FEW-SHOT LEARNING BY EXPLOITING OBJECT RELATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-shot learning trains image classifiers over datasets with few examples per category. It poses challenges for the optimization algorithms, which typically require many examples to fine-tune the model parameters for new categories. Metric-learning-based approaches avoid the optimization issue by embedding the images into a metric space and applying the nearest neighbour classifier for new categories. In this paper, we propose to exploit the object-level relation to learn the image relation feature, which is converted into a distance directly. For a new category, even though its images are not seen by the model, some objects may appear in the training images. Hence, object-level relation is useful for inferring the relation of images from unseen categories. Consequently, our model generalizes well for new categories without fine-tuning. Experimental results on benchmark datasets show that our approach outperforms state-of-the-art methods.

1 INTRODUCTION

Real-world data typically follows power-law distributions, where the majority of the data categories have only a small number of examples. For instance, to train an image classifier for food images, one would probably crawl few images for some local dishes. Similarly, there are few images for new products, e.g. new toys. However, state-of-the-art image classifiers, i.e. deep convolutional neural networks (ConvNets) Krizhevsky et al. (2012), are hungry for data. The benchmark datasets for ConvNets, including CIFAR10 and ImageNet Deng et al. (2009), usually have more than 1000 images per category. Fine-tuning ConvNets Yosinski et al. (2014) by transferring the knowledge (i.e. parameters) learned from a big dataset could alleviate the gap, but still fails to resolve the issue. This is because the widely used gradient-based optimization algorithms need many iterations over plenty of examples to adapt the ConvNets (with a large number of parameters) for new categories.

Two types of approaches have been proposed towards addressing the above issue. They are referred as *few-shot image classification*, which trains classifiers over datasets with few (e.g. less than 20) examples per category. The first set of approaches Ravi & Larochelle (2016); Li et al. (2017); Finn et al. (2017) are based on meta-learning. They train a meta learner to guide the optimization of the classifier for the new categories. They improve the optimization by providing a good initialization Finn et al. (2017), an adaptive learning rate Li et al. (2017) or even replacing the gradient-based optimization method Ravi & Larochelle (2016). The second set of approaches Koch et al. (2015); Vinyals et al. (2016); Santoro et al. (2016); Snell et al. (2017); Sung et al. (2017) are based on embedding learning. They learn an embedding function to project the images into a space and then classify images from new categories through the nearest neighbour search. No fine-tuning is required as the nearest neighbour classifier is non-parametric. The embedding functions are vital to the classification accuracy, which must be general enough to extract good embedding features for evaluating the distance/similarity between images belonging to the unseen categories.

In this paper, we propose a new few-short learning approach. It is motivated by the observation that human beings are pretty good at few-shot learning. Take the Segway in Figure 1 as an example (Koch (2015)) although the Segway could be new to us, we are familiar with its components, e.g. wheels, which are similar to those of the motors or electric scooters. Hence, we know Segway is a traffic tool for riding. Moreover, we are able to analyze the image by decomposing it. For example, we are aware of the relationship between Segway and rider. This kind of relationship-awareness helps in

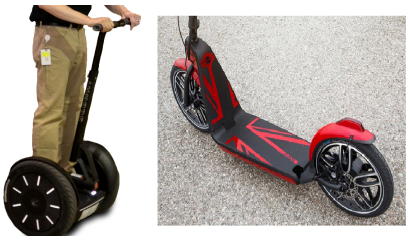


Figure 1: Example images with a Segway and an electric scooter.

recognition when we see a different rider with another Segway. However, existing methods take each image as a whole without exploiting the object-level information including relation.

Based on the above observation, we design our learning model with two parts, namely the *relation extraction network* and the *distance learning network*. We draw inspiration from the relation network Santoro et al. (2017). In particular, we compare the objects from two images instead of a single image as Santoro et al. (2017) in order to extract the relation between images. The relation is converted into a similarity score. We expect the object relationship to play a crucial role in determining the image relation for distinguishing images from different categories. The training is conducted in episodes, each of which is constructed in the same way as in the test, i.e. with few examples for each category. After training, we extract the relation feature vectors among the query image (to be classified) and each labelled image from the test dataset. Nearest neighbour classifier is then applied with the similarity score calculated from the relation feature vector. Extensive experiments on benchmark datasets confirm the superiority of our approach in terms of classification accuracy against existing work.

2 Related Work

In this section, we review the related work of few-shot image classification including meta learning and embedding learning based approaches. After that, we briefly introduce the relation network, which is exploited in our method to learn the relation of images.

2.1 Few-Shot Image Classification

Few-shot image classification is the task of training image classifiers over datasets with few examples per category. It is useful for recognizing new categories, e.g. products. With the resurgence of deep learning, most few-shot image classifiers are based on ConvNets. A simple solution is to fine-tune the ConvNets trained on a similar dataset with many examples per category. However, the widely used gradient based optimization algorithms (e.g. mini-batch Stochastic Gradient Descent, SGD) need a lot of examples to adapt (fine-tune) the ConvNets Ravi & Larochelle (2016) for the new categories.

Meta Learning Towards the optimization challenge, meta learning trains a meta learner that guides the optimization algorithms to fine-tune the learner (i.e. classifier). It is also called learning to optimize. The meta learner is trained iteratively. For each iteration, an episode is sampled from the training dataset, which has the same setting as the test scenario. In other words, an episode has the same number of categories and the same number of examples per category as the test. The meta learner is trained to fine-tune the classifier for a large number of episodes. After training, the meta learner is expected to improve the optimization (fine-tune) of the classifier for the test episodes. The meta learner of MAML Finn et al. (2017) learns good parameter initialization such that the fine-tuning can adapt the parameters quickly and effectively for the test episodes. Meta-SGD's meta learner Li et al. (2017) generates both the initialization and learning rate for the fine-tuning optimization algorithm. A more aggressive approach Ravi & Larochelle (2016) is to learn a LSTM to generate the updates for fine-tuning. It replaces the SGD with LSTM for fine-tuning.

Embedding Learning Another set of approaches eliminate the fine-tuning step to avoid the optimization problem. They learn a general embedding function to project both training and testing images into an embedding space, where nearest neighbor search could be used as the classifier. Koch et al. (2015) adapt Siamese network to do the feature embedding by training the network to predict

the relation (from the same category or different categories) of two training images. LSTM Vinyals et al. (2016) and memory-augmented networks Santoro et al. (2016) are also applied to learn the embedding space. Prototypical Network Snell et al. (2017) learns an embedding for each category by averaging the features of all samples in the category. LearningToCompare Sung et al. (2017) is the most relevant work. It compares images within one episode to learn their relation scores, which serve as the metric distance. However, it ignores the rich object-level information when doing the comparison between images. Our approach exploits the object relation to learn image relation.

2.2 Relation Network

Relation network Santoro et al. (2017) is introduced for modeling the object relation within one image. Good performance has been achieved for visual question answering. Every pair of objects is compared by concatenating their features. A final relation feature that summarizes all object relations is aggregated after some transformation. In this paper, we adapt the relation network extract the relation of objects from different images in order to measure the relation of two images. Although the test images are new to the classifier for few-shot classification, their object relations could persist in the training dataset. Consequently, the object-level relation helps to decide the image relation. In addition, we feed the relation features into a metric learning network to differentiate similar and dissimilar images in each training episode. As a result, the extracted relation features are discriminative for the test episode.

3 Methodology

3.1 Problem Definition

For few-shot image classification, we are given a support set of labeled images $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in R^D$ is the feature of an image, and $y_i \in \mathcal{C} = \{1, 2, \cdots, C\}$ is the label. If the number of images per category is K, the task is denoted as C-way K-shot classification, which classifies the images from a query set by assigning each image with a label from $\mathcal{C}$. The support set and query set together form the test data. Typically, K is small for few-shot classification, e.g. $K = 1$ or 5. Hence, it is difficult to train an effective ConvNet over the support set directly. We follow the metric learning based solution to train a model over another large labeled dataset. This additional dataset is called the training dataset, whose label space is disjoint with the support set. The model is trained to measure the similarity between images. By exploiting the object-level relation (see the next section), the generated similarity is effective for the support and query images for unseen categories as well.

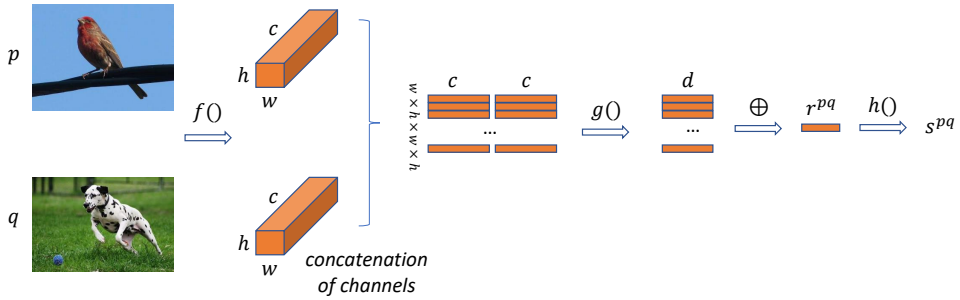


Figure 2: Dataflow of our model. p, q stand for the query image and one image from the support set respectively. f() is a CNN for feature extraction; g() is network for object relation learning; h() is a network for image relation learning; $\oplus$ denotes element-wise addition.

3.2 One-shot Classification

Our model is created based on two intuitions. First, object-level relation persists across the training and test images, even though they have disjoint label space. For example, if we have scooters and motors in the training dataset, then the knowledge about wheels and riders is helpful for recognizing Segway from the test dataset. Second, to avoid the challenge caused by gradient based optimization

algorithms (see Section 1), nearest neighbor search (NNS) is adopted as the classifier, which is non-parametric. For NNS, the distance (or similarity) is vital for good classification, which must enforce sufficient margin between similar and dis-similar pairs.

Based on the above intuitions, we create our model as shown in Figure 2. We first explain the dataflow for the case of $K = 1$, i.e. one-shot classification. The next subsection introduces the dataflow for $K > 1$. The input of the model consists of two images denoted as p and q, where p is the query image and q is an image from the support set. They go through the same ConvNet, denoted as f, to extract feature maps, i.e. $f^p \in R^{w \times h \times c}$, $f^q \in R^{w \times h \times c}$. w (reps. h) is the width (resp. height) of each feature map and c is the number of feature maps. Each position (i, j) on the feature map corresponds to a patch of the input image, which is considered as an "object"[1]. Therefore, we treat $f^p_{i,j} \in R^c$ as an object level feature. Next, f^p and f^q are combined for comparison and object level relation learning. In particular, we concatenate all object features from f^p and f^q, i.e. $[f^p_{i_p,j_p}; f^q_{i_q,j_q}]$, for all $i_p \in [1, w], j_p \in [1, h], i_q \in [1, w], j_q \in [1, h]$. As a result, we get a matrix of $w \times h \times w \times h$ rows and $2c$ columns. g is another network that extracts the relation feature $\in R^d$ between every pair of object features (Equation 2). All $w \times h \times w \times h$ object relations are aggregated via element-wise addition to get the image relation feature r^{ap} (Equation 3), which is transformed by the similarity network $h()$ to get the similarity of p and q, denoted as $s^{pq} \in (0, 1)$ (Equation 4). s^{pq} is normalized by the Sigmoid function.

$$f^p = f(p) \tag{1}$$

$$g^{pq}_{i_p,j_p,i_q,j_q} = g([f^p_{i_p,j_p}; f^q_{i_q,j_q}]) \tag{2}$$

$$r^{pq} = \sum_{i_p,j_p \in [1,w]; i_q,j_q \in [1,h]} g^{pq}_{i_p,j_p,i_q,j_q} \tag{3}$$

$$s^{ap} = sigmoid(h(r^{pq})) \tag{4}$$

3.3 K-SHOT CLASSIFICATION

For the case of $K > 1$, i.e. there is more than one example per category in the support set, the feature maps of images from the same category are averaged, i.e., f^q. The averaged feature maps are deemed as the category feature, which is also used in Prototypical Network Snell et al. (2017). The dataflow for the remaining steps is the same as for one-shot classification. The extracted relation is between the query image p and the category q.

3.4 TRAINING AND INFERENCE

Similar to the training procedure of meta learning, we train our model in episodes. For each episode, we sample C classes as the support set; and for each class, we sample K images. In addition, for each episode, some query images are sampled, which share the same label space as the support images. Then the training is similar to the test scenario, i.e. C-way K-shot. By varying the value of K, we can train and test different tasks, including one-shot and few-shot classification. During training, for each query p, we randomly sample an image q. If p and q are from the same class, then the ground truth similarity is set to $t^{pq} = 1$; otherwise, the ground truth similarity is set to $t^{pq} = 0$. The normalized similarity score s^{pq} is compared with the ground truth via cross-entropy loss (Equation 5). Under this episode based training strategy, the model is trained to maximize the ability of differentiating different classes within the same episode. For inference, the query image and each image from the support set are fed into the model to compute their similarity following Figure 2. The class with the largest similarity as the query is assigned to the query image as the classification result.

$$L(p, q) = -t^{pq} log s^{pq} - (1 - t^{pq}) log(1 - s^{pq}) \tag{5}$$

[1] It may not be a complete object.

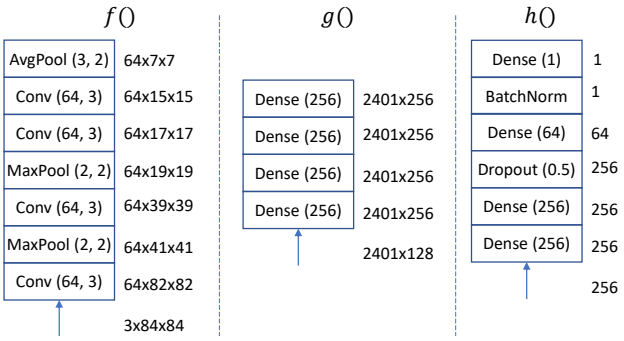


Figure 3: Details of the network architecture for omniglot.

4 Experimental Evaluation

Our experimental evaluation is to answer the question: Can exploiting object-level relation help improve few-shot learning performance? The results on public benchmark datasets confirm that our approach does improve few-shot learning.

We evaluate our approach on two public benchmark datasets: Omniglot, and miniImageNet. For each dataset, we partition it into training, validation and testing subsets. The C-way K-shot classifier is trained by sampling C classes and K examples per class for each training episode. We introduce the experiments on the two datasets respectively in the following subsections.

4.1 Evaluation on Omniglot

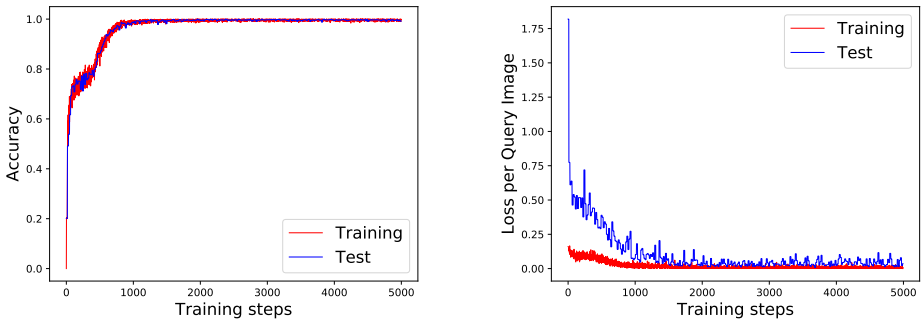
Omniglot Lake et al. (2015) contains 1623 characters (classes) from 50 different alphabets. Each class has 20 samples drawn by different people. We use 1200 classes for training (including validation), and the remaining 423 classes for testing. Following Sung et al. (2017); Snell et al. (2017), all input images are augmented by rotations in multiples of 90 degrees. In every testing episode, 15 query images per class are tested.

Table 1: Performance comparison on Omniglot dataset.

MODEL	**FINE TUNE**	**5-WAY**		**20-WAY**	
		1-shot	5-shot	1-shot	5-shot
Conv Siamese Nets Koch et al. (2015)	N	96.7%	98.4%	88.0%	96.5%
Conv Siamese Nets Koch et al. (2015)	Y	97.3%	98.4%	88.1%	97.0%
MANN Santoro et al. (2016)	N	82.8%	94.9%	-	-
Matching Nets Vinyals et al. (2016)	N	98.1%	98.9%	93.8%	98.5%
Matching Nets Vinyals et al. (2016)	Y	97.9%	98.7%	93.5%	98.7%
Neural Stat Edwards & Storkey (2016)	N	98.1%	99.5%	93.2%	98.1%
ConvNet with Memory Kaiser et al. (2017)	N	98.4%	99.6%	95.0%	98.6%
Meta Nets Munkhdalai & Yu (2017)	N	99.0%	-	97.0%	-
Prototypical Nets Snell et al. (2017)	N	98.8%	99.7%	96.0%	98.9%
MAML Finn et al. (2017)	Y	98.7±0.4%	**99.9±0.1%**	95.8±0.3%	98.9±0.2%
Meta-SGD Li et al. (2017)	Y	99.5±0.3%	**99.9±0.1%**	95.9±0.4%	99.0±0.2%
Learning2Compare Sung et al. (2017)	N	99.6±0.2%	99.8±0.1%	97.6±0.2%	99.1±0.1%
Ours	N	**99.8±0.1%**	**99.9±0.1%**	**98.2 ±0.1%**	**99.5±0.1%**

The detailed configuration of our networks is illustrated in Figure 3. 'Conv' denotes a block of 3 layers, namely convolution layer, batch normalization layer and ReLU layer. The associated numbers in the box are for the number of filters and kernel size respectively. The numbers on the right side of each box are the output feature map shape, which is interpreted as (number of channels, height, width). The numbers associated with 'MaxPool' (resp 'AvgPool') stand for the pooling kernel size and stride size. Previous papers Sung et al. (2017); Snell et al. (2017) resize the images to 28x28 or 20x20, which results in small feature maps from the last convolution layer, e.g. 1x1x64; In order to get a large feature map for object relation modeling, we resize the input images to 84x84. Consequently, the

output from f has 64 feature maps, each of size 7x7. Therefore, there are $(7 \times 7) \times (7 \times 7) = 2401$ combinations, i.e object relation features, each of size $64 + 64 = 128$. The g network processes these 2401 object relation features independently through a MLP model. The configuration of hidden layers follows RelationNet Santoro et al. (2017). The output feature of each relation is of dimension 256. All features are summed over into a single feature, which is then fed into h to generate the similarity score. All omniglot experiments are trained with Adam Kingma & Ba (2014) with a learning rate of 0.001 and no weight decay.



(a) (b)

Figure 4: Training and testing of 5-way 1-shot over omniglot dataset.

From Figure 4(b), we can see that overfitting is not a problem for our model even weight decay is 0, the datasets are not large and there are many fully connected layers in g and f. One possible reason is that averaging the object relation features has similar effect as ensemble modeling.

Following the experimental setting in previous papers, we compare our approach with existing methods on four tasks, namely 5-way 1-shot, 5-way 5-shot, 20-way 1-shot and 20-way 5 shot classification. The results in terms of classification accuracy are presented in Table 1. For existing methods, we copy their performance reported in the original papers or other published papers. Both meta-learning and metric-learning based approaches are compared. Meta-learning based solutions need to fine-tune the model over the test support dataset. For metric-learning based approaches, fine-tuning is not necessary. It may improve or decrease the performance as reported by Matching Nets Vinyals et al. (2016), shown in the 3rd and 4th rows in the table. The second column indicates whether the model is fined-tuned over the test support set or not. Our results are averaged over 600 test episodes and are reported with 95% confidence intervals. The variance is also reported. We can see that our approach outperforms existing methods for 3 out of 4 tasks. Note that the accuracy of existing solutions are very high, especially for 5-way tasks. Hence, a small improvement over the state-of-the-art should be considered as significant. The improvement for 20-way tasks is clearer. 20-way tasks are more difficult than 5-way tasks as the model needs to be more discriminative to differentiate more classes. To confirm the advantage of our approach, we perform comparison against another difficult dataset in the next subsection.

4.2 Evaluation on miniImageNet

The miniImagenet dataset Vinyals et al. (2016) consists of 60,000 colour images with 100 classes sampled from ImageNet Deng et al. (2009). Each class has 600 examples. We follow the partition scheme as in the original paper Vinyals et al. (2016) to get 64, 16, 20 classes for training, validation and testing, respectively. We resize the images to 224x224 and do channel-wise standardization. No data augmentation is conducted. miniImageNet is a more difficult benchmark than Omniglot because it has a larger number of classes and greater variations among the images within each class.

The configuration of f and g is shown in Figure 5. The network of f is almost the same as that in Figure 3 except the final average pooling layer has a larger kernel and stride size. This is to reduce the memory cost caused by large input images. f generates 64 feature maps, each of size 10x10. Consequently, we have $(10 \times 10) \times (10 \times 10) = 10,000$ combinations of object features. g processes the 10,000 combinations independently via a 4 layer MLP model. The output is summed over to

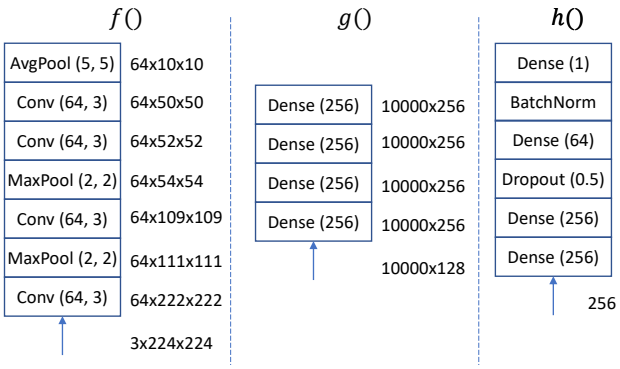


Figure 5: Details of the network architecture for miniImageNet.

generate a 256-d feature. The h network from Figure 3 is used again to generate the image relation feature. All miniImageNet experiments are trained with Adam Kingma & Ba (2014) with a learning rate of 0.001 and no weight decay.

Four tasks are conducted to do the evaluation, namely, 5-way 1-shot, 5-way 5-shot, 20-way 1-shot and 20-way 5-shot classification. In Table 2, we report the classification accuracy including the mean and variance over 600 test episodes. The performance of existing methods are copied from their original papers or other papers. We can see that our model achieves significant improvement over existing methods, especially compared with other naive version of models. Again, the margin is bigger for 20-way tasks. The above observations are consistent with the results on omniglot dataset, which indicates that our approach has larger capacity in modelling more difficult tasks.

Table 2: Performance comparison on miniImageNet dataset.

MODEL	**FINE TUNE**	**5-WAY**		**20-WAY**	
		1-shot	5-shot	1-shot	5-shot
META-SGD	Y	$50.5 \pm 1.9\%$	$64.0 \pm 1.0\%$	$17.6 \pm 0.6\%$	$29.0 \pm 0.4\%$
MATCHING NETS	N	$43.6 \pm 0.8\%$	$55.3 \pm 0.7\%$	$17.3 \pm 0.2\%$	$22.7 \pm 0.2\%$
META LSTMRavi & Larochelle (2016)	N	$43.4 \pm 0.8\%$	$60.6 \pm 0.7\%$	$16.7 \pm 0.2\%$	$26.1 \pm 0.3\%$
MAML	Y	$48.7 \pm 1.8\%$	$63.1 \pm 1.0\%$	$16.5 \pm 0.6\%$	$19.3 \pm 0.3\%$
META NETS	N	$49.2 \pm 0.9\%$	-	-	-
PROTOTYPICAL NETS	N	$49.4 \pm 0.8\%$	$68.2 \pm 0.7\%$	-	-
LEARNING2COMPARE	N	$51.4 \pm 0.8\%$	$67.1 \pm 0.7\%$	-	-
TCML Mishra et al. (2017)	N	$55.7 \pm 1.0\%$	$68.9 \pm 0.9\%$	-	-
LEARNING2COMPARE DEEP	N	$50.4 \pm 0.8\%$	$65.3 \pm 0.7\%$	-	-
OURS	N	**59.0 ± 1.0%**	**70.9± 0.5%**	**22.2 ± 0.3%**	**32.2± 0.2%**

5 DISCUSSION

In this paper, we exploit the object-level relation to infer the image relation. In particular, we consider each 'pixel' on the feature map as an object in the input image, and use the values across all channels as the object feature. In fact, one 'pixel' corresponds to one patch of the original image. Small patches may not contain any objects, while in big patches, there could be multiple objects. In the extreme case where the feature map size is 1x1, the corresponding patch is the whole image. Our model is then equivalent to LearningToCompare Sung et al. (2017). In fact, we capture object pairs from different locations, whereas LearningToCompare is restricted to element-wise match at the same spatial location. The effectiveness of our approach from the experimental study confirms that the relation extracted from multiple local patches is useful for determining the image relation. Particularly, we do one more set of comparison against Learning2Compare since it has similar architecture as our model. We change the input image size of Learning2Compare to 224x224, i.e., the same as our model. The accuracy for 5-way 1-shot and 5-way 5-shot is 50.16% and 65.98% respectively. In addition, we compare effect of the feature map size of our model. We vary the feature map size as

1x1, 3x3, 5x5, 7x7 and 9x9. The corresponding accuracy of 5-way 5-shot classification is 64.55%, 67.78%, 69.68%, 69.82%, 70.90%. The improvement becomes marginal for bigger sizes. We can see that with the increasing of objects number (feature map size), the performance improves. It indicates that the objects-level relation does make a difference.

The basic idea behind our model is to aggregate the local information for global reasoning. In this paper, we consider the spatial local information. However, the framework is extensible for other local information. For example, if we treat different feature maps as different aspects (or features) of the image, in Figure 2, we can compare (combine) these feature maps from two images to get $c \times c$ local relation features, each of size $w \times h + w \times h$. Similarly, we can compare the features of different attributes of each class with the feature maps to do zero-shot learning. It compares the local text information with local image information.

6 CONCLUSION

ConvNets have shown great success for image classification with many examples per category. However, few-shot learning is challenging because the training algorithms of ConvNets, i.e. gradient based optimization algorithms, require many iterations to fine-tune the parameters over a lot of examples for new image classes. In this paper, we avoid the fine-tuning step by training a model that is general to learn the relation of images from unseen categories. We observe that the object-level relation persists across training and test images, although the relation of images from test datasets is unseen. Therefore, we propose to exploit object-level relation to infer the image relation. In particular, object features are compared (combined) and transformed to extract the image relation feature, which is applied directly for similarity learning. Using the learned similarity from the relation feature, our approach outperforms existing algorithms for few-shot image classification tasks.

REFERENCES

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

H. Edwards and A. Storkey. Towards a Neural Statistician. *ArXiv e-prints*, June 2016.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400, 2017. URL http://arxiv.org/abs/1703.03400.

Lukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to remember rare events. *CoRR*, abs/1703.03129, 2017. URL http://arxiv.org/abs/1703.03129.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. *ICML Deep Learning Workshop*, 2015.

Gregory R. Koch. Siamese neural networks for one-shot image recognition. 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pp. 1106–1114, 2012.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. ISSN 0036-8075. doi: 10.1126/science.aab3050. URL http://science.sciencemag.org/content/350/6266/1332.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. *CoRR*, abs/1707.09835, 2017. URL http://arxiv.org/abs/1707.09835.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. Meta-learning with temporal convolutions. *CoRR*, abs/1707.03141, 2017. URL http://arxiv.org/abs/1707.03141.

Tsendsuren Munkhdalai and Hong Yu. Meta networks. *CoRR*, abs/1703.00837, 2017. URL http://arxiv.org/abs/1703.00837.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *ICLR*, 2016.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy P. Lillicrap. One-shot learning with memory-augmented neural networks. *CoRR*, abs/1605.06065, 2016. URL http://arxiv.org/abs/1605.06065.

Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy P. Lillicrap. A simple neural network module for relational reasoning. *CoRR*, abs/1706.01427, 2017. URL http://arxiv.org/abs/1706.01427.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175, 2017. URL http://arxiv.org/abs/1703.05175.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. *CoRR*, abs/1711.06025, 2017. URL http://arxiv.org/abs/1711.06025.

Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *CoRR*, abs/1606.04080, 2016. URL http://arxiv.org/abs/1606.04080.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *CoRR*, abs/1411.1792, 2014. URL http://arxiv.org/abs/1411.1792.