Relational Embedding for Few-Shot Classification

Dahyun Kang Heeseung Kwon Juhong Min Minsu Cho
Pohang University of Science and Technology (POSTECH), South Korea
http://cvlab.postech.ac.kr/research/RENet

Abstract

We propose to address the problem of few-shot classification by meta-learning "what to observe" and "where to attend" in a relational perspective. Our method leverages relational patterns within and between images via self-correlational representation (SCR) and cross-correlational attention (CCA). Within each image, the SCR module transforms a base feature map into a self-correlation tensor and learns to extract structural patterns from the tensor. Between the images, the CCA module computes cross-correlation between two image representations and learns to produce co-attention between them. Our Relational Embedding Network (RENet) combines the two relational modules to learn relational embedding in an end-to-end manner. In experimental evaluation, it achieves consistent improvements over state-of-the-art methods on four widely used few-shot classification benchmarks of miniImageNet, tieredImageNet, CUB-200-2011, and CIFAR-FS.

1. Introduction

Few-shot image classification [11, 27, 53, 72] aims to learn new visual concepts from a small number of examples. The task is defined to classify a given query image into target classes, each of which is unseen during training and represented by only a few support images. Recent methods [1,5,19,34,48,50,63,72,80,84] tackle the problem by meta-learning a deep embedding function such that the distance between images on the embedding space conforms to their semantic distance. The learned embedding function, however, often overfits to irrelevant features [4, 9, 13] and thus fails to transfer to new classes not yet observed in training. While deep neural features provide rich semantic information, it remains challenging to learn a generalizable embedding without being distracted by spurious features.

The central tenet of our approach is that relational patterns, *i.e.*, meta-patterns, may generalize better than individual patterns; an item obtains a meaning in comparison with other items in a system, and thus relevant information can be extracted from the relational structure of items. On this basis, we propose to learn "what to observe" and "where to attend" in a relational perspective and combine them to produce relational embeddings for few-shot learning.

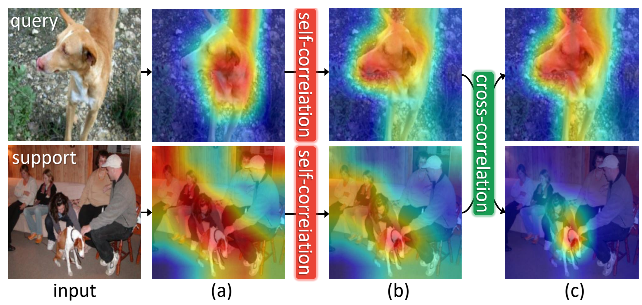

Figure 1: Relational embedding process and its attentional effects. The base image features are transformed via self-correlation to capture structural patterns within each image and then co-attended via cross-correlation to focus on semantically relevant contents between the images. (a), (b), and (c) visualize the activation maps of base features, self-correlational representation, and cross-correlational attention, respectively. See Sec. 5.6 for details.

We achieve this goal by leveraging relational patterns *within* and *between* images via (1) *self-correlational representation* (SCR) and (2) *cross-correlational attention* (CCA). The SCR module transforms a base representation into its self-correlation tensor and learns to extract structural patterns from it. Self-correlation of a deep feature map encodes rich semantic structures by correlating each activation of the feature map to its neighborhood. We perform representation learning on top of it to make relevant structural patterns of the image stand out (Fig. 1 (a)→(b)). On the other hand, the CCA module computes cross-correlation between two image representations and learns to produce co-attention from it. Cross-correlation encodes semantic correspondence relations between the two images. We learn high-dimensional convolutions on the cross-correlation tensor to refine it via convolutional matching and produce adaptive co-attention based on semantic relations between the query and the support (Fig. 1 (b)→(c)).

The proposed method combines the two modules to learn relational embeddings in an end-to-end manner; it extracts relational patterns *within* each image (via SCR), generates relational attention *between* the images (via CCA), and aggregates the cross-attended self-correlation representations to produce the embeddings for few-shot classification. Experiments on four standard benchmark datasets demonstrate that the proposed SCR and CCA modules are effective at highlighting the target object regions and significantly improve few-shot image classification accuracy.

2. Related work

Few-shot classification. Recent few-shot classification methods are roughly categorized into three approaches. The metric-based approach aims to learn an embedding function that maps images to a metric space such that the relevance between a pair of images is distinguished based on their distance [1, 9, 19, 27, 32, 34, 35, 48, 50, 63, 72, 80, 84]. The optimization-based approach meta-learns how to rapidly update models online given a small number of support samples [12, 53, 59, 66, 81]. The two aforementioned lines of work formulate few-shot classification as a meta-learning problem [2,18,60]. The transfer-learning approach [6,8,15, 37, 49, 57, 70, 76, 87] has recently shown that the standard transfer learning procedure [47, 82] of early pre-training and subsequent fine-tuning is a strong baseline for few-shot learning with deep backbone networks. Among these, our work belongs to the metric-based approach. The main idea behind a metric-based few-shot classifier is that real images are distributed on some manifolds of interest, thus the embedding function adequately trained on the training classes can be transferred to embed images of unseen target classes by interpolating or extrapolating the features [61, 69]. Our work improves the transferability of embedding by learning self- and cross-relational patterns that can better generalize to unseen classes.

Self-correlation. Self-correlation or self-similarity reveals a structural layout of an image by measuring similarities of a local patch within its neighborhood [62]. Early work uses the self-correlation itself as a robust descriptor for visual correspondence [71], object detection [7], and action recognition [24,25]. Recent work of [26,30,86] adopts self-correlation as an intermediate feature transform for a deep neural network and shows that it helps the network learn an effective representation for semantic correspondence [26], image translation [86], and video understanding [30]. Inspired by the work, we introduce the SCR module for few-shot classification. Unlike self-correlation used in the previous work, however, our SCR module uses channel-wise self-correlation to preserve rich semantic information for image recognition. Note that while self-attention [52, 75] also computes self-correlation values as attention weights for aggregation, it does not use the self-correlation tensor directly for representation learning and thus is distinct from this line of research.

Cross-correlation. Cross-correlation has long been used as a core component for a wide range of correspondence-related problems in computer vision. It is commonly implemented as a cost-volume or correlation layer in a neural network, which computes matching costs or similarities between two feature maps, and is used for stereo-matching [39,83], optical flow [10,64,79], visual correspondence [33, 41, 43, 44, 56], semantic segmentation [42, 65], video action recognition [29, 74], video object segmentation [21, 46], among others. Some recent few-shot classification methods [9, 19, 34, 84] adopt cross-correlation between a query and each support to identify relevant regions for classification. However, none of them [9, 19, 34, 84] leverage geometric relations of features in cross-correlation and they often suffer from unreliable correlation due to the large variation of appearance. Unlike these previous methods, our CCA module learns to refine the cross-correlation tensor with 4D convolution, filtering out geometrically inconsistent correlations [41, 56], to obtain reliable co-attention. In our experiment, we provide an in-depth comparison with the most related work of [19].

Our contribution can be summarized as follows:

- We propose to learn the self-correlational representation for few-shot classification, which extracts transferable structural patterns within an image.
- We present the cross-correlational attention module for few-shot classification, which learns reliable co-attention between images via convolutional filtering.
- Experiments on four standard benchmarks show our method achieves the state of the art, and ablation studies validate the effectiveness of the components.

3. Preliminary on few-shot classification

Few-shot classification aims to classify images into target classes given only a few images for each class. Deep neural networks are vulnerable to overfitting with such a small amount of annotated samples, and most few-shot classification methods [53, 63, 72] thus adopt a meta-learning framework with episodic training for few-shot adaptation. In few-shot classification, a model is optimized using training data $\mathcal{D}_{\text{train}}$ from classes $\mathcal{C}_{\text{train}}$ and then evaluated on test data $\mathcal{D}_{\text{test}}$ from unseen classes $\mathcal{C}_{\text{test}}$ where $\mathcal{C}_{\text{train}} \cap \mathcal{C}_{\text{test}} = \varnothing$. Both $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ consist of multiple episodes, each of which contains a query set $\mathcal{Q} = (\mathbf{I}_{\text{q}}, y_{\text{q}})$ and a support set $\mathcal{S} = \{(\mathbf{I}_{\text{s}}^{(l)}, y_{\text{s}}^{(l)})\}_{l=1}^{NK}$ of K image-label pairs for each N classes, *i.e.*, N-way K-shot episode [11, 72]. During training, we iteratively sample an episode from $\mathcal{D}_{\text{train}}$ and train the model to learn a mapping from $(\mathcal{S}, \mathbf{I}_{\text{q}})$ to y_{q}. During testing, the model uses the learned mapping to classify $\mathbf{I}_{\text{q}}$ as one of N classes in the support set $\mathcal{S}$ sampled from $\mathcal{D}_{\text{test}}$.

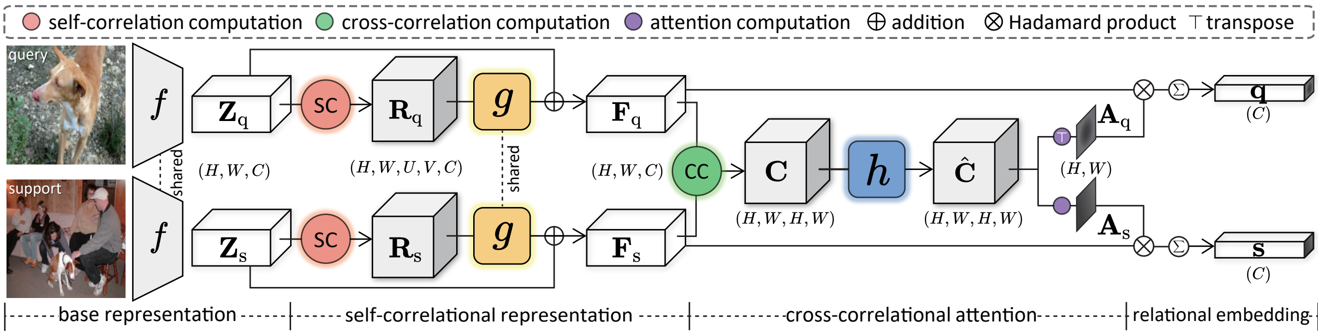

Figure 2: Overall architecture of RENet. The base representations, $\mathbf{Z}_q$ and $\mathbf{Z}_s$, are transformed to self-correlation tensors, $\mathbf{R}_q$ and $\mathbf{R}_s$, which are then updated by the convolutional block g to self-correlational representations, $\mathbf{F}_q$ and $\mathbf{F}_s$, respectively. The cross-correlation $\mathbf{C}$ is computed between the pair of image representations and then refined by the convolutional block h to $\hat{\mathbf{C}}$, which is bidirectionally aggregated to generate co-attention maps, $\mathbf{A}_q$ and $\mathbf{A}_s$. These co-attention maps are applied to corresponding image representations, $\mathbf{F}_q$ and $\mathbf{F}_s$, and their attended features are aggregated to produce the final relational embeddings, $\mathbf{q}$ and $\mathbf{s}$, respectively.

4. Our approach

In this section, we introduce the ***R**elational **E**mbedding **Net**work* (**RENet**) that addresses the challenge of generalization to unseen target classes in a relational perspective. Figure 2 illustrates the overall architecture, consisting of two main learnable modules: self-correlational representation (SCR) module and cross-correlational attention (CCA) module. We first present a brief overview of the proposed architecture in Sec. 4.1. We then present technical details of SCR and CCA in Sec. 4.2 and Sec. 4.3 respectively, and describe our training objective in Sec. 4.4.

4.1. Architecture overview

Given a pair of a query and one of support images, $\mathbf{I}_q$ and $\mathbf{I}_s$, a backbone feature extractor provides base representations, $\mathbf{Z}_q$ and $\mathbf{Z}_s \in \mathbb{R}^{H\times W\times C}$. The SCR module transforms the base representations to self-correlational representations, $\mathbf{F}_q$ and $\mathbf{F}_s \in \mathbb{R}^{H\times W\times C}$, by analyzing feature correlations within an image representation in a convolutional manner. The CCA module then takes the self-correlational representations to generate co-attention maps, $\mathbf{A}_q$ and $\mathbf{A}_s \in \mathbb{R}^{H\times W}$, which give spatial attention weights on aggregating $\mathbf{F}_q$ and $\mathbf{F}_s$ to image embeddings, $\mathbf{q}$ and $\mathbf{s} \in \mathbb{R}^{C}$. This process illustrated in Fig. 2 is applied to all support images $\mathbf{I}_s \in \mathcal{S}$ in parallel, and then the query is classified as the class of its nearest support embedding.

4.2. Self-Correlational Representation (SCR)

The SCR module takes the base representation $\mathbf{Z}$[1] and transforms it to focus more on relevant regions in an image, preparing a reliable input to the CCA module that analyzes feature correlations between a pair of different images. Figure 3a illustrates the architecture of the SCR module.

[1]For notational simplicity, we omit subscripts q and s in this subsection.

Self-correlation computation. Given a base representation $\mathbf{Z} \in \mathbb{R}^{H\times W\times C}$, we compute the Hadamard product of a C-dimensional vector at each position $\mathbf{x} \in [1, H]\times[1, W]$ and those at its neighborhood and collect them into a self-correlation tensor $\mathbf{R} \in \mathbb{R}^{H\times W\times U\times V\times C}$. With an abuse of notation, the tensor $\mathbf{R}$ can be represented as a function with a C-dimensional vector output:

$$\mathbf{R}(\mathbf{x}, \mathbf{p}) = \frac{\mathbf{Z}(\mathbf{x})}{\|\mathbf{Z}(\mathbf{x})\|} \odot \frac{\mathbf{Z}(\mathbf{x}+\mathbf{p})}{\|\mathbf{Z}(\mathbf{x}+\mathbf{p})\|}, \tag{1}$$

where $\mathbf{p} \in [-d_U, d_U]\times[-d_V, d_V]$ corresponds to a relative position in the neighborhood window such that $2d_U + 1 = U$ and $2d_V + 1 = V$, including the center position. Note that the edges of the feature map are zero-padded for sampling off the edges. The similar type of self-correlation, *i.e.*, self-similarity, has been used as a relational descriptor for images and videos that suppresses variations in appearance and reveals structural patterns [62]. Unlike the previous methods [7, 24, 62], which reduce a pair of feature vectors into a scalar correlation value, we use the channel-wise correlation, preserving rich semantics of the feature vectors for classification.

Self-correlational representation learning. To analyze the self-correlation patterns in $\mathbf{R}$, we apply a series of 2D convolutions along $U \times V$ dimensions. As shown in Fig. 3a, the convolutional block follows a bottleneck structure [68] for computational efficiency, which is comprised of a point-wise convolution layer for channel size reduction, two 3×3 convolution layers for transformation, and another point-wise convolution for channel size recovery. Between the convolutions, batch normalization [23] and ReLU [45] are inserted. This convolutional block $g(\cdot)$ gradually aggregates local correlation patterns without padding, thus reducing their spatial dimensions from $U \times V$ to 1×1

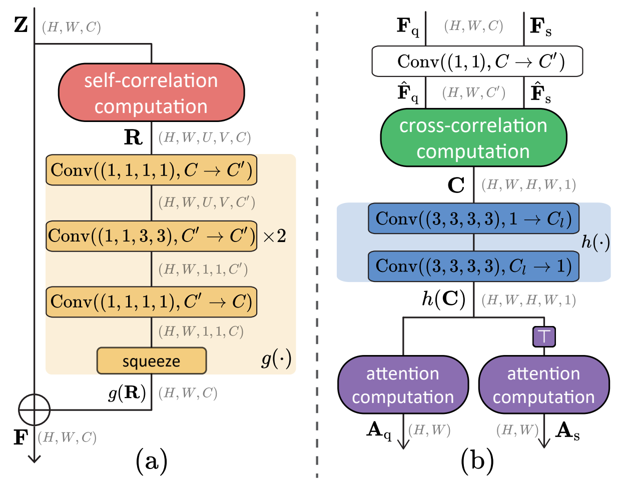

Figure 3: Architecture of SCR and CCA modules. (a): The SCR module captures relational patterns in the input self-correlation $\mathbf{R}$ by convolving it over $U \times V$ dimensions. The result $g(\mathbf{R})$ is added to the base representation $\mathbf{Z}$ to form the self-correlational representation $\mathbf{F}$ (Eq. 2). **(b)**: The CCA module refines the cross-correlation which will be summarized into co-attention maps, $\mathbf{A}_\mathrm{q}$ and $\mathbf{A}_\mathrm{s}$ (Eq. 4).

such that the output $g(\mathbf{R})$ has the same size with $\mathbf{Z}$, *i.e.*, $g : \mathbb{R}^{H \times W \times U \times V \times C} \to \mathbb{R}^{H \times W \times C}$. This process of analyzing structural patterns may be complementary to appearance patterns in the base representation $\mathbf{Z}$. We thus combine the two representations to produce the self-correlational representation $\mathbf{F} \in \mathbb{R}^{H \times W \times C}$:

$$\mathbf{F} = g(\mathbf{R}) + \mathbf{Z}, \tag{2}$$

which reinforces the base features with relational features and helps the few-shot learner better understand "what to observe" within an image. Our experiments show that SCR is robust to intra-class variations and helps generalization to unseen target classes.

4.3. Cross-Correlational Attention (CCA)

The CCA module takes an input pair of query and support SCRs, $\mathbf{F}_\mathrm{q}$ and $\mathbf{F}_\mathrm{s}$, and produces corresponding attention maps, $\mathbf{A}_\mathrm{q}$ and $\mathbf{A}_\mathrm{s}$. These spatial attention maps are used to aggregate each representation to an embedding vector. Figure 3b visualizes the pipeline of the CCA module.

Cross-correlation computation. We first transform both query and support representations, $\mathbf{F}_\mathrm{q}$ and $\mathbf{F}_\mathrm{s} \in \mathbb{R}^{H \times W \times C}$, into more compact representations using a point-wise convolutional layer, reducing its channel dimension C to C'. From the outputs, $\hat{\mathbf{F}}_\mathrm{q}$ and $\hat{\mathbf{F}}_\mathrm{s} \in \mathbb{R}^{H \times W \times C'}$, we construct a 4-dimensional correlation tensor $\mathbf{C} \in \mathbb{R}^{H \times W \times H \times W}$:

$$\mathbf{C}(\mathbf{x}_\mathrm{q}, \mathbf{x}_\mathrm{s}) = \mathrm{sim}(\hat{\mathbf{F}}_\mathrm{q}(\mathbf{x}_\mathrm{q}), \hat{\mathbf{F}}_\mathrm{s}(\mathbf{x}_\mathrm{s})), \tag{3}$$

where $\mathbf{x}$ denotes a spatial position on the feature map and $\mathrm{sim}(\cdot, \cdot)$ means the cosine similarity between two features.

Convolutional matching. The cross-correlation tensor $\mathbf{C}$ may contain unreliable correlations, *i.e.*, matching scores, due to the large appearance variations in the few-shot learning setup. To disambiguate those unreliable matches, we employ the convolutional matching process [41, 56] that refines the tensor by 4D convolutions with matching kernels; 4D convolution on the tensor plays the role of geometric matching by analyzing the consensus of neighboring matches in the 4D space. As shown in Fig. 3b, the convolutional matching block $h(\cdot)$ consists of two 4D convolutional layers; the first convolution produces multiple correlation tensors with multiple matching kernels, increases channel size to C_l, and the second convolution aggregates them to a single 4D correlation tensor, *i.e.*, $\hat{\mathbf{C}} = h(\mathbf{C}) \in \mathbb{R}^{H \times W \times H \times W}$. Batch normalization and ReLU are inserted between the convolutions. We empirically found that two 4D convolutional layers are sufficient for our CCA module.

Co-attention computation. From the refined tensor $\hat{\mathbf{C}}$, we produce co-attention maps, $\mathbf{A}_\mathrm{q}$ and $\mathbf{A}_\mathrm{s}$, which reveal relevant contents between the query and the support. The attention map for the query $\mathbf{A}_\mathrm{q} \in \mathbb{R}^{H \times W}$ is computed by

$$\mathbf{A}_\mathrm{q}(\mathbf{x}_\mathrm{q}) = \frac{1}{HW} \sum_{\mathbf{x}_\mathrm{s}} \frac{\exp\left(\hat{\mathbf{C}}(\mathbf{x}_\mathrm{q}, \mathbf{x}_\mathrm{s})/\gamma\right)}{\sum_{\mathbf{x}'_\mathrm{q}} \exp\left(\hat{\mathbf{C}}(\mathbf{x}'_\mathrm{q}, \mathbf{x}_\mathrm{s})/\gamma\right)}, \tag{4}$$

where $\mathbf{x}$ is a position at the feature map and γ is a temperature factor. Since $\hat{\mathbf{C}}(\mathbf{x}_\mathrm{q}, \mathbf{x}_\mathrm{s})$ is a matching score between the positions $\mathbf{x}_\mathrm{q}$ and $\mathbf{x}_\mathrm{s}$, the attention value $\mathbf{A}_\mathrm{q}(\mathbf{x}_\mathrm{q})$ of Eq. (4) can be interpreted as converting the matching score of $\mathbf{x}_\mathrm{q}$, *i.e.*, a position at the query image, to the average probability of $\mathbf{x}_\mathrm{q}$ being matched to a position at the support image. The attention map for the support $\mathbf{A}_\mathrm{s}$ is similarly computed by switching the query and the support in Eq. (4).

These co-attention maps improve few-shot classification accuracy by meta-learning cross-correlational patterns and adapting "where to attend" with respect to the images given at test time.

4.4. Learning relational embedding

In this subsection, we derive relational embeddings $\mathbf{q}$ and $\mathbf{s} \in \mathbb{R}^C$ from $\mathbf{F}_\mathrm{q}, \mathbf{F}_\mathrm{s}, \mathbf{A}_\mathrm{q}$ and $\mathbf{A}_\mathrm{s}$. We then conclude our method by describing the learning objective.

Attentive pooling. To obtain the final embedding of the query, $\mathbf{q} \in \mathbb{R}^C$, each position of $\mathbf{F}_\mathrm{q} \in \mathbb{R}^{H \times W \times C}$ is multiplied by the spatial attention map $\mathbf{A}_\mathrm{q} \in \mathbb{R}^{H \times W}$ followed by pooling:

$$\mathbf{q} = \sum_{\mathbf{x}_\mathrm{q}} \mathbf{A}_\mathrm{q}(\mathbf{x}_\mathrm{q}) \mathbf{F}_\mathrm{q}(\mathbf{x}_\mathrm{q}). \tag{5}$$

Note that the elements of $\mathbf{A}_\mathrm{q}$ sum up to 1, and thus the attentive embedding $\mathbf{q}$ is a convex combination of $\mathbf{F}_\mathrm{q}$ attended in the context of the support. The final embedding of the support is computed similarly by attending the support feature map $\mathbf{F}_{\text{s}}$ by $\mathbf{A}_{\text{s}}$ followed by pooling:

$$\mathbf{s} = \sum_{\mathbf{x}_{\text{s}}} \mathbf{A}_{\text{s}}(\mathbf{x}_{\text{s}})\mathbf{F}_{\text{s}}(\mathbf{x}_{\text{s}}). \tag{6}$$

On an N-way K-shot classification setting, this co-attentive pooling generates a set of NK different views of a query, $\{\mathbf{q}^{(l)}\}_{l=1}^{NK}$, and a set of support embeddings attended in the context of the query, $\{\mathbf{s}^{(l)}\}_{l=1}^{NK}$.

Learning objective. The proposed RENet is end-to-end trainable from scratch. While most of the recent few-shot classification methods adopt the two-stage training scheme [59, 78, 80, 84] of initial pre-training and subsequent episodic training, we adopt the single-stage training scheme [19, 48] that jointly trains the proposed modules as well as the backbone network by combining two losses: the anchor-based classification loss $\mathcal{L}_{\text{anchor}}$ and the metric-based classification loss $\mathcal{L}_{\text{metric}}$. First, $\mathcal{L}_{\text{anchor}}$ is computed with an additional fully-connected classification layer on top of average-pooled base representation $\mathbf{z}_{\text{q}}$. This loss guides the model to correctly classify a query of class $c \in \mathcal{C}_{\text{train}}$:

$$\mathcal{L}_{\text{anchor}} = -\log \frac{\exp(\mathbf{w}_c^\top \mathbf{z}_{\text{q}} + \mathbf{b}_c)}{\sum_{c'=1}^{|\mathcal{C}_{\text{train}}|} \exp(\mathbf{w}_{c'}^\top \mathbf{z}_{\text{q}} + \mathbf{b}_{c'})}, \tag{7}$$

where $[\mathbf{w}_1^\top, \cdots, \mathbf{w}_{|\mathcal{C}_{\text{train}}|}^\top]$ and $[\mathbf{b}_1, \cdots, \mathbf{b}_{|\mathcal{C}_{\text{train}}|}]$ are weights and biases in the fully-connected layer, respectively. Next, the metric-based loss $\mathcal{L}_{\text{metric}}$ [63, 72] is computed by cosine similarity between a query and support prototype embeddings. Before computing the loss, we average the K query embedding vectors each of which is attended in the context of k^{th} support from n^{th} class to compute $\{\bar{\mathbf{q}}^{(n)}\}_{n=1}^{N}$. Similarly, we average the K support embeddings for each class to obtain a set of prototype embeddings: $\{\bar{\mathbf{s}}^{(n)}\}_{n=1}^{N}$. The metric-based loss guides the model to map a query embedding close to the prototype embedding of the same class:

$$\mathcal{L}_{\text{metric}} = -\log \frac{\exp(\text{sim}(\bar{\mathbf{s}}^{(n)}, \bar{\mathbf{q}}^{(n)})/\tau)}{\sum_{n'=1}^{N} \exp(\text{sim}(\bar{\mathbf{s}}^{(n')}, \bar{\mathbf{q}}^{(n')})/\tau)}, \tag{8}$$

where $\text{sim}(\cdot, \cdot)$ is cosine similarity and τ is a scalar temperature factor. At inference, the class of the query is predicted as that of the nearest prototype.

The objective combines the two losses:

$$\mathcal{L} = \mathcal{L}_{\text{anchor}} + \lambda \mathcal{L}_{\text{metric}}, \tag{9}$$

where λ is a hyper-parameter that balances the loss terms. Note that the fully-connected layer involved in computing $\mathcal{L}_{\text{anchor}}$ is discarded during inference.

5. Experimental results

In this section, we evaluate RENet on standard benchmarks and compare the results with the recent state of the arts. We also conduct ablation studies to validate the effect of the major components. For additional results and analyses, we refer the readers to our supplementary material.

5.1. Datasets

For evaluation, we use four standard benchmarks for few-shot classification: *mini*ImageNet, *tiered*ImageNet, CUB-200-2011, and CIFAR-FS. ***mini*ImageNet** [72] is a subset of ImageNet [58] consisting of 60,000 images uniformly distributed over 100 object classes. The train/validation/test splits consist of 64/16/20 object classes, respectively. ***tiered*ImageNet** [55] is a challenging dataset in which train/validation/test splits are disjoint in terms of super-classes from the ImageNet hierarchy, which typically demands better generalization than other datasets. The respective train/validation/test splits consist of 20/6/8 super-classes, which are super-sets of 351/97/160 sub-classes. **CUB-200-2011** (CUB) [73] is a dataset for fine-grained classification of bird species, consisting of 100/50/50 object classes for train/validation/test splits, respectively. Following the recent work of [80, 84], we use pre-cropped images to human-annotated bounding boxes. **CIFAR-FS** [3] is built upon CIFAR-100 [28] dataset. Following the recent work of [3], we use the same train/validation/test splits consisting of 64/16/20 object classes, respectively. For all the datasets, $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, and $\mathcal{D}_{\text{test}}$ are disjoint in terms of object classes such that $\mathcal{C}_{\text{train}} \cap \mathcal{C}_{\text{val}} = \mathcal{C}_{\text{val}} \cap \mathcal{C}_{\text{test}} = \mathcal{C}_{\text{test}} \cap \mathcal{C}_{\text{train}} = \varnothing$.

5.2. Implementation details

We adopt ResNet12 [17] following the recent few-shot classification work [48, 54, 80, 84]. The backbone network takes an image with spatial size of 84×84 as an input and provides a base representation $\mathbf{Z} \in \mathbb{R}^{5 \times 5 \times 640}$ followed by shifting its channel activations by the channel mean of an episode [84]. For our CCA module, we adopt separable 4D convolutions [79] with kernel size of $3 \times 3 \times 3 \times 3$ for its effectiveness in approximating the original 4D convolutions [56] as well as efficiency in terms of both memory and time. The output of the 4D convolution $\hat{\mathbf{C}}$ is normalized such that the entities in the pair of spatial map to be zero-mean and unit-variance to stabilize training. We set $C' = 64$ in SCR and $C_l = 16$ in CCA module. For the N-way K-shot evaluation, we test 15 query samples for each class in an episode and report average classification accuracy with 95% confidence intervals of randomly sampled 2,000 test episodes. The hyperparameter λ is set to 0.25, 0.5, 1.5 for ImageNet derivatives, CIFAR-FS, CUB, respectively. γ is set to 2 for CUB and 5 otherwise. We set $\tau = 0.2, U, V = 5$ in our experiments.

method	backbone	5-way 1-shot	5-way 5-shot
cosine classifier [6]	*ResNet12*	55.43 ± 0.81	77.18 ± 0.61
TADAM [48]	*ResNet12*	58.50 ± 0.30	76.70 ± 0.30
Shot-Free [54]	*ResNet12*	59.04	77.64
TPN [38]	*ResNet12*	59.46	75.65
PPA [51]	*WRN-28-10*†	59.60 ± 0.41	73.74 ± 0.19
wDAE-GNN [16]	*WRN-28-10*†	61.07 ± 0.15	76.75 ± 0.11
MTL [66]	*ResNet12*	61.20 ± 1.80	75.50 ± 0.80
LEO [59]	*WRN-28-10*†	61.76 ± 0.08	77.59 ± 0.12
RFS-simple [70]	*ResNet12*	62.02 ± 0.63	79.64 ± 0.44
DC [36]	*ResNet18*	62.53 ± 0.19	79.77 ± 0.19
ProtoNet [63]	*ResNet12*	62.39 ± 0.21	80.53 ± 0.14
MetaOptNet [31]	*ResNet12*	62.64 ± 0.82	78.63 ± 0.46
SimpleShot [77]	*ResNet18*	62.85 ± 0.20	80.02 ± 0.14
MatchNet [72]	*ResNet12*	63.08 ± 0.80	75.99 ± 0.60
S2M2 [40]	*ResNet34*†	63.74 ± 0.18	79.45 ± 0.12
CAN [19]	*ResNet12*	63.85 ± 0.48	79.44 ± 0.34
NegMargin [37]	*ResNet12*	63.85 ± 0.81	81.57 ± 0.56
CTM [32]	*ResNet18*	64.12 ± 0.82	80.51 ± 0.13
DeepEMD [84]	*ResNet12*	65.91 ± 0.82	82.41 ± 0.56
FEAT [80]	*ResNet12*	66.78 ± 0.20	82.05 ± 0.14
RENet (ours)	*ResNet12*	**67.60 ± 0.44**	**82.58 ± 0.30**

(a) Results on *mini*ImageNet dataset.

method	backbone	5-way 1-shot	5-way 5-shot
TPN [38]	*ResNet12*	59.91 ± 0.94	73.30 ± 0.75
cosine classifier [6]	*ResNet12*	61.49 ± 0.91	82.37 ± 0.67
Shot-Free [54]	*ResNet12*	63.52	82.59
PPA [51]	*WRN-28-10*†	65.65 ± 0.92	83.40 ± 0.65
MetaOptNet [31]	*ResNet12*	65.99 ± 0.72	81.56 ± 0.53
LEO [59]	*WRN-28-10*†	66.33 ± 0.05	81.44 ± 0.09
wDAE-GNN [16]	*WRN-28-10*†	68.18 ± 0.16	83.09 ± 0.12
ProtoNet [63]	*ResNet12*	68.23 ± 0.23	84.03 ± 0.16
CTM [32]	*ResNet18*	68.41 ± 0.39	84.28 ± 1.73
MatchNet [72]	*ResNet12*	68.50 ± 0.92	80.60 ± 0.71
RFS-simple [70]	*ResNet12*	69.74 ± 0.72	84.41 ± 0.55
CAN [19]	*ResNet12*	69.89 ± 0.51	84.23 ± 0.37
FEAT [80]	*ResNet12*	70.80 ± 0.23	84.79 ± 0.16
DeepEMD [84]	*ResNet12*	71.16 ± 0.87	**86.03 ± 0.58**
RENet (ours)	*ResNet12*	**71.61 ± 0.51**	85.28 ± 0.35

(b) Results on *tiered*ImageNet dataset.

Table 1: Performance comparison in terms of accuracy (%) with 95% confidence intervals on **(a)** *mini*ImageNet and **(b)** *tiered*ImageNet. "†" denotes larger backbones than *ResNet12*.

method	backbone	5-way 1-shot	5-way 5-shot
ProtoNet [63]	*ResNet12*	66.09 ± 0.92	82.50 ± 0.58
RelationNet [67]	*ResNet34*†	66.20 ± 0.99	82.30 ± 0.58
MAML [67]	*ResNet34*†	67.28 ± 1.08	83.47 ± 0.59
cosine classifier [6]	*ResNet12*	67.30 ± 0.86	84.75 ± 0.60
MatchNet [72]	*ResNet12*	71.87 ± 0.85	85.08 ± 0.57
NegMargin [37]	*ResNet18*	72.66 ± 0.85	89.40 ± 0.43
S2M2 [40]	*ResNet34*†	72.92 ± 0.83	86.55 ± 0.51
FEAT* [80]	*ResNet12*	73.27 ± 0.22	85.77 ± 0.14
DeepEMD [84]	*ResNet12*	75.65 ± 0.83	88.69 ± 0.50
RENet (ours)	*ResNet12*	**79.49 ± 0.44**	**91.11 ± 0.24**

(a) Results on CUB-200-2011 dataset.

method	backbone	5-way 1-shot	5-way 5-shot
cosine classifier [6]	*ResNet34*†	60.39 ± 0.28	72.85 ± 0.65
S2M2 [40]	*ResNet34*†	62.77 ± 0.23	75.75 ± 0.13
Shot-Free [54]	*ResNet12*	69.2	84.7
RFS-simple [70]	*ResNet12*	71.5 ± 0.8	86.0 ± 0.5
ProtoNet [63]	*ResNet12*	72.2 ± 0.7	83.5 ± 0.5
MetaOptNet [31]	*ResNet12*	72.6 ± 0.7	84.3 ± 0.5
Boosting [14]	*WRN-28-10*†	73.6 ± 0.3	86.0 ± 0.2
RENet (ours)	*ResNet12*	**74.51 ± 0.46**	**86.60 ± 0.32**

(b) Results on CIFAR-FS dataset.

Table 2: Performance comparison in terms of accuracy (%) with 95% confidence intervals on **(a)** CUB-200-2011 and **(b)** CIFAR-FS. "†" denotes larger backbones than *ResNet12*, and "*" denotes reproduced one.

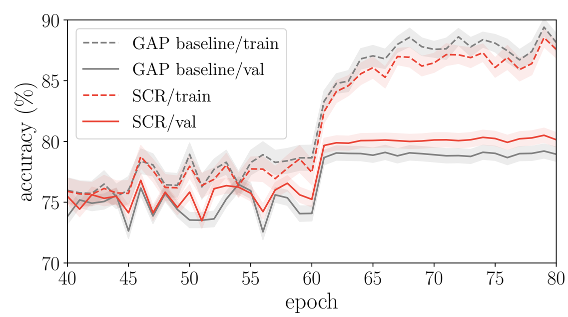

Figure 4: Learning curves of the GAP baseline and SCR in terms of accuracy (%) with 95 % confidence intervals on CUB-200-2011. The curves for the first 40 epochs are omitted.

5.3. Comparison to the state-of-the-art methods

Tables 1 and 2 compare RENet and current few-shot classification methods on four datasets. Our model uses a smaller backbone than that of several methods [16, 40, 51, 59] yet sets a new state of the art in both 5-way 1-shot and 5-shot settings on *mini*ImageNet, CUB-200-2011, and CIFAR-FS datasets while being comparable to DeepEMD [84] on *tiered*ImageNet. Note that DeepEMD iteratively performs back-propagation steps at each inference, which is very slow; it takes 8 hours to evaluate 2,000 5-way 5-shot episodes while ours takes 1.5 minutes on the same machine with an Intel i7-7820X CPU and an NVIDIA TitanXp GPU. We also find RENet outperforms transfer learning methods [6, 37, 70, 77] that are not explicitly designed to learn cross-relation between a query and supports. However, RENet benefits from explicitly meta-learning cross-image relations and is able to better recognize image relevance adaptively to given few-shot examples.

SCR	CCA	*mini*-ImageNet	CUB
✗	✗	65.33	77.54
✓	✗	66.66 (+1.33)	78.69 (+1.15)
✗	✓	65.90 (+0.57)	78.49 (+0.95)
✓	✓	**67.60** (+2.27)	**79.49** (+1.95)

Table 3: Effects of SCR and CCA.

id	CCA	$h(\cdot)$ channel sizes $C_{\text{in}} \to C_l \to C_{\text{out}}$	*mini*-ImgNet	CUB
(a)	✗	✗	65.33	77.54
(b)	✓	✗	65.73	77.75
(c)	✓	$1 \to 1 \to 1$	65.75	78.05
(d)	✓	$64 \to 16 \to 1$	66.18	78.10
(e)	✓	$1 \to 16 \to 1$	**65.90**	**78.49**

Table 4: Effects of CCA variants.

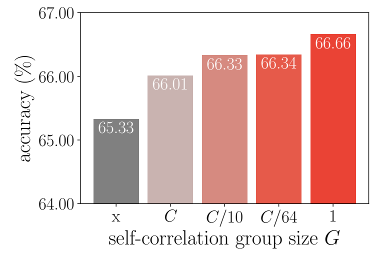

Figure 5: Effects of the group size in SCR on *mini*ImageNet.

5.4. Ablation studies

To investigate the effects of core modules in RENet, we conduct extensive ablation studies either in the absence of each module or by replacing them with others and compare the results in the 5-way 1-shot setting. For ease of comparison, we use a baseline model called GAP baseline that applies global-average pooling to base representations to obtain final embeddings.

Effects of the proposed modules. Table 3 summarizes the effects of the SCR and CCA modules. Without SCR, the model skips self-correlational learning, replacing its output $\mathbf{F}$ with the base representation $\mathbf{Z}$. Without CCA, the model skips computing cross-correlation and obtains final image embeddings by simply averaging either $\mathbf{Z}$ or $\mathbf{F}$. Both modules consistently improve classification accuracies on both datasets. From the results, we observe that the effectiveness of CCA is more solid on CUB than that on *mini*ImageNet. As the CCA module provides co-attention from the geometric consensus in cross-correlation patterns, it is particularly beneficial for a task where objects across different classes exhibit small geometric variations. We also experimentally show that the self-correlational representation generalizes well to unseen classes than the base representation does as seen in Fig. 4; the SCR achieves lower training accuracy but higher validation accuracy than the GAP baseline.

Design choices of SCR. To see the effectiveness of channel-wise correlation in SCR, we replace the Hadamard product in Eq. (1) with group-wise cosine similarity in computing a self-correlation $\mathbf{R} \in \mathbb{R}^{H\times W\times U\times V\times C/G}$ and interpolate the group size G. Namely, a group size $G > 1$ compresses the channels of self-correlation, and $G = 1$ becomes equivalent to the proposed method. Figure 5 shows that the self-correlation with $G = C$, which represents the feature relation as a similarity scalar, is already effective, and further, the performance gradually increases as smaller group sizes are used; the model benefits from relational information, and the effect becomes greater with richer relation in the channel-wise correlation as similarly observed in [85].

Design choices of CCA. We vary the components in the CCA module and denote the variants from (b) to (d) in Table 4 to verify our design choice. In this study, we exclude SCR learning to focus on the impact of the CCA. We first examine a non-parametric baseline (b) by ablating all learnable parameters in the CCA module, *i.e.*, we replace $\hat{\mathbf{C}}$ in Eq. (4) with the cross-correlation between $\mathbf{Z}_{\text{q}}$ and $\mathbf{Z}_{\text{s}}$. It shows marginal improvement from the GAP baseline (a), which implies that the naïve cross-correlation hardly gives reliable co-attention maps. Another variant (c) validates that the hidden channel dimension C_l (Fig. 3b) helps the model capture diverse cross-correlation patterns. The last variant (d) constructs cross-correlation preserving the channel dimension using Hadamard product instead of cosine similarity in Eq. (3). Although it provides much information to the module and requires more learnable parameters ((d): 797.3K *vs.* (e): 45.8K), it is not very effective than the proposed one (e) possibly because too abundant correlations between two independent images negatively affect model generalization.

method	self	cross	*mini*-ImgNet	CUB	# add. params.
GAP baseline	✗	✗	65.33	77.54	0K
SE [20]	✓(self-attn)	✗	63.34	78.40	83.8K
non-local [75]	✓(self-attn)	✗	65.00	77.11	822.1K
local [52]	✓(self-attn)	✗	66.26	78.19	1644.1K
SCE [22]	✓(self-sim)	✗	63.39	78.43	89.2K
CAN* [19]	✗	✓	65.66	77.77	0.3K
SCR	✓(self-corr)	✗	66.66	78.69	157.3K
CCA	✗	✓	65.90	78.49	45.8K
SCR + CCA	✓(self-corr)	✓	**67.60**	**79.49**	203.2K

Table 5: Accuracy (%) and the number of additional learnable parameters of different attention modules. "*" denotes a reproduced result under a controlled environment for a fair comparison, and the underline denotes the best performance among others.

5.5. Comparison with other attention modules

In Table 5, we compare the proposed modules with other attention modules by replacing ours with others. We first compare self-attention methods [20, 52, 75] that attend to *appearance* features based on feature similarity, while our SCR module extracts *relational* features based on local self-correlation. In the comparison, SCR outperforms self-attention methods, suggesting the effectiveness of learning self-correlation patterns for few-shot learning. We find that

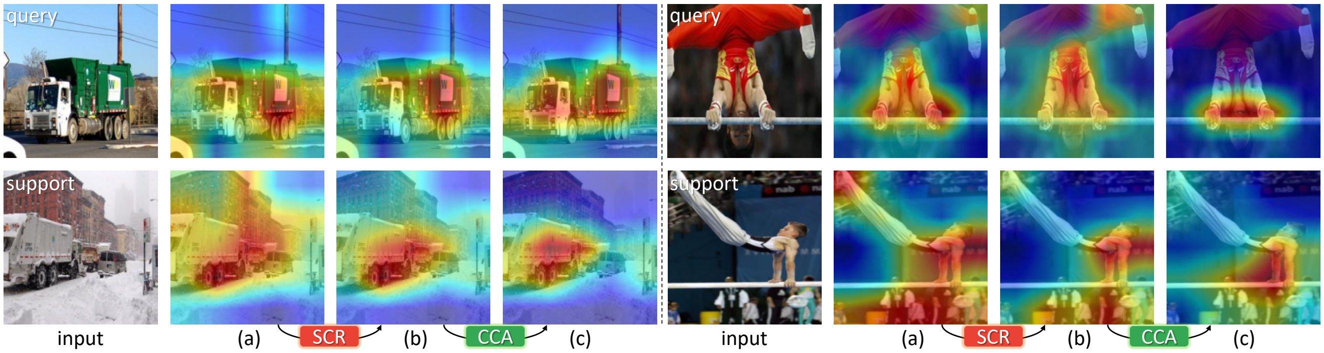

Figure 6: Effects of RENet. **(a)**: Channel activation of base representation. **(b)**: Channel activation of SCR. **(c)**: Attention map of CCA.

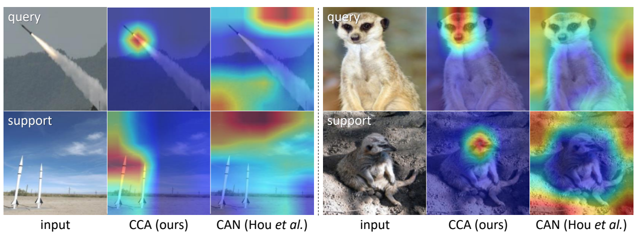

Figure 7: Co-attention comparison with CAN [19]. Our CCA better attends to common objects against confusing backgrounds.

learning such relational patterns of "how each feature correlates with its neighbors" is transferable to unseen classes and compensates the lack of data issue of few-shot learning. While SCR outperforms most methods, it closely competes with SCE [22] on CUB. SCE computes cosine similarity between a reference position and its neighbors and concatenates their similarities at the channel dimension in a fixed order. SCE is powerful on the CUB dataset (77.54% $\rightarrow$ 78.43%) that has relatively little pose variation across images, however, it is disadvantageous on the *mini*ImageNet dataset (65.33% $\rightarrow$ 63.39%). This is because SCE imprints the positional order in channel indices, which limits observing diverse neighborhood relations, whereas SCR uses multiple channels to capture various relations with neighbors.

In Table 5, we also observe that CCA performs better than CAN [19] as well as other self-attention methods with a reasonable amount of additional parameters. CAN first averages a 4D cross-correlation to a 2D correlation tensor and feeds it to multi-layer perceptrons that produce an attention mask, which is repeated similarly by switching the query and the support to generate co-attention maps. We empirically find that the process of averaging the initial 4D correlation collapses fine and crucial match details between images. Figure 7 shows two examples that CAN is overwhelmed by dominant backgrounds and hardly attends to small objects. Whereas, the CCA module updates cross-correlations while retaining the spatial dimensions hence successfully attends to relevant objects.

Combining the SCR and the CCA modules, our model outperforms all the other methods.

5.6. Qualitative results

The relational embedding process and its attentional effects are shown in Figs. 1 and 6. The columns (a) and (b) visualize the averaged channel activation of the base representation $\mathbf{Z}$ and the self-correlational representation $\mathbf{F}$, respectively. The column (c) visualizes the 2D attention map $\mathbf{A}$. The images are randomly sampled from the *mini*ImageNet validation set, and activations are bi-linearly interpolated to the input image size. The results demonstrate that the SCR module can deactivate irrelevant features via learning self-correlation with neighborhood, *e.g.*, the activation of a building behind a truck decreases. The subsequent CCA module generates co-attention maps that focus on the common context between a query and a support, *e.g.*, the hands grasping the bars are co-attended.

6. Conclusion

In this work, we have proposed the relational embedding network for few-shot classification, which leverages the self-correlational representation and the cross-correlational attention. Combining the two modules, our method has achieved the state of the art on the four standard benchmarks. One of our experimental observations is that self-attention mechanism [52, 75] is prone to overfitting to the training set so that it does not generalize to unseen classes in the few-shot learning context. Our work, however, has shown that learning structural correlations between visual features better generalizes to unseen object classes and brings performance improvement to few-shot image recognition, suggesting a promising direction of relational knowledge as a transferable prior.

Acknowledgements. This work was supported by Samsung Electronics Co., Ltd. (IO201208-07822-01) and the IITP grants (No.2019-0-01906, AI Graduate School Program - POSTECH) (No.2021-0-00537, Visual common sense through self-supervised learning for restoration of invisible parts in images) funded by Ministry of Science and ICT, Korea.

References

[1] Kelsey Allen, Evan Shelhamer, Hanul Shin, and Joshua Tenenbaum. Infinite mixture prototypes for few-shot learning. In *Proc. International Conference on Machine Learning (ICML)*, 2019. 1, 2

[2] Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. *Learning a synaptic learning rule*. Citeseer, 1990. 2

[3] Luca Bertinetto, Joao F Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *Proc. International Conference on Learning Representations (ICLR)*, 2018. 5

[4] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *Proc. International Conference on Learning Representations (ICLR)*, 2019. 1

[5] Kaidi Cao, Maria Brbic, and Jure Leskovec. Concept learners for few-shot learning. In *Proc. International Conference on Learning Representations (ICLR)*, 2021. 1

[6] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations (ICLR)*, 2019. 2, 6

[7] Thomas Deselaers and Vittorio Ferrari. Global and efficient self-similarity for object classification and detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010. 2, 3

[8] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *International Conference on Learning Representations*, 2019. 2

[9] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2

[10] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015. 2

[11] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(4):594–611, 2006. 1, 2

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. International Conference on Machine Learning (ICML)*, 2017. 2

[13] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *Proc. International Conference on Learning Representations (ICLR)*, 2019. 1

[14] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 6

[15] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[16] Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[18] Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *Proc. International Conference on Artificial Neural Networks (ICANN)*, 2001. 2

[19] Ruibing Hou, Hong Chang, MA Bingpeng, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2, 5, 6, 7, 8

[20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7

[21] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 2

[22] Shuaiyi Huang, Qiuyue Wang, Songyang Zhang, Shipeng Yan, and Xuming He. Dynamic context correspondence network for semantic alignment. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 7, 8

[23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. International Conference on Machine Learning (ICML)*, 2015. 3

[24] Imran N Junejo, Emilie Dexter, Ivan Laptev, and Patrick Pérez. Cross-view action recognition from temporal self-similarities. In *Proc. European Conference on Computer Vision (ECCV)*, 2008. 2, 3

[25] Imran N Junejo, Emilie Dexter, Ivan Laptev, and Patrick Pérez. View-independent action recognition from temporal self-similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(1):172–185, 2011. 2

[26] Seungryong Kim, Dongbo Min, Bumsub Ham, Sangryul Jeon, Stephen Lin, and Kwanghoon Sohn. Fcss: Fully convolutional self-similarity for dense semantic correspondence. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[27] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *International Conference on Machine Learning (ICML) Workshop on Deep Learning*, 2015. 1, 2

[28] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, 2009. 5

[29] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2

[30] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Learning self-similarity in space and time as generalized motion for video action recognition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[31] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[32] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6

[33] Shuda Li, Kai Han, Theo W Costain, Henry Howard-Jenkins, and Victor Prisacariu. Correspondence networks with adaptive neighbourhood consensus. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[34] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[35] Yann Lifchitz, Yannis Avrithis, and Sylvaine Picard. Local propagation for few-shot learning. In *International Conference on Pattern Recognition (ICPR)*, 2021. 2

[36] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. Dense classification and implanting for few-shot learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[37] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2, 6

[38] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *Proc. International Conference on Learning Representations (ICLR)*, 2018. 6

[39] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[40] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020. 6

[41] Juhong Min and Minsu Cho. Convolutional hough matching networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 4

[42] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[43] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Hyperpixel flow: Semantic correspondence with multi-layer neural features. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[44] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Learning to compose hypercolumns for visual correspondence. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2

[45] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*, 2010. 3

[46] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019. 2

[47] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[48] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 1, 2, 5, 6

[49] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[50] Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[51] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6

[52] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2, 7, 8

[53] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *Proc. International Conference on Learning Representations (ICLR)*, 2017. 1, 2

[54] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 5, 6

[55] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *Proc. International Conference on Learning Representations (ICLR)*, 2018. 5

[56] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consen-

sus networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2, 4, 5

[57] Pau Rodríguez, Issam Laradji, Alexandre Drouin, and Alexandre Lacoste. Embedding propagation: Smoother manifold for few-shot classification. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2

[58] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[59] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2018. 2, 5, 6

[60] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987. 2

[61] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[62] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 2, 3

[63] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1, 2, 5, 6

[64] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[65] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2

[66] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6

[67] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6

[68] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[69] Joshua B Tenenbaum. Mapping a manifold of perceptual observations. *Advances in Neural Information Processing Systems (NeurIPS)*, 1998. 2

[70] 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 *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2, 6

[71] Atousa Torabi and Guillaume-Alexandre Bilodeau. Local self-similarity-based registration of human rois in pairs of stereo thermal-visible videos. *Pattern Recognition*, 46(2):578–589, 2013. 2

[72] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 1, 2, 5, 6

[73] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, 2011. 5

[74] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[75] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 7, 8

[76] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *Proc. International Conference on Machine Learning (ICML)*, 2020. 2

[77] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019. 6

[78] Davis Wertheimer, Luming Tang, and Bharath Hariharan. Few-shot classification with feature map reconstruction networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 5

[79] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2, 5

[80] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 5, 6

[81] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2

[82] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. 2

[83] Jure Žbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research (JMLR)*, 17(1):2287–2318, 2016. 2

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

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 1, 2, 5, 6

[85] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7

[86] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. The spatially-correlative loss for various image translation tasks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[87] Imtiaz Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. Laplacian regularized few-shot learning. In *Proc. International Conference on Machine Learning (ICML)*, 2020. 2