Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 21, No. 3, 2026

A few-shot image classification method based on Ball k-means algorithm

Ruifeng Bai

Chongqing Institute of Foreign Studies, Chongqing 401120, China

Abstract: Ball K-means algorithm is an accelerated and precise clustering method that uses spheres to represent clusters, thereby reducing the distance calculations between data points and centroids. Initially designed for graph data, it was not efficient or accurate for clustering centroids in image data. This paper adjusts the ball K-means algorithm and integrates it with image analysis, expanding its application domain and introducing a novel theoretical framework. The method represents each class of query samples using spherical clusters, identifies neighboring clusters, and distinguishes between active and stable regions. It adaptively updates centroids based on the correlation between active region sample points and neighboring cluster centroids. The resulting centroids are combined with class representatives from training samples to form more distinctive spherical prototypes. The Euclidean distance between query samples and these prototypes is computed for classification based on the nearest-neighbor assignment principle, enhancing model generalization through adaptive features. Experimental results demonstrate that our method achieves improved classification accuracy and robustness on miniImageNet and tieredImageNet datasets.

Keywords: Ball k-means; k-means; Iteration; Image classification; iAdaptability.

1. Introduction

The k-means algorithm, a classic unsupervised learning method, has been widely used in image classification since its inception [1] Its primary function is to simplify image representation by clustering image features, thereby enhancing classification performance. With the advancement of deep learning technologies, k-means has been combined with deep feature extraction, enabling clustering in higher-dimensional feature spaces and effectively handling the vast features generated by deep learning models. However, issues with computational efficiency and scalability remain. To address these, researchers have developed various optimization algorithms, such as Mini-Batch k-means [2] and other variants [3] [4], to speed up computation and reduce memory usage. Despite these improvements, these algorithms still lack adaptive mechanisms, limiting their generalization capabilities. To overcome this, Shuyin Xia and Guoyin Wang introduced the concept of spherical clusters into k-means and developed an accelerated and precise k-means clustering method, known as ball k-means [5]. This algorithm accurately identifies neighboring clusters for each cluster, calculating distances only between points and neighboring cluster centroids rather than all centroids, and adaptively updates centroids.

The ball k-means algorithm is a clustering method that stands out due to its use of spheres to describe each cluster, reducing the distance calculations between points and centroids. This approach precisely identifies neighbor clusters for each cluster, requiring distance calculations only between points and neighbor cluster centroids rather than all centroids. This design allows the ball k-means algorithm to significantly reduce computational overhead and improve efficiency and scalability when processing large-scale data [6]. Additionally, as an unsupervised learning method, the ball k-means algorithm independents of labeled data for training to potentially save on manpower and time expenses. Owing to its strong generalization ability, it can well classify new samples, demonstrating robust adaptability in handling unknown data in practical applications. In addition, it can effectively handle noise and outliers in data [7], while possessing a certain adaptability to the distribution of the data, enabling it to deal with different types of datasets.

Ball K-means, initially applied to tabular data, has been extended to the image domain, broadening its application and introducing new theoretical techniques to the field of image processing [8] [9]. This paper combines ball K-means with few-shot learning [10] [11], aiming to enable effective learning and prediction with minimal labeled samples. However, ball K-means still relies heavily on the initial cluster centroids, which can significantly impact clustering results. Additionally, in the image domain, Euclidean distance alone may not accurately capture the relationship between samples.

This paper presents the following contributions:

(1) This paper introduces the ball k-means algorithm and proposes a simple yet effective method, offering new insights into the few-shot classification problem.

(2) By applying the ball K-means algorithm to the query set for adaptive learning of class features and fusing with the support set to form the granular-ball prototype, the issue of limited training data in few-shot learning is alleviated. Moreover, this approach endows the model with adaptive characteristics and enhanced robustness.

(3) Adjusting the ball K-means algorithm to fit few-shot learning allows for solving few-shot classification problems across domains, broadening the application scope of this algorithm.

2. Related Work

2.1. Granular-ball computing

The granular-ball computing, developed by Shuyin Xia and Guoyin Wang et al. [12], is a significant model in granular computing that has emerged in recent years. This method employs granular-balls of varying sizes to adaptively

represent and enclose the sample space, using these granular-balls as the basis for learning. Shuyin Xia and Guoyin Wang exploited tree structures to partition the space and generate equivalence classes, proposing a rectangle-based granular-ball rough set that allows for continuous data representation and analysis through equivalence classes [13]. Building on this, they further utilized granular-ball computing to eliminate the issue of heterogeneous propagation in traditional neighborhood rough sets, enabling the neighborhood rough set approximations to be represented using equivalence classes. This work established a unified model representation for Pawlak rough sets and neighborhood rough sets, achieving the first adaptive equivalence class representation of rough sets for continuous attributes [14].

2.1.1. Ball k-means

Shuyin Xia and Guoyin Wang further incorporated spherical clustering into the K-means algorithm, developing a simple and fast K-means clustering method called "ball K-means" [12], which uses granular-ball to describe each cluster and improves efficiency by reducing unnecessary distance calculations between data points and cluster centers. The algorithm adopts a neighbor search strategy to identify neighbor clusters for each cluster, allowing data points to compute distances only to adjacent cluster centers. Each ball is divided into a "stable region" and an "active region," where data points in the stable region remain unchanged throughout iterations, while those in the active region may adjust between neighbor clusters. By minimizing Euclidean distance calculations between ball centers in each iteration, the balls gradually stabilize over time. Using hyperspherical quantization to cluster the space yields more accurate neighborhood relationships without requiring additional parameters and eliminates the computational overhead of individual samples found in existing acceleration algorithms [5]

Ball k-means diverges from existing state-of-the-art k-means algorithms that rely on adjusting upper and lower bounds for individual samples, eliminating the need to maintain bounds for each sample, achieving a more precise, straightforward, and adaptive computation process. The overall algorithm is simple yet highly efficient.

2.2. Few-shot Learning

The goal of few-shot learning [15] is to develop a classifier that can generalize well to new classes with only limited labeled samples for each class [16]. To address this problem, researchers have proposed various methods, mainly categorized into metric learning [17] and meta-learning. Metric learning focuses on learning to measure the similarity between pairs of samples, assigning higher scores to similar pairs and lower scores to dissimilar pairs, Such as Matching Networks (MN) [18], Prototypical Networks (PN) [19], and Relation Networks (RN) [20] and so on [21-23]. Meta-learning [24-25] emphasizes cross-task learning, where all tasks follow the same paradigm. It adapts to new tasks not at the sample level but at the task level, learning a task-agnostic rather than task-specific model, Among them, MAML [26] and Reptile [27] are two classical methods, while other methods are variants derived from them [28-31]. Metric learning can be considered a specialized form of meta-learning, where similarity measurement across tasks allows the results to be transferred between tasks. Both approaches have achieved notable success in few-shot image classification but have yet to address the fundamental challenge of limited training data in this domain. However, both methods face challenges related to limited training data and poor robustness. [32-33]

3. Model

The ball k-means algorithm has been adapted to address the requirements of few-shot image classification tasks. In this modified version, initial cluster centers are determined not by random selection as in traditional k-means methods, but by using prototypes during the initial spherical clustering of the query set. To determine whether samples in the active region need to be relocated, the correlation between a sample and neighbor cluster centers is computed instead of measuring the Euclidean distance between them. Furthermore, during the iteration process, upper-level cluster centers and samples propagate to the lower level until the iteration is complete, resulting in stable cluster centers. Ultimately, granular-ball prototypes are obtained by merging the prototypes and cluster centers through weighted fusion. The Euclidean distance is computed between the query set samples and the prototypes, employing a nearest-neighbor assignment technique for classifying the samples. The Fig. 1 presents the overall architecture of the model, elucidating the entire process.

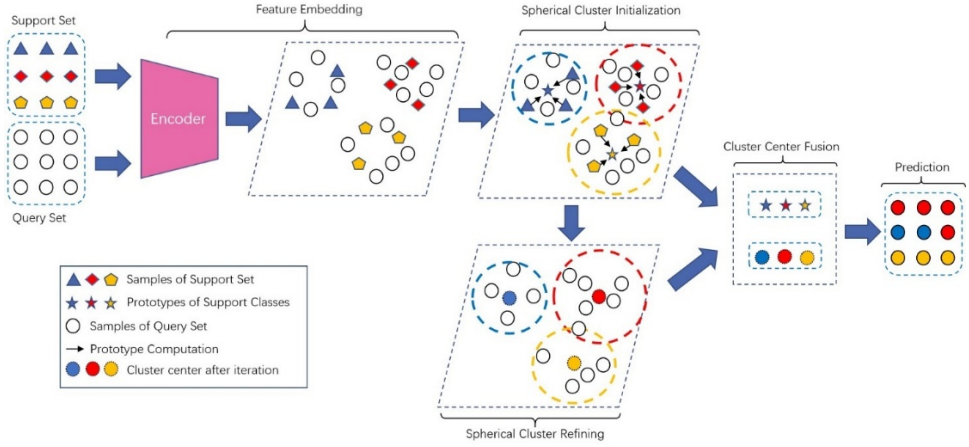

Fig. 1 Granular-ball Prototypical Network

The architecture of the granular-ball prototype network, taking 3-way 3-shot as an example, the query set samples are partitioned into 3 clusters based on prototypes. Using the granular algorithm, 3 stable cluster centers are obtained, which are then fused with the prototypes to derive granular prototypes. Finally, classification is performed based on these

prototypes.

3.1. Problem Definition

Due to the scarcity of labeled samples, few-shot classification often leads to overfitting. Employing cross-task training mechanisms after Matching Networks, such as Prototypical Networks and Relation Networks, using meta-learning training strategies has proven to be effective in mitigating overfitting. The dataset for meta-learning is split into training and testing subsets. From the training data, a random selection of N classes and K labeled samples per class are taken to form the support set, which is used to construct each meta-learning task, known as the N-way K-shot task. A subset of samples is then drawn from the remaining samples of the selected Nclasses to form the query set. By learning across a series of distinct meta-learning tasks, the model learns to distinguish the N classes and predict the query set data. During testing, the test data, divided into support and query sets, is fed into the model to classify images in the query set.

The support set S, query set Q, and meta-learning task T are defined as follows.

$$S = \{(X_i, y_i)\}_{i=1}^{N\times K} \quad (1)$$

$$Q = \{(\tilde{X}_i, \tilde{y}_i)\}_{i=1}^{N\times q} \quad (2)$$

$$T = \{(S_i, Q_i)\}_{i=1}^{m} \quad (3)$$

$X_i, \tilde{X}_i, y_i$ and $\tilde{y}_i$ respectively denote the samples from the support set, the samples from the query set, and their respective labels, N denotes the number of sample categories, K indicates the number of samples per class in the support set, q represents the number of samples per class in the query set, m specifies the number of meta-learning tasks.

3.2. Granular-ball Prototypical Network

We integrate the particle-sphere algorithm with image processing using a few-shot learning approach to develop a Granular-Ball Prototypical Network (GBPNet), divides the query sets into multiple spherical clusters. By iteratively and adaptively updating the cluster centers, stable centers are combined with the prototypes obtained from the mean values of the support sets through weighted fusion, creating granular-ball prototypes that are more characteristic of each class. The Euclidean distance between each sample in the query set and these prototypes is then calculated, and classification is conducted based on the nearest-neighbor assignment principle.

3.2.1. Division of initial clusters

The prototype network employs an encoder (such as a convolutional neural network) to embed the features of the support and query set samples, where refers to the parameters of the encoder. The prototype is formed by calculating the average of the feature vectors from samples within class i of the support set, as given by the following formula:

The colored dots indicate their respective classes, while triangles represent computed cluster centers. It can be observed that some samples have been incorrectly assigned to different classes.

$$P_j = \frac{1}{|S_j|}\sum_{(X_j,y_j)\in S_j} f_\theta(X_j) \quad (4)$$

j represents the collection of samples from class j in the support set, while X_j refers to the feature vectors of support set samples obtained through the embedding function.

To determine the initial spherical cluster partitioning, assign each query set sample to the nearest prototype as shown below:

$$b(\tilde{X}_i) = argmin_{j=1,\dots,k}\{\, d(f_\theta(\tilde{X}_i), P_j)\} \quad (5)$$

$b(\tilde{X}_i)$ is the conditions satisfied by the query set sample distribution, and $(f_\theta(\tilde{X}_i), P_j)$ indicates the Euclidean distance between the prototype of class j and the feature vector from a sample in the query set.

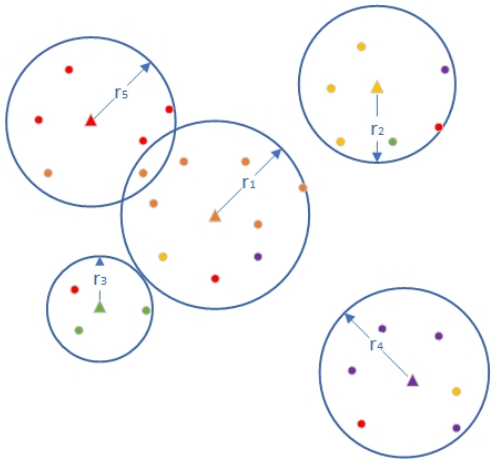

Fig. 2 Initialization of spherical clusters

3.2.2. Definition of spherical cluster

The query set samples are initially divided into k clusters based on the prototypes, with each cluster representing information about the corresponding class in the query set. For obtaining more accurate information, cluster centers undergo adaptive iteration. This process relies on the definition of a ball, where the structure of a ball is characterized by its radius and centroid. The clusters radius and centroid define the cluster itself, which is referred to as a spherical cluster. During subsequent iterations, both the radius and centroid are updated. The initialization of the spherical clusters is shown in Fig.2. The formula is shown below:

$$c_j^t = \begin{cases} \frac{1}{|Q_j|}\sum_{(\tilde{X}_i,\tilde{y}_i)\in Q_j}\{f_\theta(\tilde{X}_i)|b(\tilde{X}_i) = j\} & ,t = 0 \\ \frac{1}{|Q_j|+1}(\sum_{(\tilde{X}_i,\tilde{y}_i)\in Q_j}(\{f_\theta(\tilde{X}_i)|b(\tilde{X}_i) = j\}) + c_j^{t-1}) & ,t > 0 \end{cases},$$

$$r_j^t = max(d(f_\theta(\tilde{X}_i), c_j^t)) \quad (6)$$

t is the number of iterations, refers to the set of samples in class j within the query set, and $f_\theta(\tilde{X}_i)$ represents the feature vector obtained by the embedding function of the query set samples.

3.2.3. Definition of neighbor clusters

For samples that are far from the centroid of spherical cluster i, calculating their distance to the centroid only increases the computational load. By adopting a method to find neighbor clusters to spherical cluster i, the workload of distance calculations can be reduced. In subsequent iterations, only the distances between the centroids and the samples in the neighbor clusters are computed, rather than distances to samples in all clusters centorids, thereby reducing the number of distance calculations between samples and the centroids. A cluster qualifies as a neighbor cluster to spherical cluster i if it meets the following conditions and the neighbor cluster division is as Fig.3:

$$\frac{1}{2}\Big(d\big(c_i, c_j\big)\Big) < r_j, i = 1, \dots, k \text{ and } i \neq j \quad (7)$$

From condition (4), its evident that the mutual ne ighbor relationship is asymmetrical. Therefore, as depicted in the diagram, any two spherical clusters i and j conform to one of

the specified relationships:

①Spherical clusters i and j are mutual neighbors (such as c_5 and c_1).

②Spherical cluster i is a neighbor of j, while j is not a neighbor of i (such as c_1 and c_2, c_1 and c_3).

③No mutual neighbor relationship exists between them (such as c_1 and c_4).

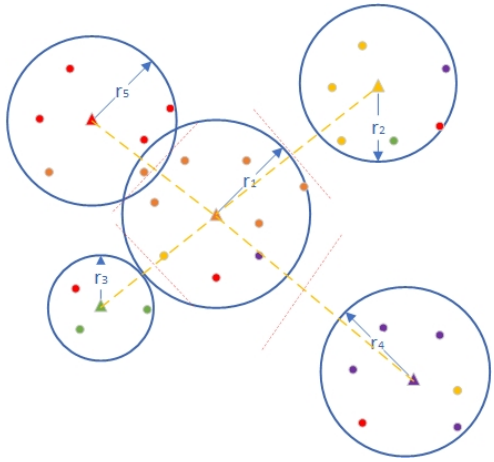

Fig. 3 Neighbor cluster division

The yellow dashed line represents the distance between cluster centers, while the red dashed line perpendicular to this distance represents half of that distance. In terms of the relationship between lines and circles, the red dashed line intersects with the circle to indicate adjacency between spherical clusters.

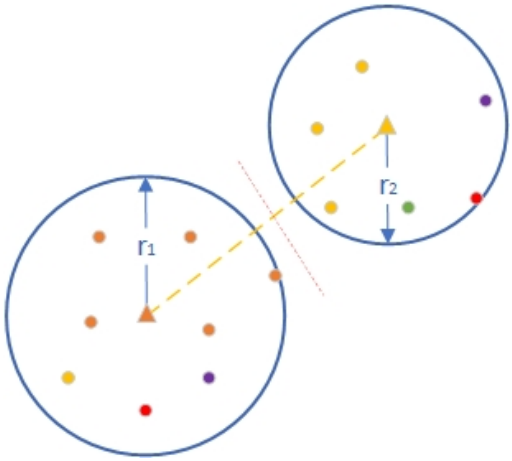

Fig. 4 Stable spherical clusters

This situation arises when searching for neighboring clusters, and represents the state after each coarse adaptive iteration update.

3.2.4. Division of spherical cluster regions

A spherical cluster with neighbor clusters indicates that its assigned sample points can be relocated. We need to further determine the direction in which the sample points should move. The region of spherical cluster i is divided into a stable region and an active region based on half the distance between its centroid and the centroids of neighbor clusters. Sample points within the stable region are excluded from the correlation assessment with neighbor clusters during subsequent iterations. If the following formula is met, the area is considered a stable region and as shown Fig.4:

$$\Re_C^0 = \frac{1}{2} min(d(c_i, c_j))_{c_j \in \{N_{C_i}\}} \quad (8)$$

When a spherical cluster has no neighbor clusters, its stable region equals the entire spherical cluster. The annular area outside the stable region is referred to as the active region. The active region is subdivided into neighboring annular areas, each associated with a corresponding nearest neighbor cluster. The number of neighboring annular areas matches the number of neighbor clusters. Sample points located in any annular region of spherical cluster i can only move within that cluster or among its first i neighbor clusters. The Fig.5 shows how each globular cluster divides its region according to its neighbors If condition (10) is met, the sample point should be moved to neighbor cluster j:

$$\Re_C^i = \begin{cases} \frac{1}{2} d(c_i, c_j) < \frac{1}{2} d(f_\theta(\tilde{X}_i), c_i) < \frac{1}{2} d(c_i, c_{j+1}), 0 < j < k' \\ \frac{1}{2} d(c_i, c_j) < \frac{1}{2} d(f_\theta(\tilde{X}_i), c_i) < r_i, \qquad j = k' \end{cases} \quad (9)$$

$$f_\theta(\tilde{X}_i) \cdot c_i < f_\theta(\tilde{X}_i) \cdot c_j \quad (10)$$

$\Re_C^i$ represents the i annular region, while $\Re_C^0$ denotes the stable region.

3.2.5. Prediction of neighbor clusters during iteration

The above definitions of neighbor clusters and the division of spherical cluster regions aim to reduce the number of calculations between sample points and centroids. Predicting neighbor clusters helps minimize the number of centroid-to-centroid distance calculations during the iterative process. Neighbor clusters are predicted based on the relationships computed for spherical cluster i in the previous iteration. If the following conditions are met, spherical cluster j is not a neighbor cluster of spherical cluster i in the current iteration, and the distance calculation between them can be omitted:

$$d(c_i^{t-1}, c_j^{t-1}) \geq 2r_i^t + d(c_i^{t-1}, c_i^t) + d(c_j^{t-1}, c_j^t) \quad (11)$$

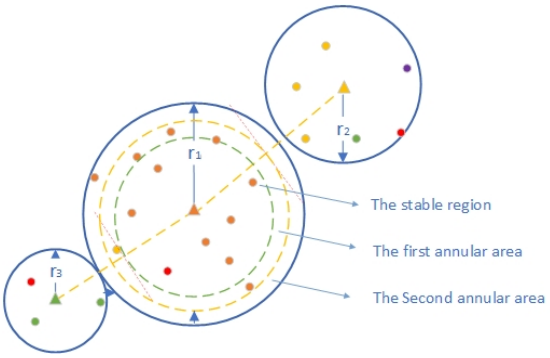

Fig. 5 Annular region partitioning

During the iteration process, samples within the green dashed line do not move. Samples between the yellow dashed line and the green dashed line only move within their own circle around the green cluster center. Samples outside the yellow dashed line are allowed to move within any of the three circles.

3.2.6. Granular-ball prototype

After the iterations conclude, the centroid of each spherical cluster can accurately represent the sample information of each class in the query set. The support set provides only one or a few samples, so the resulting prototypes may not fully capture each classs information. Weighted fusion of the spherical cluster center and the prototype of the support set is performed to construct a spherical prototype that that more closely aligns with the category characteristics:

$$c_j^* = \alpha P_j + (1 - \alpha) c_j \quad (12)$$

α represents the weight parameter, which changes with the number of training times.

After obtaining the granular-ball prototype, compute the

softmax of the Euclidean distance between the query set samples and the granular-ball prototype to obtain the assignment matrix:

$$A_{ij} = \frac{exp\left(-d(f_\theta(\tilde{X}_i), C_j^*)\right)}{\sum_{j=1}^{N} exp\left(-d(f_\theta(\tilde{X}_i), C_j^*)\right)} \quad (13)$$

By computing cross-entropy, the discrepancy between the predicted model results and the actual

labels can be quantified. The smaller this value, the closer the prediction is to the true labels, and vice versa, a larger discrepancy indicates a more significant difference:

$$L(\theta) = -\frac{1}{M*N}\sum_{i=1}^{M}\sum_{j=1}^{N}(\tilde{Y}_{ij} log A_{ij}) \quad (14)$$

$\tilde{Y}_{ij}$ represents the true label matrix, while M denotes the number of samples in the query set.

4. Experiment

4.1. Data Set

The granular-ball prototype network was evaluated across two datasets, tieredImageNet and miniImageNet, and compared to classical networks.

4.1.1. MiniImageNet

The miniImageNet dataset is a widely recognized benchmark used in few-shot learning research. It was initially introduced by Vinyals et al. in 2016 for their matching networks model and is derived from the larger ImageNet dataset, specifically ILSVRC-12. Key features of miniImageNet include:

Classes and Samples: It comprises 100 classes, each containing 600 images, resulting in a total of 60,000 images.

Image Size: All images have been resized to 84×84 pixels.

Training, Validation, Testing Split: The dataset is typically divided into 64 classes for training, 16 for validation, and 20 for testing, based on the division provided by Ravi and Larochelle in 2017.

Due to its challenging classification task with limited training samples per class, it has become one of the most widely used benchmarks for developing and evaluating few-shot learning models.

4.1.2. TieredImageNet

The tieredImageNet dataset is a large-scale benchmark designed for few-shot learning tasks. It was introduced by Ren et al. in 2018 to provide a more challenging and realistic setting. Here are some key characteristics:

Classes and Categories: The dataset includes 608 classes grouped into 34 broader, higher-level categories, based on the hierarchical structure of ILSVRC-12. This grouping helps ensure that training and testing classes are semantically disjoint, making the classification task more challenging.

Splits: The classes are divided into subsets as follows:

Training: 20 categories containing 351 classes.

Validation: 6 categories containing 97 classes.

Testing: 8 categories containing 160 classes.

Image Size: All images are resized to 84×84 pixels.

The hierarchical nature of the dataset ensures that the classes in the training set do not overlap with those in the testing set, allowing for a more robust assessment of few-shot learning models generalization abilities.

4.2. Experimental configuration

The effectiveness of the approach was demonstrated by employing an identical network structure as used in conventional models. The encoder specifically consists of four convolutional blocks, where each block contains a 3 × 3 convolution with 64 filters, followed by batch normalization, ReLU nonlinearity and 2 × 2 max pooling ization, collectively referred to as "conv-64F". In accordance with the episodic training mechanism, N-way K-shot training episodes were sampled to replicate the N-way K-shot testing tasks. Our method was evaluated with a 5-way configuration, including 5-shot and 1-shot setups. For all configurations, 15 query samples were used, and the mean accuracy was determined from more than 2000 testing episodes. The experiments were carried out using PyTorch on an NVIDIA V100 GPU. All models were optimized with Adam, starting with a learning rate of 0.001. For both miniImageNet and tieredImageNet datasets, we constructed 200 episodes to train our granular-ball prototype network, reducing the learning rate by half every 20 epochs.

4.3. Experimental results

Table 1. Accuracy in few-shot classification on miniImageNet

Model	backbone	5-way 1-shot	5-way 5-shot
MN [18]	Conv-64F	43.56	55.31
SSL [34]	Conv-64F	50.60	65.71
RN [20]	Conv-64F	50.44	65.32
MAML [26]	Conv-64F	48.70	63.11
PN [19]	Conv-64F	49.20	66.20
Reptile [27]	Conv-64F	49.97	65.99
FEAT [35]	Conv-64F	51.05	67.53
HFFCR [23]	Conv-64F	51.79	65.74
CGRN [22]	Conv-64F	50.85	64.13
Bayesian [36]	Conv-64F	50.02	65.48
BOIL [37]	Conv-64F	49.61	65.44
ECMT [38]	Conv-64F	49.07	65.73
LMPNet [39]	Conv-64F	49.87	67.56
GBPNet	Conv-64F	52.32	67.90

Table 2. Accuracy in few-shot classification on tieredImageNet

Model	backbone	5-way 1-shot	5-way 5-shot
MN [18]	Conv-64F	42.10	50.04
SSL [34]	Conv-64F	52.93	71.71
RN [20]	Conv-64F	53.18	69.38
MAM [26]	Conv-64F	49.56	70.30
PN [19]	Conv-64F	50.32	69.42
Reptile [27]	Conv-64F	52.36	71.03
FEAT [35]	Conv-64F	51.93	69.58
CGRN [22]	Conv-64F	53.54	70.53
BOIL [37]	Conv-64F	49.35	69.37
ECMT [38]	Conv-64F	48.19	65.50
LMPNet [39]	Conv-64F	47.50	72.16
DCAP [40]	Conv-64F	53.51	70.78
GBPNet	Conv-64F	53.78	71.24

To ensure a fair comparison with the most classical networks, we reimplemented the above methods based on the experimental settings reported in the miniImageNet and tieredImageNet papers. The experimental results can be found

in Tables 1 and 2. For the miniImageNet dataset, in the 5-way 1-shot and 5-way 5-shot configurations, the parameter was set to epoch×0.005 and epoch/50×0.2, respectively. For the tieredImageNet dataset, was set to epoch×0.005 and epoch/20×0.1, respectively. The parameters were adjusted with increasing episodes to obtain the optimal values. The experimental results are shown in Table 1 and Table 2.

4.4. Ablation experiment

Directly incorporating the granular-ball algorithm into few-shot classification yielded suboptimal accuracy, prompting us to adapt it for the few-shot classification task. ①The initial spherical clusters are divided based on prototypes rather than using the original k-means method. ②During each iterative computation of the cluster centroids, the feature vectors of centroids from the previous iteration are incorporated. ③ When determining whether a sample point should move to a neighbor cluster, correlation is used instead of comparing Euclidean distances. As Table 3 shown, the following ablation experiments were conducted on miniImageNet.

Table 3. Accuracy in few-shot classification on MiniImagenet as each method is incrementally integrated into the model

Model	backbone	5-way 1-shot	5-way 5-shot
Ball K-means	Conv-64F	44.62	61.58
Ball K-means+②	Conv-64F	44.67	61.12
Ball K-means+③	Conv-64F	46.43	62.47
Ball K-means+①	Conv-64F	49.56	67.47
Ball K-means+①+②	Conv-64F	51.68	67.37
Ball K-means+①+③	Conv-64F	51.15	67.17
Ball K-means+①+②+③	Conv-64F	52.32	67.90

4.5. Efficiency and accuracy test

We also compared the efficiency and accuracy of the original algorithm with our modified version. Efficiency was assessed by recording the number of iterations and time required for each centroid to stabilize, while accuracy was compared based on the final classification results. For example, in the 5-way 5-shot setting, the experimental results are shown in the Table 4.

Table 4. Comparison of Efficiency and Accuracy on miniImageNet

Model	t	time	Acc
Ball k-menas	4,5	0.1s	60.99
GBPNet	2,3	0.06s	67.90

4.6. Robustness test

During the feature extraction process, images are transformed into feature vectors, but this process is prone to noise, which can introduce errors in the extracted features and lead to a significant decline in the accuracy of classification models, thereby compromising robustness. To assess the robustness of our proposed model, we introduced noise to the data. We extracted one sample point per class from the support set and added a fixed value to its feature vector, turning these samples into outliers for training. This approach allowed us to evaluate the robustness of our proposed method in comparison to the prototype network.

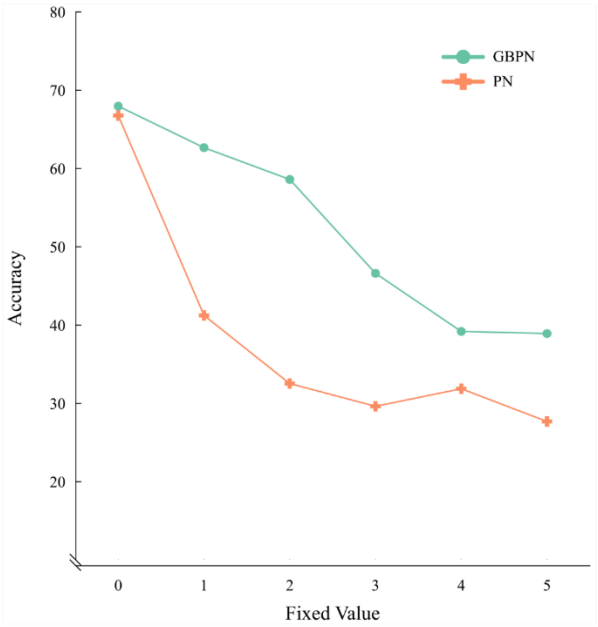

Fig. 6 Comparative robustness

In the line chart, both our model and the prototype network exhibit a downward trend in accuracy as the fixed value increases. The rate of decline effectively demonstrates the robustness of both models under the influence of noise. The prototype network shows a steep decline in accuracy with increasing noise, whereas our proposed model also decreases but at a more gradual slope, indicating better robustness in the granular-ball prototype network.

As Fig. 6 shown that this difference arises because the prototype network is constructed by averaging the support set sample points, which means that outliers can skew the prototype, adversely affecting the classification capability of the network. On the other hand, although the granular-ball prototype network uses prototypes to initialize the spherical clusters of the query set, the cluster centroids are adaptively updated through subsequent iterations, reducing the impact of outliers introduced during initial partitioning. This adaptive updating strengthens the granular-ball prototype and is the key reason it demonstrates better robustness than the prototype network.

5. Conclusion

In this study, We integrate the ball k-means algorithm with image processing using a few-shot learning approach. The encoder module maps input samples into a low-dimensional embedding space, then preliminarily divides the query set samples into spherical clusters based on prototypes. For each spherical cluster, neighbor clusters are identified, active and stable regions are delineated. By iteratively updating the centroids through adaptive movement of samples between neighbor clusters within the active region, the centroids are adjusted until they reach a stable state. These centroids are then fused with the prototypes to form a robust granular-ball prototype that accurately represents the features of each class in the query set. Ultimately, the query set is classified according to the granular-ball prototype using the nearest-neighbor classification method. Experimental results demonstrate that our method delivers superior classification performance.

In our future work plan, we intend to integrate the ball k-means algorithm with the field of image segmentation. Initially, we will perform preliminary clustering on pixel or

superpixel features, where each spherical cluster can represent different regions or segments within the image. Stable regions can be directly associated with distinct segmentation parts, while active regions will aid in refining segmentation boundaries. Iterative updates to the cluster centroids will enhance the accuracy and quality of the segmentation.

References

[1] Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. Electronics, 9(8), 1295. https://doi.org/10.3390/electronics9081295

[2] Chavan, M., Patil, A., Dalvi, L., & Patil, A. (2015). Mini batch k-means clustering on large dataset. International Journal of Science, Engineering and Technology Research, 4(7), 1356-1358.

[3] Nie, F., Li, Z., Wang, R., & Li, X. (2023). An effective and efficient algorithm for k-means clustering with new formulation. IEEE Transactions on Knowledge and Data Engineering, 35(4), 3433-3443. https://doi.org/10.1109/TKDE.2022.3142576

[4] Zhu, X., Sun, J., He, Z., Jiang, J., & Wang, Z. (2023). Staleness-reduction mini-batch k-means. IEEE Transactions on Neural Networks and Learning Systems, 34(12), 10247-10258. https://doi.org/10.1109/TNNLS.2023.3262814

[5] Xia, S., Peng, D., Meng, D., Zhang, C., Wang, G., Giem, E., Wei, W., & Chen, Z. (2022). Ball k-means: Fast adaptive clustering with no bounds. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(1), 87-99. https://doi.org/10.1109/TPAMI.2020.3038692

[6] Xia, S., Liu, Y., Ding, X., Wang, G., Yu, H., & Luo, Y. (2019). Granular ball computing classifiers for efficient, scalable and robust learning. Information Sciences, 483, 136-152. https://doi.org/10.1016/j.ins.2019.01.010

[7] Xia, S., Zheng, S., Wang, G., Gao, X., & Wang, B. (2023). Granular ball sampling for noisy label classification or imbalanced classification. IEEE Transactions on Neural Networks and Learning Systems, 34(4), 2144-2155. https://doi.org/10.1109/TNNLS.2021.3111362

[8] Xia, S., Dai, X., Wang, G., Gao, X., & Giem, E. (2022). An efficient and adaptive granular-ball generation method in classification problem. IEEE Transactions on Neural Networks and Learning Systems, 33(10), 5610-5624. https://doi.org/10.1109/TNNLS.2022.3161503

[9] Xie, Q., Zhang, Q., Xia, S., Zhao, F., Wu, C., Wang, G., & Ding, W. (2024). GBG++: A fast and stable granular ball generation method for classification. IEEE Transactions on Emerging Topics in Computational Intelligence, 8(3), 2255-2268. https://doi.org/10.1109/TETCI.2024.3372653

[10] Song, Y., Wang, T., Cai, P., Mondal, S. K., & Sahoo, J. P. (2023). A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. ACM Computing Surveys, 55(13s), Article 271. https://doi.org/10.1145/3582688

[11] Wang, J., Liu, K., Zhang, Y., Leng, B., & Lu, J. (2023). Recent advances of few-shot learning methods and applications. Science China Technological Sciences, 66(4), 920-944. https://doi.org/10.1007/s11431-022-2323-6

[12] Xia, S., Wang, G., & Gao, X. (2023). Granular ball computing: An efficient, robust, and interpretable adaptive multi-granularity representation and computation method. arXiv, arXiv:2304.11171. https://doi.org/10.48550/arXiv.2304.11171

[13] Xia, S., Wu, S., Chen, X., Wang, G., Gao, X., Zhang, Q., Giem, E., & Chen, Z. (2022). GRRS: Accurate and efficient neighborhood rough set for feature selection. IEEE Transactions on Knowledge and Data Engineering, 34(11), 5427-5440. https://doi.org/10.1109/TKDE.2022.3166428

[14] Xia, S., Wang, C., Wang, G., Gao, X., Ding, W., Yu, J., Zhai, Y., & Chen, Z. (2023). GBRS: A unified granular-ball learning model of Pawlak rough set and neighborhood rough set. IEEE Transactions on Neural Networks and Learning Systems, 34(9), 6298-6311. https://doi.org/10.1109/TNNLS.2022.3172692

[15] Li, X., Yang, X., Ma, Z., & Xue, J.-H. (2023). Deep metric learning for few-shot image classification: A review of recent developments. Pattern Recognition, 138, 109381. https://doi.org/10.1016/j.patcog.2023.109381

[16] Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys, 53(3), Article 63. https://doi.org/10.1145/3386252

[17] Bellet, A., Habrard, A., & Sebban, M. (2022). Metric learning. Springer Nature. https://doi.org/10.1007/978-3-031-05964-1

[18] Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., & Kavukcuoglu, K. (2016). Matching networks for one shot learning. Advances in Neural Information Processing Systems, 29, 3637-3645.

[19] Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. Advances in Neural Information Processing Systems, 30, 4077-4087.

[20] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S., & Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1199-1208). IEEE. https://doi.org/10.1109/CVPR.2018.00131

[21] 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. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7260-7268). IEEE. https://doi.org/10.1109/CVPR.2019.00743

[22] Jia, X., Su, Y., & Zhao, H. (2023). Few-shot learning via relation network based on coarse-grained granulation. Applied Intelligence, 53(1), 996-1008. https://doi.org/10.1007/s10489-022-03650-y

[23] Jia, X., Mao, Y., Pan, Z., Wang, Z., & Ping, P. (2024). Few-shot learning based on hierarchical feature fusion via relation networks. International Journal of Approximate Reasoning, 170, 109186. https://doi.org/10.1016/j.ijar.2024.109186

[24] Vanschoren, J. (2019). Meta-learning. In Automated machine learning: Methods, systems, challenges (pp. 35-61). Springer. https://doi.org/10.1007/978-3-030-05318-5_2

[25] Wang, J. X. (2021). Meta-learning in natural and artificial intelligence. Current Opinion in Behavioral Sciences, 38, 90-95. https://doi.org/10.1016/j.cobeha.2020.10.010

[26] Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In International Conference on Machine Learning (pp. 1126-1135). PMLR.

[27] Nichol, A., Achiam, J., & Schulman, J. (2018). On first-order meta-learning algorithms. arXiv, arXiv:1803.02999. https://doi.org/10.48550/arXiv.1803.02999

[28] Antoniou, A., Edwards, H., & Storkey, A. (2019). How to train your MAML. In Seventh International Conference on Learning Representations (ICLR 2019).

[29] Jamal, M. A., & Qi, G.-J. (2019). Task agnostic meta-learning for few-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11719-11727). IEEE. https://doi.org/10.1109/CVPR.2019.01199

[30] Rajeswaran, A., Finn, C., Kakade, S. M., & Levine, S. (2019). Meta-learning with implicit gradients. Advances in Neural Information Processing Systems, 32, 113-124.

[31] Rajasegaran, J., Khan, S., Hayat, M., Khan, F. S., & Shah, M. (2020). iTAML: An incremental task-agnostic meta-learning approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13588-13597). IEEE. https://doi.org/10.1109/CVPR42600.2020.01360

[32] Kaya, M., & Bilge, H. Ş. (2019). Deep metric learning: A survey. Symmetry, 11(9), 1066. https://doi.org/10.3390/sym11091066

[33] Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2022). Meta-learning in neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(9), 5149-5169. https://doi.org/10.1109/TPAMI.2021.3079209

[34] Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., & Zemel, R. S. (2018). Meta-learning for semi-supervised few-shot classification. In 6th International Conference on Learning Representations (ICLR 2018).

[35] Ye, H.-J., Hu, H., Zhan, D.-C., & Sha, F. (2020). Few-shot learning via embedding adaptation with set-to-set functions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8808-8817). IEEE. https://doi.org/10.1109/CVPR42600.2020.00883

[36] Patacchiola, M., Turner, J., Crowley, E. J., O'Boyle, M., & Storkey, A. J. (2020). Bayesian meta-learning for the few-shot setting via deep kernels. Advances in Neural Information Processing Systems, 33, 16108-16118.

[37] Oh, J., Yoo, H., Kim, C., & Yun, S.-Y. (2020). BOIL: Towards representation change for few-shot learning. arXiv, arXiv:2008.07408. https://doi.org/10.48550/arXiv.2008.07408

[38] Ravichandran, A., Bhotika, R., & Soatto, S. (2019). Few-shot learning with embedded class models and shot-free meta training. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 331-339). IEEE. https://doi.org/10.1109/ICCV.2019.00042

[39] Huang, H., Wu, Z., Li, W., Huo, J., & Gao, Y. (2021). Local descriptor-based multi-prototype network for few-shot learning. Pattern Recognition, 116, 107935. https://doi.org/10.1016/j.patcog.2021.107935

[40] He, J., Hong, R., Liu, X., Xu, M., & Sun, Q. (2022). Revisiting local descriptor for improved few-shot classification. ACM Transactions on Multimedia Computing, Communications and Applications, 18(2s), Article 53. https://doi.org/10.1145/3472815