Clustered-patch Element Connection for Few-shot Learning

Jinxiang Lai[1], **Siqian Yang**[1], **Junhong Zhou**[2], **Wenlong Wu**[1], **Xiaochen Chen**[1], **Jun Liu**[1], **Bin-Bin Gao**[*1], **Chengjie Wang**[*1,3]

[1]Tencent Youtu Lab, China
[2]Southern University of Science and Technology, China
[3]Shanghai Jiao Tong University, China
layjins1994@gmail.com, {seasonsyang, ezrealwu, husonchen}@tencent.com,
12011801@mail.sustech.edu.cn, {junsenselee, csgaobb}@gmail.com, jasoncjwang@tencent.com

Abstract

Weak feature representation problem has influenced the performance of few-shot classification task for a long time. To alleviate this problem, recent researchers build connections between support and query instances through embedding patch features to generate discriminative representations. However, we observe that there exists semantic mismatches (foreground/ background) among these local patches, because the location and size of the target object are not fixed. What is worse, these mismatches result in unreliable similarity confidences, and complex dense connection exacerbates the problem. According to this, we propose a novel Clustered-patch Element Connection (CEC) layer to correct the mismatch problem. The CEC layer leverages Patch Cluster and Element Connection operations to collect and establish reliable connections with high similarity patch features, respectively. Moreover, we propose a CECNet, including CEC layer based attention module and distance metric. The former is utilized to generate a more discriminative representation benefiting from the global clustered-patch features, and the latter is introduced to reliably measure the similarity between pair-features. Extensive experiments demonstrate that our CECNet outperforms the state-of-the-art methods on classification benchmark. Furthermore, our CEC approach can be extended into few-shot segmentation and detection tasks, which achieves competitive performances.

1 Introduction

In contrast to general deep learning task [Krizhevsky *et al.*, 2012], *Few-Shot Learning* (FSL) aims to learn a transferable classifier with amount seen images (base class) and few labeled unseen images (novel class). Due to the lack of effective features from unseen classes, a robust feature embedding model is indispensable. Recent researchers[Hou *et al.*, 2019; Rizve *et al.*, 2021; Xu *et al.*, 2021a] manage to design an embedding network for generating more discriminative features.

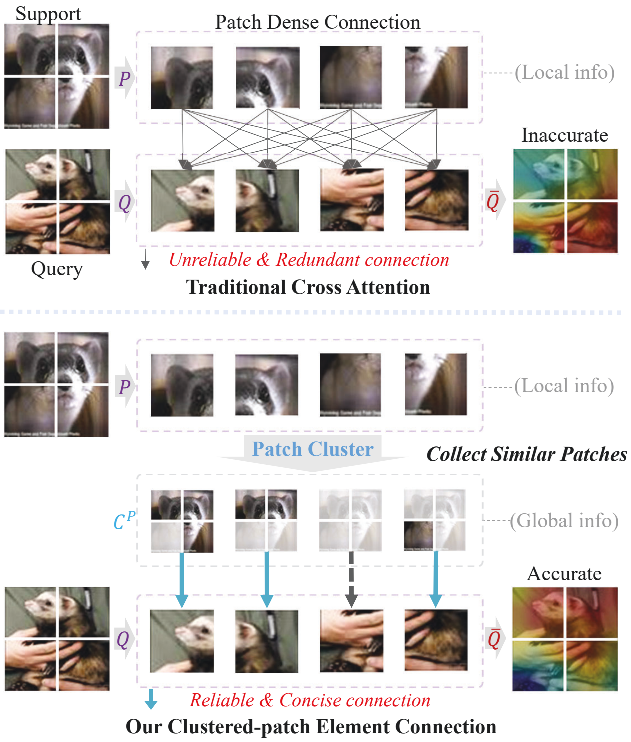


Figure 1: Comparison between traditional Cross Attention and our Clustered-patch Element Connection. The proposed Clustered-patch Element Connection, which utilizes the global info C^p integrated from support feature P to perform element connection with query Q leading to a confident and clear connection, is able to generate a more clear and precise relation map than Cross Attention. The detailed Patch Cluster operation is illustrated in Fig.2. The visualization comparisons are referred to Fig.4(a).

Specifically, cross attention based methods [Hou *et al.*, 2019; Xu *et al.*, 2021a; Xu *et al.*, 2021b] focus on reducing the background noise and highlighting the target region to generate more discriminative representations. The core idea of these methods is to divide extracted features into patches and connect all local patch features. However, as shown in Fig.1, we observe that the target object may be located randomly with different scales among the query images. Hence, these methods suffer two main problems: inconsistent seman-

[*]Corresponding Author

tic in feature space, and unreliable and redundant connections. To tackle these problems, we propose a Clustered-patch Element Connection (CEC) layer which consists of Patch Cluster and Element Connection operations. In detail, given inputs features P (support) and Q (query), CEC layer firstly obtains the global clustered-patch C^p features by Patch Cluster operation as illustrated in Fig.2, then performs Element Connection on Q by using C^p, and finally produces a more discriminative representation $\bar{Q}$. Patch Cluster aims to collect the objects in source feature P that are similar to the reference patch in Q, which adaptively alignments the P into C^P to obtain a consistent semantic feature for each patch of Q. Then, with the global clustered-patch features, CEC layer generates more reliable and concise connections than cross attention.

According to CEC layer, we find the key of generating accurate relation map is to obtain appropriate clustered-patch features. In this paper, four solutions are introduced to perform Patch Cluster, including MatMul, Cosine, GCN and Transformer. Different from the naive MatMul and Cosine modes, we propose the meta-GCN and Transformer based Patch Cluster operations to obtain a more robust clustered-patch by implementing additional feature refinement. The insight of meta-GCN is constructing a dynamic correlation-based adjacent for each current input pair-features, other than the static GCN [Kipf and Welling, 2017] using a fixed adjacent. Besides, the transformer structure obtains global information via modeling a spatio-temporal correlation among instances, which generates a more accurate relation map.

Along with the description of CEC mechanism, we propose three CEC-based modules: (I) The Clustered-patch Element Connection Module (CECM) distinguishes the background and the object for each image pair (support and query) at the feature level adaptively, which gives a more precise highlights at the regions of target object; (II) The Self-CECM enhances the semantic feature of target object in a self-attention manner to make the representation more robust; (III) The Clustered-patch Element Connection Distance (CECD) is a CEC-based distance metric which measures the similarity between pair-features via the obtained reliable relation map.

For few-shot classification task, we introduce a novel Clustered-patch Element Connection Network (CECNet) as illustrated in Fig.3, which learns a generalize-well embedding benefiting from auxiliary tasks, generates a discriminative representation via CECM and Self-CECM, and measures a reliable similarity map via CECD. Furthermore, we derive a novel CEC-based embedding module named CEC Embedding (CECE), which can be applied into few-shot semantic segmentation (FSSS) and few-shot object detection (FSOD) tasks. We simply stack the proposed CECE after the backbone network of the existing FSSS and FSOD methods, which achieves consistent improvements around $1\% - 3\%$.

To summarize, our main contributions are:

- We propose a Clustered-patch Element Connection (CEC) layer to strengthen the target regions of query features by element-wisely connecting them with the global clustered-patch features. Four different CEC modes are introduced, including MatMul, Cosine, GCN and Transformer.
- We derive three CEC-based modules: CECM and Self-CECM modules are utilized to produce more discriminative representations, and CECD is able to measure a reliable similarity map.
- With CEC-based modules and auxiliary tasks, a novel CECNet model is designed for few-shot classification. CECNet improves state-of-the-arts on few-shot classification benchmark, and the experiments demonstrate that our method is effective in FSL.
- Furthermore, our CECE (i.e. CEC-based embedding module) can be extended into few-shot segmentation and detection tasks, which achieves performance improvements around $1\% - 3\%$ on the corresponding benchmarks.

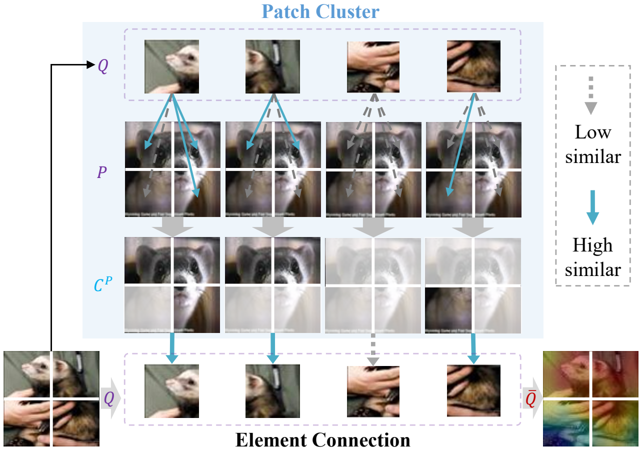


Figure 2: Patch Cluster and Element Connection.

2 Related Work

Few-Shot Learning The FSL algorithms aim to recognize novel categories with few labeled images, and a category-disjoint base set with abundant images is provided for pre-training. The classic FSL tasks include few-shot classification [Finn *et al.*, 2017; Vinyals *et al.*, 2016; Snell *et al.*, 2017; Hou *et al.*, 2019; Tian *et al.*, 2020], semantic segmentation [Zhang *et al.*, 2020b; Siam *et al.*, 2019; Malik *et al.*, 2021] and object detection [Kang *et al.*, 2019; Wang *et al.*, 2020; Qiao *et al.*, 2021]. More introductions are presented in APPENDIX. In a word, the existing FSL methods lack a uniform function to control the connections among the patches between support and query instances semantically.

Other Related Works are introduced in APPENDIX, such as **Auxiliary Task for FSL** [Hou *et al.*, 2019; Rizve *et al.*, 2021], **Graph Convolutional Network (GCN)** [Bruna *et al.*, 2013], and **Transformer** [Vaswani *et al.*, 2017].

3 Problem Definition

3.1 Few-Shot Classification

A classic few-shot classification problem is specified as a N-way K-shot task, which means solving a N-class classification problem with only K labeled instances provided per class. In the recent investigations[Hou *et al.*, 2019; Snell *et al.*, 2017], the source dataset is divided into three category-disjoint parts: training set X_{train}, validation set X_{val} and test set X_{test}. Moreover, the episodic training mechanism is widely adopted. An episode consists of two sets (randomly sampling in N categories): support and query. Let $\mathcal{S} = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ ($n_s = N \times K$) denote the support set, and $\mathcal{Q} = \{(x_i^q, y_i^q)\}_{i=1}^{n_q}$ denote the query set. Note that n_s and n_q are the size of corresponding sets. Especially, $\mathcal{S} = \{\mathcal{S}^1, \mathcal{S}^2, ..., \mathcal{S}^k\}$, where $\mathcal{S}^k$ denotes the support set of the k^{th} category in $\mathcal{S}$. Specifically, let (P, Q) $\in \mathbb{R}^{hw \times c}$ denote the support and query features, which are extracted from support subset and query instance ($\mathcal{S}^k$, x^q). Note that c, h, w are channel number, height, width of features, respectively.

3.2 Cross Attention

The traditional Cross Attention [Hou *et al.*, 2019] proves that highlighting target regions could generate more discriminative representations, leading to accuracy improvements for FSL. The key is to generate an fine-grained relation map $R^Q \in \mathbb{R}^{hw}$ to represent the target regions in $Q \in \mathbb{R}^{hw \times c}$. Then, a spatial-wise feature attention can be obtained through $R^Q \odot Q$, where $\odot$ is the Element-wise Product. The traditional Cross Attention produces relation map R^Q for Q by:

$$R^Q = h\left(\frac{P}{||P||_2}\left(\frac{Q}{||Q||_2}\right)^T\right), \tag{1}$$

where h is a CNN-based layer to refine the correlation matrix $\frac{P}{||P||_2}(\frac{Q}{||Q||_2})^T \in \mathbb{R}^{hw \times hw}$. According to Eq.1, the Cross Attention produces relation map by *Local-to-Local* fully connection among local feature patches of P and Q. In detail, $(P_i, Q_j) \in \mathbb{R}^{1 \times c}$ represent a pair of support feature patch and query feature patch among $(P, Q) \in \mathbb{R}^{hw \times c}$. As shown in Fig.1, the target object may be located unregularly among the query images at different scale, which results in inconsistent semantic in feature space, i.e feature patches P_i and Q_j may be semantically inconsistent. This *semantically inconsistent problem* causes low confident correlation between patches, and the complex Local-to-Local fully connection further accumulates this inaccurate bias, which affect the quality of the generated relation map.

To establish concise and clear connections among global and local features, we propose a Clustered-patch Element Connection layer (CEC), which consists of two key operations: *Patch Cluster* and *Element Connection.*

4 Clustered-patch Element Connection

4.1 Patch Cluster

As illustrated in Fig.2, Patch Cluster operation obtains a set C^p, named Clustered-patch, via collecting those objects in support feature set P, which are similar to the reference patch in Q. We define a generic Patch Cluster operation f_{PC} as:

$$C^p = f_{PC}(Q, P) = \phi\left(g\left(Q, P\right)P\right). \tag{2}$$

Here P is the input source feature, Q is the input reference feature, and $C^p \in \mathbb{R}^{hw \times c}$ is the output Clustered-patch. A pairwise function g computes an affinity matrix representing relationship between Q and P. The clustered patches can be refined by function ϕ. In detail, we divide the source image into $w \times h$ patches. Here, w and h are the same as the size of the features in P, which is convenient for element connection operation. A Clustered-patch $C^p \in \mathbb{R}^{hw \times c}$ collects $w \times h$ clusters. Each cluster collects the patch-features in $P = [P_1, P_2, ..., P_{hw}] \in \mathbb{R}^{hw \times c}$ that are similar to the corresponding patch-feature in the reference patch-features in Q. Therefore, C^p is semantically similar to Q.

To implement the Patch Cluster operation, we give four solutions including MatMul, Cosine, GCN and Transformer.

MatMul A simplest way to obtain the clustered patches is treating MatMul operation as the pairwise function g (in Eq.(2)) and not implementing any further embedding refinement. Formally,

$$C^p = \sigma\left(QP^T\right)P, \tag{3}$$

where σ is softmax function.

Cosine A simple extension of the MatMul version is to compute cosine similarity in feature space. Formally,

$$C^p = \sigma\left(\frac{Q}{||Q||_2}\left(\frac{P}{||P||_2}\right)^T\right)P. \tag{4}$$

GCN GCN [Kipf and Welling, 2017] updates the input features P via utilizing a pre-defined adjacent matrix $A \in \mathbb{R}^{hw \times hw}$ and a learnable weight matrix $W \in \mathbb{R}^{c \times c}$. Formally, the updated features $G^p \in \mathbb{R}^{hw \times c}$ can be expressed as: $G^p = \delta(APW)$, where $\delta(\cdot)$ is the nonlinear activation function ($Sigmoid(\cdot)$ or $ReLU(\cdot)$). However, the adjacent matrix A used in GCN is fixed for all inputs after training, which is not able to recognize the new categories in few-shot task. Comparing Eq.(2) and the definition of GCN, we observe that the affinity matrix $g\left(Q, P\right)$ can be considered as the adjacent matrix A, because they all try to describe the relationship between features P and Q. Hence, we derive a meta-GCN through replacing the static adjacent matrix with the dynamic affinity matrix. Formally, the meta-GCN based Patch Cluster operation is derived as follows:

$$C^p = \delta\left[\sigma\left(\frac{Q}{||Q||_2}\left(\frac{P}{||P||_2}\right)^T\right)PW\right]. \tag{5}$$

Transformer The Transformer[Vaswani *et al.*, 2017] based Patch Cluster operation is defined as follows:

$$C^p = FFN\{\sigma[(W_q Q)(W_k P^T)]W_v P\}, \tag{6}$$

where, FFN is the Feed-Forward Network in transformer, W_q, W_k, W_v are learnable weights (e.g. convolution layers).

4.2 Element Connection

According to the global semantic features C^p obtained from Patch Cluster operation, element Connection operation generates the relation map R^Q for Q by simply computing the patch-wise cosine similarity between Q and C^p. Finally, we obtain a rectified discriminative representation by the Element Connection operation f_{EC}:

$$\begin{gathered}\bar{Q} = f_{EC}(Q, C^p) = \left(\sigma\left(R^Q\right) + 1\right) \odot Q, \\ where, \quad R^Q = \left(\frac{Q}{||Q||_2} \otimes \frac{C^p}{||C^p||_2}\right) \in \mathbb{R}^{hw},\end{gathered} \tag{7}$$

where, $\otimes$ is Patch-wise Dot Product, $\odot$ is Element-wise Product. The n^{th} position of R^Q is $R_n^Q = \frac{Q_n}{||Q_n||_2} \cdot \frac{C_n^p}{||C_n^p||_2}$, where $\cdot$

is Dot Product. The visualizations of the CEC-based relation map R^Q are shown at the last column in Fig.4(b). Overall, the Clustered-patch Element Connection (CEC) layer is able to highlight the regions of Q that are semantically similar to P. Formally, CEC layer f_{CEC} is expressed as:

$$\bar{Q} = f_{CEC}(Q, P) = f_{EC}\left(Q, f_{PC}(Q, P)\right). \tag{8}$$

4.3 Discussion

Compared with traditional Cross Attention, the key point of our Clustered-patch Element Connection is to perform the Global-to-Local element connection between the Clustered-patch C^p (global) and query Q (local). It is able to generate a more clear and precise relation map, as shown in Fig. 4(a) visualizations. As demonstrated in Tab. 2, our CEC-based approach achieves 4% accuracy improvement than the traditional Cross Attention based CAN [Hou *et al.*, 2019].

Generally, the advantages of our Clustered-patch Element Connection are: (I) The relation map generated by Element Connection is more confident than Cross Attention, because the global Clustered-patch feature C^p is more stable and representative than the local feature P. (II) Element Connection (1-to-1 patch-connection) has more clear connection relationship than Cross Attention (1-to-hw patch-connection).

Moreover, the respective advantages of different solutions for realizing Patch Cluster are: (I) These four solutions can be divided into two groups: fixed (i.e. MatMul and Cosine) and learnable (i.e. GCN and Transformer) solutions. The fixed solutions can be used to perform patch clustering without additional learnable parameters, while the learnable solutions are data-driven to refine the affinity matrix or clustered-patch. (II) According to experimental results in Tab. 2, the learnable solutions are better than the fixed ones when they are applied as a embedding layer for feature enhancing (i.e. CECM defined in Eq. 9), which indicates that the learnable solutions can generate better embedding features. In contrast, according to Tab. 3, the fixed solutions are better than the learnable ones when they are applied as the distance metric for measuring similarity (i.e. CECD defined in Eq. 11), which indicates fixed solutions can obtain more reliable similarity scores.

5 CEC Network for Few-Shot Classification

5.1 CEC Module and Self-CEC Module

According to the CEC layer mentioned above, we propose two derivative modules: the CEC Module (CECM) and the Self-CEC Module (self-CECM). The CECM is able to highlight the mutual similar regions via learning the semantic relevance between pair feature. Specifically, CECM transfers the input pair-features (P, Q) $\in \mathbb{R}^{hw \times c}$ into more discriminative representations ($\bar{P}$, $\bar{Q}$) $\in \mathbb{R}^{hw \times c}$. Formally, its function f_{CECM} is expressed as:

$$\begin{aligned} (\bar{Q}, \bar{P}) &= f_{CECM}(Q, P), \\ where, \quad \bar{Q} &= f_{CEC}(Q, P), \quad \bar{P} = f_{CEC}(P, Q). \end{aligned} \tag{9}$$

The Self-CECM enhances the semantic feature of target object via self-connection, which turns the input Q into $\bar{\bar{Q}} \in \mathbb{R}^{hw \times c}$. Formally, Self-CECM function f_{SCECM} is expressed as:

$$\bar{\bar{Q}} = f_{SCECM}(Q) = f_{CEC}(Q, Q). \tag{10}$$

The CECM exploit the relation between P and Q via $\bar{Q} = f_{CEC}(Q, P)$, while Self-CECM exploit the relation between the input itself via $\bar{\bar{Q}} = f_{CEC}(Q, Q)$, i.e. Self-CECM explores the relation between the patches of input image. Because we assume that patch-features of the target are mutually similar, Self-CECM can enhance the target region by clustering the similar regions.

5.2 CECNet Framework

Then, we give the overall Clustered-patch Element Connection Network (CECNet). The framework is shown in Fig.3, which integrates CECM, Metric Classifier and Fine-tune Classifier for few-shot classification task, and Rotation Classifier and Global Classifier for the auxiliary tasks. The network involves three stages: Base Training, Novel Fine-tuning and Novel Inference.

Base Training As illustrated in Fig.3, every image x^q in query set $\mathcal{Q} = \{(x_i^q, y_i^q)\}_{i=1}^{n_q}$ is rotated with [0°, 90°, 180°, 270°] and outputs a rotated $\tilde{\mathcal{Q}} = \{(\tilde{x}_i^q, \tilde{y}_i^q)\}_{i=1}^{n_q \times 4}$. The support subset $\mathcal{S}^k$ and the rotated query instance $\tilde{x}^q$ are processed by the embedding f_θ and produces the prototype feature $P^k = \frac{1}{|\mathcal{S}^k|} \sum_{x_i^s \in \mathcal{S}^k} f_\theta(x_i^s)$ and query feature $Q = f_\theta(\tilde{x}^q) \in \mathbb{R}^{c \times h \times w}$, respectively. Then each pair-features (P^k, Q) are processed via CECM to enhance the mutually similar regions and generates more discriminative features ($\bar{P}^k$, $\bar{Q}^k$) for the subsequent classification. Note that the inputs and outputs of CECM will be reshaped to satisfied its format. Finally, CECNet is optimized via multi-task loss contributing from metric classifier and auxiliary tasks.

Novel Fine-tuning The Fine-tune Classifier consists of Self-CECM and a linear layer as shown in Fig.3. In fine-tuning phase, the pre-trained embedding f_θ is frozen, and the Fine-tune Classifier is optimized with cross-entropy loss.

Novel Inference In inductive inference, the overall prediction of CECNet is $Y = Y_M + Y_F$, where Y_M and Y_F are the results of Metric and Fine-tune Classifiers respectively.

5.3 Metric Classifier

As illustrated in Eq. 7, the proposed CEC layer is able to generate a reliable relation map R^Q. The relation map R^Q can also be utilized as a similarity map, and the mean of R^Q is the similarity score. Therefore, we obtain the CECD distance metric d_{CECD} which is expressed as:

$$d_{CECD}(\bar{Q}, \bar{P}) = \left(\frac{\bar{Q}}{||\bar{Q}||_2} \otimes \frac{C^{\bar{p}}}{||C^{\bar{p}}||_2} \right) \in \mathbb{R}^{hw}. \tag{11}$$

With the proposed CECD distance metric, the Metric Classifier make predictions by measuring the similarity between the query and the N support classes. Following [Hou *et al.*, 2019], the patch-wise classification strategy is used to produce precise feature representations. In detail, each patch-wise feature $\bar{Q}_n^k$ at n^{th} spatial position of $\bar{Q}^k$, is recognized as N classes. And the probability of predicting $\bar{Q}_n^k$ as k^{th}

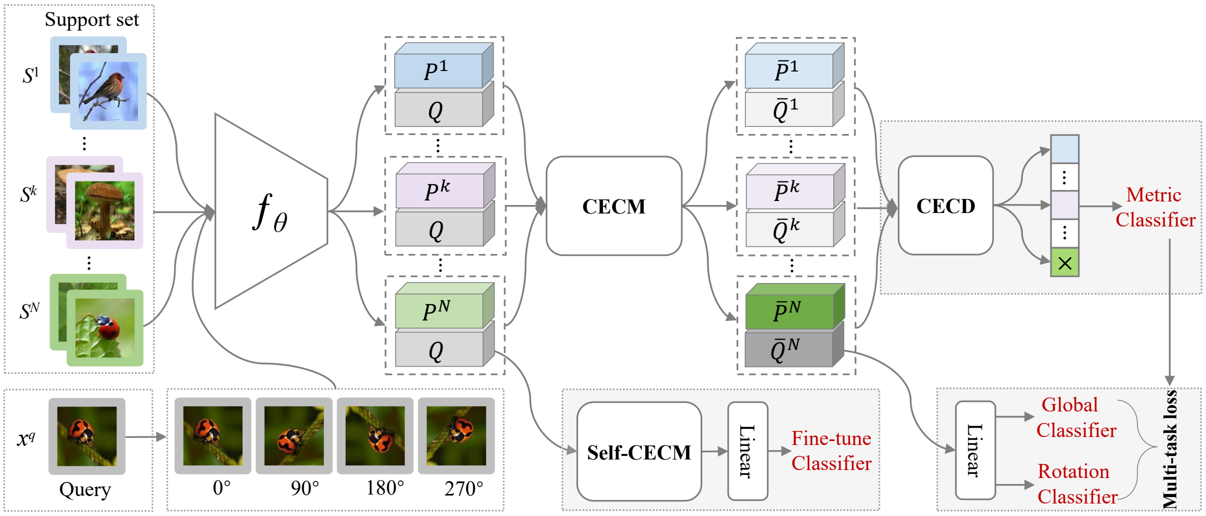


Figure 3: The proposed CECNet framework. The CECM is able to highlight the mutually similar regions, the CECD is utilized to measure similarity of pair-features. And Self-CECM enhances the semantic feature of target object via self-connection.

class is:

$$\hat{Y}(y=k|\bar{Q}_n^k) = \frac{\exp(R_n^k)}{\sum_{i=1}^{N}\exp(R_n^i)}, \quad (12)$$
$$where, \quad R^k = d_{CECD}(\bar{Q}^k, \bar{P}^k) \in \mathbb{R}^{hw},$$

where, the similarity map R^k is obtained by the CECD distance metric formulated in Eq. 11, and the similarity score R_n^k is the n^{th} position of R^k.

5.4 Fine-tune Classifier

The Fine-tune Classifier consists of Self-CECM and a linear layer. It predicts the query feature $\bar{\bar{Q}}$ into N categories by a linear layer W_F. And its loss is computed as:

$$\begin{aligned}\mathcal{L}_F &= PCE\left(W_F(\bar{\bar{Q}}), N^q\right) \\ &= -\sum_{i=1}^{n_q}\sum_{n=1}^{h\times w} N_i^q \log\left(\sigma\left(W_F(\bar{\bar{Q}}_n)_i\right)\right),\end{aligned} \quad (13)$$

where, PCE is patch-wise cross-entropy, and N_i^q is the ground truth of x_i^q with N categories of few-shot task.

5.5 Objective functions in Base Training

Metric Loss The metric classification loss with the ground-truth few-shot label $\tilde{y}^q$ is:

$$\mathcal{L}_M = -\sum_{i=1}^{n_q}\sum_{n=1}^{h\times w}\log\hat{Y}(y=\tilde{y}_i^q|(\bar{Q}_n)_i). \quad (14)$$

Auxiliary Loss The loss of Global Classifier is $\mathcal{L}_G = PCE(W_G(\bar{Q}), D^q)$, where D_i^q is the global category of $\tilde{x}_i^q$ with D classes of train set, and W_G is a fully-connected layer. Similarly, the loss of Rotation Classifier is $\mathcal{L}_R = PCE(W_R(\bar{Q}), B^q)$, where B_i^q is the rotation category of $\tilde{x}_i^q$ with four classes, and W_R is a fully-connected layer.

Multi-Task Loss Then, inspired by [Jinxiang and Siqian, 2022], the overall loss is defined as:

$$\mathcal{L} = \frac{1}{2}\mathcal{L}_M + \sum_{j=G,R}\left((\lambda+w_j)\mathcal{L}_j + log\frac{1}{(\lambda+w_j)}\right), \quad (15)$$

where $w = \frac{1}{2\alpha^2}$ and α is a learnable variable. The hyper-parameter λ is utilized to balance the few-shot and auxiliary tasks, of which the influence is studied in Tab. 4.

6 Experiments on Few-Shot Classification

Datasets The two popular FSL classification benchmark datasets are *mini*ImageNet and *tiered*ImageNet, where detailed introductions are presented in APPENDIX.

Experimental Setup We report the mean accuracy by testing 2000 episodes randomly sampled from meta-test set. According to Tab. 4, the hyperparameter λ is set to 1.0 and 2.0 for ResNet-12 and WRN-28, respectively. Other implementation details can be found in our public code.

6.1 Comparison with State-of-the-arts

As shown in Tab.1, we compare with the state-of-the-art few-shot methods on miniImageNet and tieredImageNet datasets. It shows that our CECNet outperforms the existing SOTAs, which demonstrates the effectiveness and strength of our CEC based methods. Different from existing metric-based methods [Zhang *et al.*, 2020a; Yang *et al.*, 2022; Jiangtao *et al.*, 2022] extracting support and query features independently, our CECNet enhances the semantic feature regions of mutually similar objects and obtains more discriminative representations. Comparing to the metric-based Meta-DeepBDC [Jiangtao *et al.*, 2022], CECNet achieves 1.98% higher accuracy on 1-shot. Some metric-based methods [Xu *et al.*, 2021a; Hou *et al.*, 2019] apply cross attention, while our CECNet still surpasses DANet [Xu *et al.*, 2021a] with an accuracy improvement up to 2.36% under WRN-28 backbone, which demonstrates the strength of our Clustered-patch Element Connection.

Model	Backbone	miniImageNet		tieredImageNet	
		1-shot	5-shot	1-shot	5-shot
ProtoNet [Snell *et al.*, 2017]	Conv4	49.42 ± 0.78	68.20 ± 0.66	53.31 ± 0.89	72.69 ± 0.74
Our CECNet	Conv4	**54.45** ± **0.47**	**70.57** ± **0.38**	**56.59** ± **0.50**	**72.86** ± **0.42**
CAN [Hou *et al.*, 2019]	ResNet-12	63.85 ± 0.48	79.44 ± 0.34	69.89 ± 0.51	84.23 ± 0.37
DeepEMD [Zhang *et al.*, 2020a]	ResNet-12	65.91 ± 0.82	82.41 ± 0.56	71.16 ± 0.87	86.03 ± 0.58
IENet [Rizve *et al.*, 2021]	ResNet-12	66.82 ± 0.80	84.35 ± 0.51	71.87 ± 0.89	86.82 ± 0.58
DANet [Xu *et al.*, 2021a]	ResNet-12	67.76 ± 0.46	82.71 ± 0.31	71.89 ± 0.52	85.96 ± 0.35
MCL [Yang *et al.*, 2022]	ResNet-12	67.36 ± 0.20	83.63 ± 0.20	71.76 ± 0.20	86.01 ± 0.20
Meta-DeepBDC [Jiangtao *et al.*, 2022]	ResNet-12	67.34 ± 0.43	84.46 ± 0.28	72.34 ± 0.49	**87.31** ± **0.32**
Our CECNet	ResNet-12	**69.32** ± **0.46**	**84.65** ± **0.32**	**73.14** ± **0.50**	86.88 ± 0.36
PSST [Zhengyu *et al.*, 2021]	WRN-28	64.16 ± 0.44	80.64 ± 0.32	-	-
DANet [Xu *et al.*, 2021a]	WRN-28	67.84 ± 0.46	82.74 ± 0.31	72.18 ± 0.52	86.26 ± 0.35
Our CECNet	WRN-28	**70.20** ± **0.46**	**85.00** ± **0.30**	**73.84** ± **0.50**	**87.36** ± **0.34**

Table 1: Comparing to existing approaches on 5-way FSL classification task on miniImageNet and tieredImageNet. Our CECNet adopts the proposed CECM(T) attention module, CECD(C) distance metric, and Self-CECM.

Model	Attention Module	Distance Metric	Param	miniImageNet	
				1-shot	5-shot
ProtoG	-	cosine	7.75M	61.87	78.87
CAN	CAM		7.75M	63.85	79.44
CECNet	CECM(M)	cosine	7.75M	67.69	81.84
	CECM(C)		7.75M	67.65	81.79
	CECM(G)		8.00M	67.80	82.15
	CECM(T)		10.25M	**67.91**	**82.40**

Table 2: The 5-way classification results studying the influence of CECM with ResNet-12. In line with the setting of CAN, cosine distance metric is applied, and Rotation and Fine-tune classifications are not applied. The CECM(M/C/G/T) denote different modes of Patch Cluster such as MatMul, Cosine, GCN and Transformer. Based on ProtoNet, ProtoG adds auxiliary global classification task.

Model	Attention Module	Distance Metric	Param	miniImageNet	
				1-shot	5-shot
ProtoG	-	cosine	7.75M	61.87	78.87
CECNet	-	CECD(M)	7.75M	67.50	82.00
		CECD(C)	7.75M	**67.89**	**82.02**
		CECD(G)	8.00M	67.79	81.74
		CECD(T)	10.25M	67.44	81.17
CECNet	CECM(T)	CECD(M)	10.25M	67.64	81.24
		CECD(C)	10.25M	**68.27**	**82.59**
		CECD(G)	11.25M	66.52	78.55
		CECD(T)	12.75M	64.37	78.32

Table 3: The 5-way classification results studying the influence of CECD with ResNet-12. The setting is consistent with Tab.2, except for distance metric. The CECD(M/C/G/T) denote different modes such as MatMul, Cosine, GCN and Transformer.

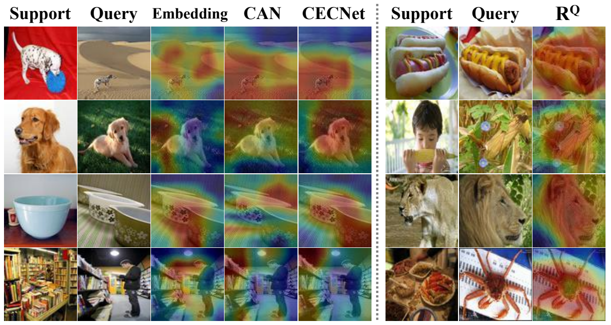
6.2 Ablation Study

Influence of CECM As shown in Tab.2, comparing CECNet to ProtoG, it shows consistent improvements on 1/5-shot classifications, because our CECM enhances the mutually similar regions and produces more discriminative representations. Comparing with CAN adopting cross attention module CAM, our CECNet achieves obvious improvements up to 4.06% on 1-shot task. The results of CECM(M), CECM(C), CECM(G) and CECM(T) show that CECM is not sensitive to alternative modes such as MatMul, Cosine, GCN and Transformer, which indicates the generic Patch Cluster behavior is the key insight for the improvements.

Influence of CECD As shown in Tab.3 without attention module, comparing CECNet to ProtoG, it shows consistent improvements, because our CECD distance metric can obtain a more reliable similarity map. Besides, the results show that the best combination is CECM(T) + CECD(C).

Influence of Multi-Task Loss In Tab.4 with the integration of auxiliary tasks, our CECNet obtains large improvements, which indicates that learning a good embedding is helpful.

Influence of CECM+CECD As shown in Tab.5, comparing to ProtoG (no-attention + cosine), our methods adopting CECM(T) + cosine and no-attention + CECD(C) achieve obvious improvements, which demonstrates the effectiveness of the proposed CECM and CECD. The combination of CECM(T) + CECD(C) obtains further performance gains.



(a) (b)

Figure 4: (a) The class activation maps on 5-way 1-shot classification, where *Embedding* belongs to CECNet. (b) The visualizations of our CEC-based relation map R^Q.

Influence of Self-CECM As illustrated in Tab.6, the baseline is the Metric Classifier of CECNet, and the competitor is Fine-tune Classifier with only Linear layer. By comparing

λ	Loss weights			ResNet-12		WRN-28	
	Metric	Global	Rotation	1-shot	5-shot	1-shot	5-shot
-	0.5	-	-	62.45	79.50	61.98	76.64
-	0.5	-	1.0	65.54	79.55	63.47	77.62
-	0.5	1.0	-	68.27	82.59	67.13	81.95
-	0.5	1.0	1.0	**68.86**	**83.67**	**69.49**	**83.71**
0.5	0.5	w_G	w_R	69.05	83.86	69.33	83.55
1.0	0.5	w_G	w_R	**69.32**	**84.21**	69.66	84.09
1.5	0.5	w_G	w_R	69.15	84.03	69.86	84.30
2.0	0.5	w_G	w_R	69.18	83.29	**70.20**	**84.59**

Table 4: The 5-way classification results on *mini*ImageNet studying the influence of multi-task loss applied in CECNet.

Attention Module	Distance Metric	Param	miniImageNet	
			1-shot	5-shot
-	cosine	7.75M	65.59 ± 0.47	80.94 ± 0.33
CECM(T)	cosine	10.25M	68.27 ± 0.46	83.43 ± 0.32
-	CECD(C)	7.75M	68.79 ± 0.46	83.39 ± 0.32
CECM(T)	CECD(C)	10.25M	**69.32 ± 0.46**	**84.21 ± 0.32**

Table 5: The 5-way results studying the influence of CECM+CECD, under ResNet-12 applying multi-task loss with $\lambda = 1.0$.

Self-CECM+Linear to Linear, it shows consistent improvements, which demonstrates the usefulness of Self-CECM. By comparing Metric+Fine-tune to Metric Classifier, it shows an improvement on 5-shot classification.

6.3 Visualization Analysis

Fig.4(a) shows the class activation maps [Bolei *et al.*, 2016] of our CECNet and CAN [Hou *et al.*, 2019]. Comparing CECNet to its *Embedding*, CECNet can highlight the target object which is unseen in the pre-training stage. Comparing to CAN, CECNet is more accurate and has larger receptive fields. The essential is that our Clustered-patch Element Connection utilizes the global info to implement element connection leading to a more confident correlation and a more clear connection. Fig.4(b) shows the visualizations of the CEC-based relation map R^Q generated by CECNet via Eq.7. Our CEC approach produces a high-quality relation map with a more complete region for the target.

7 Applications on FSSS and FSOD Tasks

In this section, we first introduce a novel CEC-based embedding module named CEC Embedding (CECE). Then, we extend the proposed CECE into few-shot semantic segmentation (FSSS) and object detection (FSOD) tasks. The experimental results in Tab.7 and Tab.8 show that our CECE can achieve performance improvements around $1\% - 3\%$, and more extensive results are presented in APPENDIX.

CEC Embedding f_{CECE} is expressed as:

$$Q' = f_{CECE}(Q) = f_{CEC}(Q, W_E). \tag{16}$$

where, $\{Q, Q'\} \in \mathbb{R}^{hw \times c}$ are the input and output features respectively, and $W_E \in \mathbb{R}^{n_e \times c}$ are learnable weights (pytorch code is $W_E = nn.Embedding(n_e, c)$, and n_e represents the number of semantic groups, and the empirical setting is $n_e = 5$). The proposed CECE can enhance the target regions of input features that are semantically similar to W_E, where W_E contains the semantic information of base categories after trained on the base dataset.

Metric classifier	Fine-tune Classifier		miniImageNet	
	Self-CECM	Linear	1-shot	5-shot
✓	-	-	**70.20 ± 0.46**	84.59 ± 0.30
-	-	✓	69.20 ± 0.47	84.40 ± 0.30
-	✓	✓	69.36 ± 0.46	84.78 ± 0.30
✓	✓	✓	**70.20 ± 0.46**	**85.00 ± 0.30**

Table 6: The 5-way results of CECNet studying the influence of Self-CECM, under WRN-28 applying multi-task loss with $\lambda = 2.0$.

Model	PASCAL-5^i		COCO-20^i	
	1-shot	5-shot	1-shot	5-shot
PPNet [Liu *et al.*, 2020]	51.5	62.0	25.7	36.2
RePRI [Malik *et al.*, 2021]	59.3	64.8	36.6	45.2
RePRI+CECE(M)	60.4	**66.5**	**38.3**	**46.9**
RePRI+CECE(T)	**60.5**	66.2	38.1	46.7

Table 7: Comparison on PASCAL-5^i and COCO-20^i few-shot semantic segmentation benchmarks using mIoU with ResNet-50. The CECE(M/T) denote different modes of MatMul and Transformer.

Model	PASCAL		COCO	
	1-shot	5-shot	1-shot	5-shot
DeFRCN [Qiao *et al.*, 2021]	52.5	60.7	6.5	15.3
MFDC [Wu *et al.*, 2022]	56.1	62.2	10.8	16.4
MFDC+CECE(M)	**59.4**	63.4	**11.5**	**17.2**
MFDC+CECE(T)	58.7	**64.9**	11.2	16.9

Table 8: Comparison on PASCAL Novel Split 3 (nAP50) and COCO (nmAP) few-shot object detection benchmarks with ResNet-101.

CECE Applications As an embedding module, our CECE can be stacked after the backbone network. To verify the effectiveness of the proposed CECE, we insert it into the FSSS method RePRI [Malik *et al.*, 2021] and FSOD method MFDC [Wu *et al.*, 2022], via stacking CECE after their backbones. As illustrated in Tab.7 and Tab.8, our CECE can make consistent improvements upon RePRI and MFDC methods.

8 Conclusion

We propose a novel Clustered-patch Element Connection network (CECNet) for few-shot classification. Firstly, we design a Clustered-patch Element Connection (CEC) layer, which strengthens the target regions of query features by element-wisely connecting them with the clustered-patch features. Then three useful CEC-based modules are derived: CECM and Self-CECM generate more discriminative features, and CECD distance metric obtains a reliable similarity map. Extensive experiments prove that our method is effective, and achieves the state-of-the-arts on few-shot classification benchmark. Furthermore, our CEC approach can be extended into few-shot segmentation and detection tasks, which achieves competitive improvements.

References

[Bolei *et al.*, 2016] Zhou Bolei, Khosla Aditya, Lapedriza Agata, Oliva Aude, and Torralba Antonio. Learning deep features for discriminative localization. In *CVPR*, 2016.

[Bruna *et al.*, 2013] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[Hou *et al.*, 2019] Ruibing Hou, Hong Chang, MA Bingpeng, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *NeurIPS*, 2019.

[Jiangtao *et al.*, 2022] Xie Jiangtao, Long Fei, Lv Jiaming, Wang Qilong, and Li Peihua. Joint distribution matters: Deep brownian distance covariance for few-shot classification. In *CVPR*, 2022.

[Jinxiang and Siqian, 2022] Lai Jinxiang and Yang Siqian. Adaptive multi distance metrics for few-shot classification. In *arXiv*, 2022.

[Kang *et al.*, 2019] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8420–8429, 2019.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[Liu *et al.*, 2020] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *ECCV*, 2020.

[Malik *et al.*, 2021] Boudiaf Malik, Kervadec Hoel, Imtiaz Masud Ziko, Piantanida Pablo, Ben Ayed Ismail, and Dolz Jose. Few-shot segmentation without meta-learning: A good transductive inference is all you need? In *CVPR*, 2021.

[Qiao *et al.*, 2021] Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, and Chi Zhang. Defrcn: Decoupled faster r-cnn for few-shot object detection. In *ICCV*, 2021.

[Rizve *et al.*, 2021] Mamshad Nayeem Rizve, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Exploring complementary strengths of invariant and equivariant representations for few-shot learning. In *CVPR*, 2021.

[Siam *et al.*, 2019] Mennatullah Siam, Boris N Oreshkin, and Martin Jagersand. AMP: Adaptive masked proxies for few-shot segmentation. In *ICCV*, 2019.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

[Tian *et al.*, 2020] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *ECCV*, 2020.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[Vinyals *et al.*, 2016] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016.

[Wang *et al.*, 2020] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*, 2020.

[Wu *et al.*, 2022] Shuang Wu, Wenjie Pei, Dianwen Mei, Fanglin Chen, Jiandong Tian, and Guangming Lu. Multi-faceted distillation of base-novel commonality for few-shot object detection. In *ECCV*, 2022.

[Xu *et al.*, 2021a] Chengming Xu, Yanwei Fu, Chen Liu, Chengjie Wang, Jilin Li, Feiyue Huang, Li Zhang, and Xiangyang Xue. Learning dynamic alignment via meta-filter for few-shot learning. In *CVPR*, 2021.

[Xu *et al.*, 2021b] Luo Xu, Wei Longhui, Wen Liangjian, Yang Jinrong, Xie Lingxi, Xu Zenglin, and Tian Qi. Rectifying the shortcut learning of background for few-shot learning. *NeurIPS*, 2021.

[Yang *et al.*, 2022] Liu Yang, Zhang Weifeng, Xiang Chao, Zheng Tu, Cai Deng, and He Xiaofei. Learning to affiliate: Mutual centralized learning for few-shot classification. In *CVPR*, 2022.

[Zhang *et al.*, 2020a] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, 2020.

[Zhang *et al.*, 2020b] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S Huang. SG-one: Similarity guidance network for one-shot semantic segmentation. *IEEE Transactions on Cybernetics*, 2020.

[Zhengyu *et al.*, 2021] Chen Zhengyu, Ge Jixie, Zhan Heshen, Huang Siteng, and Wang Donglin. Pareto self-supervised training for few-shot learning. In *CVPR*, 2021.