF-HRNet is equivalent to the model without hierarchical loss function under the 5-way 1-shot and 5-shot strategies. The accuracy of the SSF-HRNet reaches 99.02% and 99.56% in 5-way 1-shot and 5-shot. The existing classification accuracy in Omniglot has reached a high level, which aligns with our expectations.
(3) The experimental results show that our hierarchical loss function has achieved effective results on both tieredImageNet and Omniglot datasets. We consider both coarse- and fine-grained classes at the same time and jointly optimize the coarse- and fine-grained classification losses to learn the optimal model and learn the knowledge of different granularities in various networks and scenarios.

Table 2 Effectiveness of hierarchical loss function strategy on the tieredImageNet and Omniglot datasets (%)

Dataset	$\mathcal{L}$	5-way 1-shot	5-way 5-shot
TieredImageNet		57.76	73.75
	✓	**58.68**	**74.47**
Omniglot		98.98	99.55
	✓	**99.02**	**99.56**

$\mathcal{L}$ is the hierarchical loss function, and the best results are marked in bold

5.3 Efficiency comparison of SSFR and HRNC

To investigate the effectiveness of SSFR and HRNC modules, we use 5-way 1-shot and 5-shot strategies to conduct the experiments on the tieredImageNet dataset. The comparison results are listed in Table 3.

From Table 3, we can obtain the following observations:

(1) We replace the self-similarity feature with the base feature without using the SSFR module. The accuracy of the model without the SSFR module reaches 57.43% and 74.04% in 5-way 1-shot and 5-shot strategies. Compared with the model without using the SSFR and HRNC modules, the performance of the model is improved by 3.17% and 2.7% in 1-shot and 5-shot. We leverage the label hierarchical tree structure to con-

Table 3 Contributions of SSFR and HRNC in SSF-HRNet (%)

SSFR	HRNC	5-way 1-shot	5-way 5-shot
		54.26	71.34
✓		57.29	74.02
	✓	57.43	74.04
✓	✓	**58.68**	**74.47**

The best results are marked in bold

struct HRNC. It is useful to construct a hierarchical relation network by an inter-class label hierarchical tree structure.

(2) We skip the hierarchical measurement and only calculate the similarity score of fine-grained classes without the HRNC module. We transform the intra-class features before measuring the relationship between the two samples. We highlight the contribution of intra-class features, and the model accuracy without the HRNC module is 57.29% and 74.02% on 1-shot and 5-shot. The performance of the model is improved by 3.03% and 2.68% without using SSFR and HRNC modules, which demonstrates that using SSFR to transform features is useful for improving the model performance. However, the number of feature channels extracted by SSFR is large and has a low resolution, which increases the difficulty of relation modeling.

(3) Both SSFR and HRNC modules of SSF-HRNet improve classification accuracies on the tieredImagenet dataset. In particular, the combination of the two modules improves the accuracy by about 4.42% in 1-shot, and the improvements in 5-shot are about 3.04%. The accuracy of SSF-HRNet in 1-shot learning is always higher than that in 5-shot learning, which illustrates that the self-similarity feature based on the HRNC module is more helpful when training data is extremely scarce.

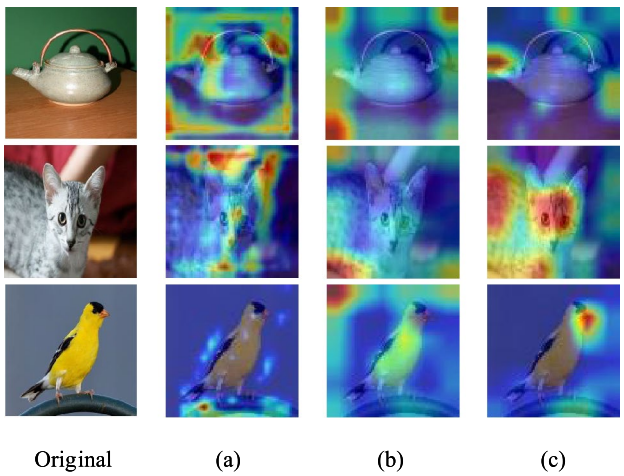

Fig. 6 Visualization of SSF-HRNet model. **a**, **b**, and **c** are respectively the visualization of the base features of the relation network, our model without the SSFR module, and our SSF-HRNet model

5.4 Visualization of SSF-HRNet model

In this section, we have visualized the SSF-HRNet model. We use GradCAM [27] to display the attention range, and the image is randomly sampled from tieredImageNet. We use GradCAM to apply to three models, relation network, our model without the SSFR module, and SSF-HRNet. The visualization results of the three models are shown in Fig. 6. We can get the following observations:

(1) Figure 6 shows three examples, cupping utensils, cat, and bird. The relation network and the model without SSFR are suppressed by the dominant background, and they pay little attention to the main target objects. It indicates that our SSF-HRNet model successfully processes related objects.

(2) Compared with the relation network and the model without the SSFR module, our SSF-HRNet makes the target image features more prominent by learning the self-similarity information of the feature neighborhood while reducing the irrelevant feature information.

5.5 Comparison with different models

This section compares SSF-HRNet with other representative few-shot learning models on tieredImageNet and Omniglot. The comparison models include the hierarchical models considering hierarchical class structure and the flat model without hierarchical class structure.

The results on the tieredImageNet dataset are listed in Table 4. Our SSF-HRNet model outperforms most other models significantly in a 5-way 1-shot and 5-shot strategy. We can obtain the following observations:

(1) The accuracy of the SSF-HRNet model with hierarchy under the 5-way 1-shot strategy is 3.7% higher than

Table 4 Comparison with different models on tieredImageNet (%)

Model	H	5-way 1-shot	5-way 5-shot
PN [29]	N	48.58	69.57
RN* [32]	N	54.26	71.34
MAML [6]	N	51.67	70.30
MN [36]	N	54.02	70.11
CovaMNet [17]	N	54.98	71.52
LwoF [7]	N	50.90	66.69
REPTILE [21]	N	52.36	71.03
TPN [20]	Y	57.53	72.85
HMRN [30]	Y	57.98	74.70
SSF-HRNet*	Y	58.68	74.47

H indicates hierarchical structure. *is our experimental results and the remaining experimental results are copied from the original literature

Table 5 Comparison with different models on the Omniglot dataset (%)

Model	H	5-way 1-shot	5-way 5-shot
PN [29]	N	98.80	99.70
RN [32]	N	99.60	99.80
MAML [6]	N	98.70	99.90
MN [36]	N	98.10	98.90
IMP [1]	N	98.40	99.50
PRN [19]	N	99.27	99.91
APL [23]	N	97.90	99.90
C2F w/ BDE-MetaBL [42]	Y	96.43	98.53
HMRN [30]	Y	98.95	99.41
SSF-HRNet*	Y	99.02	99.56

H indicates hierarchical structure. *is our experimental results and the remaining experimental results are copied from the original literature

that of the optimal model without a hierarchical structure in the model. Each class has only one sample in the 5-way 1-shot strategy. It is difficult for the model to recognize new classes through scarce data. Therefore, SSF-HRNet can improve this problem, and our hierarchical tree structure plays a role in the model as auxiliary information.

(2) Compared with the hierarchical structure model, SSF-HRNet achieves 58.68% classification accuracy for 5-way 1-shot experiments on tieredImageNet. The performance of SSF-HRNet is 0.7% better than the best model and 1.15% higher than the sub-optimal model. It confirms that the self-similarity feature representation module works.

(3) The classification accuracy of SSF-HRNet under a 5-way 5-shot strategy is better than most models. More specifically, the accuracy of the SSF-HRNet model with hierarchy under the 5-way 5-shot strategy is 2.95% higher than the optimal model without hierarchy. It proves that SSF-HRNet is beneficial to image classification with few-shot learning.

(4) Under the 5-way 5-shot strategy, compared with the hierarchical model, the accuracy of the SSF-HRNet model is 74.47%, and the classification accuracy of the HMRN model is 74.70%. Furthermore, HMRN [30] obtains satisfactory results through class structure information in the 5-way 5-shot strategy, which is almost similar to the inter-class label hierarchy information we use. Therefore, SSF-HRNet helps improve the learning ability of few-shot learning.

The performance of the proposed SSF-HRNet and state-of-the-art models with the 5-way 1-shot and 5-shot accuracy on Omniglot is listed in Table 5.

From Table 5, we can obtain the following observations:

(1) The accuracy of SSF-HRNet reaches 99.02% and 99.56% on 5-way 1-shot and 5-shot, respectively. Although the performance of SSF-HRNet is 0.58% and 0.35% lower than that of the optimal flat model in 5-way 1-shot and 5-shot, the accuracy of SSF-HRNet is still higher than that of most flat models, which confirms the effectiveness of SSF-HRNet.

(2) Compared with the hierarchical models, SSF-HRNet outperforms C2F w/ BDE-MetaBL by 2.56% and 1.03% in 1-shot and 5-shot episodes. C2F w/ BDE-MetaBL and SSF-HRNet use the class hierarchical tree structure in classification. The main difference between C2F w/ BDE-MetaBL and SSF-HRNet is that C2F w/ BDE-MetaBL uses the pseudo-tag process from coarse to fine to construct the classification task. In contrast, we based fine-grained classes on constructing coarse-grained classes through a class hierarchical tree structure to assist classification.

(3) Compared with flat models, the SSF-HRNet performance is not always optimal. The main difference between SSF-HRNet and a flat model is whether to use a hierarchical structure. The hierarchical dataset used by SSF-HRNet does not have a hierarchical structure in itself and is constructed from the semantic subordination and dependency between various classes in WordNet. This may deviate from the hierarchical structure in the real world.

6 Conclusions and future work

We proposed a Self-Similarity Feature model based on Hierarchical Relation Network (SSF-HRNet), which combined self-similarity and hierarchy by intra-class features and inter-class label information. It fully uses single sample information to alleviate the problem of insufficient sample data. We exploited the self-similarity feature representation module to transform the intra-class features to extract more specific feature information. Then, we combined the hierarchical structure of inter-class labels to establish a hierarchical relation network to assist with few-shot learning. The experimental results show that SSF-HRNet performs better than other few-shot learning models. However, SSF-HRNet has strict requirements for the hierarchical structure of data. We assume that data has only two layers of coarse and fine granularity, which is not universal in real life. In future work, we will explore hierarchical models on larger datasets to improve the performance of the method. In addition, we will investigate other advanced feature extraction methods for more applications such as object detection and image segmentation.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant No. 62141602 and the Natural Science Foundation of Fujian Province under Grant No. 2021J011003.

References

1. Allen K, Shelhamer E, Shin H, Tenenbaum J (2019) Infinite mixture prototypes for few-shot learning. International Conference on Machine Learning 232–241
2. Cao C, Zhang Y (2022) Learning to compare relation: semantic alignment for few-shot learning. IEEE Trans Image Process 31:1462–1474
3. Chen C, Li K, Wei W, Zhou J, Zeng Z (2021) Hierarchical graph neural networks for few-shot learning. IEEE Trans Circuits Syst Video Technol 32(1):240–252
4. Chu W, Li Y, Chang J, Wang Y (2019) Spot and learn: a maximum-entropy patch sampler for few-shot image classification. IEEE/CVF Conference on Computer Vision and Pattern Recognition 6251–6260
5. Deselaers T, Ferrari V (2010) Global and efficient self-similarity for object classification and detection. IEEE/CVF Conference on Computer Vision and Pattern Recognition 1633–1640
6. Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks. International Conference on Machine Learning 1126–1135
7. Gidaris S, Komodakis N (2018) Dynamic few-shot visual learning without forgetting. IEEE/CVF Conference on Computer Vision and Pattern Recognition 4367–4375
8. Guo S, Zhao H (2021) Hierarchical classification with multi-path selection based on granular computing. J Artif Intell Res 54:2067–2089
9. Yang X, Liang S, Yu H, Gao S, Qian Y (2019) Pseudo-label neighborhood rough set: measures and attribute reductions. Int J Approx Reason 105:112–129
10. Jamal M, Qi G (2019) Task agnostic meta-learning for few-shot learning. IEEE/CVF Conference on Computer Vision and Pattern Recognition 11719–11727
11. Junejo I, Dexter E, Laptev I, Perez P (2010) View-independent action recognition from temporal self-similarities. IEEE Trans Pattern Anal Mach Intell 33(1):172–185
12. Kang D, Kwon H, Min J, Cho M (2021) Relational embedding for few-shot classification. IEEE/CVF International Conference on Computer Vision 8822–8833
13. Kim S, Min D, Ham B, Jeon S, Lin S, Sohn K (2017) Fcss: fully convolutional self-similarity for dense semantic correspondence. IEEE/CVF Conference on Computer Vision and Pattern Recognition 6560–6569
14. Kwon H, Kim M, Kwak S, Cho M (2021) Learning self-similarity in space and time as generalized motion for video action recognition. IEEE/CVF International Conference on Computer Vision, 13065–13075
15. Li A, Luo T, Lu Z, Xiang T, Wang L (2019) Large-scale few-shot learning: knowledge transfer with class hierarchy. IEEE/CVF Conference on Computer Vision and Pattern Recognition 7212–7220
16. Li W, Wang L, Xu J, Huo J, Gao Y, Luo J (2019) Revisiting local descriptor based image-to-class measure for few-shot learning. IEEE/CVF Conference on Computer Vision and Pattern Recognition 7260–7268
17. Li W, Xu J, Huo J, Wang L, Gao Y, Luo J (2019) Distribution consistency based covariance metric networks for few-shot learning. AAAI Conference on Artificial Intelligence 8642–8649
18. Liu L, Zhou T, Long G, Jiang J, Zhang C (2020) Many-class few-shot learning on multi-granularity class hierarchy. IEEE Trans Knowl Data Eng 34(5):2293–2305
19. Liu X, Zhou F, Liu J, Jiang L (2020) Meta-learning based prototype-relation network for few-shot classification. Neurocomputing 383:224–234
20. Liu Y, Lee J, Park M, Kim S, Yang E, Hwang S, Yang Y (2019) Learning to propagate labels: transductive propagation network for few-shot learning. International Conference on Learning Representations 1–14
21. Nichol A, Achiam J, Schulman J (2018) On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999
22. Qiang W, Li J, Su B, Fu J, Xiong H, Wen J (2023) Meta attention-generation network for cross-granularity few-shot learning. Int J Computer Vision 1–23
23. Ramalho T, Garnelo M (2019) Adaptive posterior learning: few-shot learning with a surprise-based memory module. International Conference on Learning Representations 1–14
24. Ren Z, Zhang Y, Wang S (2022) A hybrid framework for lung cancer classification. Electronics 11(10):1614
25. Ren Z, Zhang Y, Wang S (2022) LCDAE: data augmented ensemble framework for lung cancer classification. Technol Cancer Res Treatment 21:1–14
26. Santoro A, Bartunov S, Botvinick M, Wierstra D, Lillicrap T (2016) Meta-learning with memory-augmented neural networks. International Conference on Machine Learning 1842–1850
27. Selvaraju R, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: visual explanations from deep networks via gradient-based localization. IEEE/CVF International Conference on Computer Vision 618–626
28. Shechtman E, Irani M (2007) Matching local self-similarities across images and videos. IEEE/CVF Conference on Computer Vision and pattern recognition pp 1–8
29. Snell J, Swersky K, Zemel R (2017) Prototypical networks for few-shot learning. Adv Neural Inform Process Syst 4077–4087
30. Su Y, Zhao H, Lin Y (2022) Few-shot learning based on hierarchical classification via multi-granularity relation networks. Int J Approx Reason 142:417–429
31. Sun Q, Liu Y, Chua S, Tat, Schiele B (2019) Meta-transfer learning for few-shot learning. IEEE/CVF Conference on Computer Vision and Pattern Recognition 403–412
32. Sung F, Yang Y, Zhang L, Xiang T, Torr P, Hospedales T (2018) Learning to compare: relation network for few-shot learning. IEEE/CVF Conference on Computer Vision and Pattern Recognition 1199–1208
33. Tian G, Liu J, Zhao H, Yang W (2022) Small object detection via dual inspection mechanism for uav visual images. Appl Intell 1–14
34. Tieppo E, Santos R, Barddal J, Nievola J (2021) Hierarchical classification of data streams: a systematic literature review. J Artif Intell Res pp 1–40
35. Torabi A, Bilodeau G (2013) Local self-similarity-based registration of human rois in pairs of stereo thermal-visible videos. Pattern Recogn 46(2):578–589
36. Vinyals O, Blundell C, Lillicrap T, Kavukcuoglu K, Wierstra D (2016) Matching networks for one shot learning. Adv Neural Inform Process Syst 3630–3638
37. Wang B, Liang J, Yao Y (2022) A trilevel analysis of uncertainty measuresin partition-based granular computing. J Artif Intell Res 1–43
38. Wang F, Li C, Zeng Z, Xu K, Cheng S, Liu Y, Sun S (2021) Cornerstone network with feature extractor: a metric-based few-shot model for chinese natural sign language. Appl Intell 51(10):7139–7150

39. Wang G, Yang J, Xu J (2017) Granular computing: from granularity optimization to multi-granularity joint problem solving. Granular Comput 2(3):105–120
40. Wang Y, Liu R, Lin D, Chen D, Li P, Hu Q, Chen C (2021) Coarse-to-fine: progressive knowledge transfer-based multitask convolutional neural network for intelligent large-scale fault diagnosis. IEEE Trans Neural Netw Learn Syst 761–774
41. Wang Y, Yao Q, Kwok J, Ni L (2020) Generalizing from a few examples: a survey on few-shot learning. ACM Comput Surveys 53(3):1–34
42. Yang J, Yang H, Chen L (2021) Towards cross-granularity few-shot learning: coarse-to-fine pseudo-labeling with visual-semantic meta-embedding. ACM International Conference on Multimedia 3005–3014
43. Yao Y (2004) A partition model of granular computing. Trans Rough Sets I:232–253
44. Yu Y, Zhang D, Wang S, Ji Z, Zhang Z (2022) Local spatial alignment network for few-shot learning. Neurocomputing 497:182–190
45. Zhang H, Zhang J, Koniusz P (2019) Few-shot learning via saliency-guided hallucination of samples. IEEE/CVF Conference on Computer Vision and Pattern Recognition 2770–2779
46. Zhang Y, Yang Q (2021) A survey on multi-task learning. IEEE Trans Knowl Data Eng 34(12):5586–5609
47. Zhao H, Hu Q, Zhu P, Wang Y, Wang P (2021) A recursive regularization based feature selection framework for hierarchical classification. IEEE Trans Knowl Data Eng 33(7):2833–2846
48. Zhao H, Yu S (2019) Cost-sensitive feature selection via the $l_{2,1}$ -norm. Int J Approxim Reason 104:25–37
49. Zheng C, Cham T, Cai J (2021) The spatially-correlative loss for various image translation tasks. IEEE/CVF Conference on Computer Vision and Pattern Recognition 16407–16417
50. Zhong Z, Li C, Pang J (2022) Hierarchical message-passing graph neural networks. Data Mining and Knowledge Discovery 1–28