TASK LEVEL DATA AUGMENTATION FOR META-LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Data augmentation is one of the most effective approaches for improving the accuracy of modern machine learning models, and it is also indispensable to train a deep model for meta-learning. However, most current data augmentation implementations applied in meta-learning are the same as those used in the conventional image classification. In this paper, we introduce a new data augmentation method for meta-learning, which is named as “Task Level Data Augmentation” (referred to Task Aug). The basic idea of Task Aug is to increase the number of image classes rather than the number of images in each class. In contrast, with a larger amount of classes, we can sample more diverse task instances during training. This allows us to train a deep network by meta-learning methods with little over-fitting. Experimental results show that our approach achieves state-of-the-art performance on miniImageNet, CIFAR-FS, and FC100 few-shot learning benchmarks. Once paper is accepted, we will provide the link to code.

1 INTRODUCTION

Although the machine learning systems have achieved a human-level ability in many fields with a large amount of data, learning from a few examples is still a challenge for modern machine learning techniques. Recently, the machine learning community has paid significant attention to this problem, where few-shot learning is the common task for meta-learning (e.g., Ravi & Larochelle (2017); Finn et al. (2017); Vinyals et al. (2016); Snell et al. (2017)). The purpose of few-shot learning is to learn to maximize generalization accuracy across different tasks with few training examples. In a classification application of the few-shot learning, tasks are generated by sampling from a conventional classification dataset; then, training samples are randomly selected from several classes in the classification dataset. In addition, a part of the examples is used as training examples and testing examples. Thus, a tiny learning task is formed by these examples. The meta-learning methods are applied to control the learning process of a base learner, so as to correctly classify on testing examples.

Data augmentation is widely used to improve the training of deep learning models. Usually, the data augmentation is regarded as an explicit form of regularization He et al. (2016); Simonyan & Zisserman (2014); Krizhevsky et al. (2012). Thus, the data augmentation aims at artificially generating the training data by using various translations on existing data, such as: adding noises, cropping, flipping, rotation, translation, etc. The general idea of data augmentations is increasing the number of data by change data slightly to be different from original data, but the data still can be recognized by human. The new data involved in the classes are identical to the original data. However, the minimum units of meta-learning are the tasks rather than data. Increasing the data of original class cannot increase the types of task instances. Therefore, “Task Aug” increases the data that can be clearly recognized as the different classes as the original data. With novel classes, the more diverse task instances can be generated. This is important for the meta-learning, since meta-learning models must predict unseen classes during the testing phase. Therefore, a larger number of classes is helpful for models to generate task instances with different classes.

In this work, the natural images are augmented by being rotated 90, 180, 270 degrees (we show examples in Figure 1). We compare two cases, 1) the new images are converted to the classes of original images and 2) the new images are separated to the new classes. The proposed method is evaluated by experiments with the state of art meta-learning Methods Snell et al. (2017); Lee

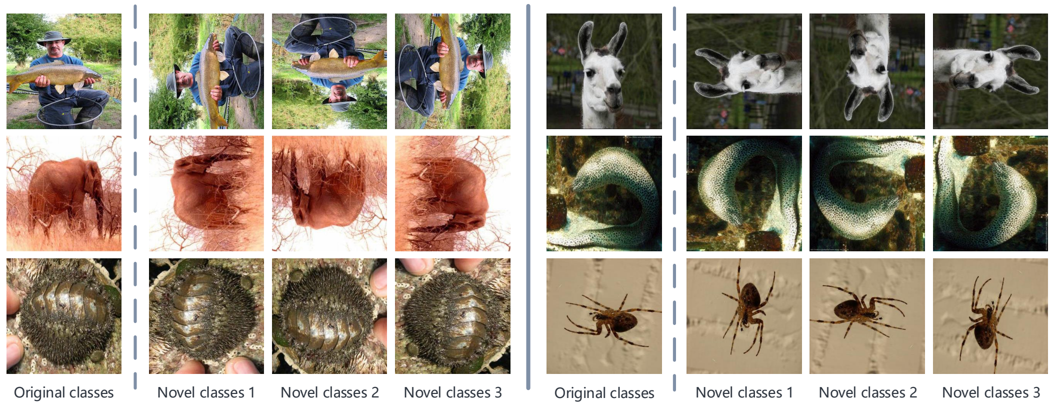

Figure 1: Examples of the novel created classes.

et al. (2019); Bertinetto et al. (2018) on CIFAR-FS, FC100, miniImageNet few-shot learning tasks with the standard training protocol, and the training protocol with ensemble method Huang et al. (2017). The experimental result analysis shows that Task Aug can reduce over-fitting and improve the performance, while the conventional data augmentation (referred to Data Aug) of rotation, which converts the novel data into the classes of original data, does not improve the performance and even causes the worse result. In the comparative experiments, Task Aug achieves the best accuracy of the meta-learning methods applied. Besides, the best results of our experiments exceed the current state-of-art result over a large margin.

2 Related Work

Meta-learning involves two hierarchies learning processes: low-level and high-level. The low-level learning process learns to deal with general tasks, often termed as the “inner loop”; and the high-level learning process learns to improve the performance of a low-level task, often termed as the “outer loop”. Since models are required to handle sensory data like images, deep learning methods are often applied for the “outer loop”. However, the machine learning methods applied for the “inner loop” are very diverse. Based on different methods in the “inner loop”, meta-learning can be applied in image recognition Fei-Fei et al. (2006); Santoro et al. (2016); Finn et al. (2017); Vinyals et al. (2016); Ravi & Larochelle (2017), image generation Antoniou et al. (2017); Zhang et al. (2018); Rezende et al. (2016), reinforce learning Finn et al. (2017); Al-Shedivat et al. (2017), and etc. This work focuses on few-shot learning image recognition based on meta-learning. Therefore, in the experiment, the methods applied in the “inner loop” are able to classify data, and they are K-nearest neighbor (KNN), Support Vector Machine (SVM) and ridge regression, respectively Snell et al. (2017); Lee et al. (2019); Bertinetto et al. (2018).

Previous studies have introduced many popular regularization techniques to few-shot learning from deep learning, such as weight decay, dropout, label smooth Bertinetto et al. (2018), and data augmentation. Common data augmentation techniques for image recognition are usually designed manually and the best augmentation strategies depend on dataset. However, in natural color image datasets, random cropping and random horizontal flipping are the most common. Since the few-shot learning tasks consist of natural color images, the random horizontal flipping and random cropping are applied in few-shot learning. In addition, color (brightness, contrast, and saturation) jitter is often applied in the works of few-shot learning Gidaris & Komodakis (2018); Qiao et al. (2018).

Other data augmentation technologies related to few-shot learning include generating samples by few-shot learning and generating samples for few-shot learning. The former tried to synthesize additional examples via transferring, extracting, and encoding to create the data of the new class, that are intra-class relationships between pairs of reference classes’ data instances Hariharan & Girshick (2017); Schwartz et al. (2018). The later tried to apply meta-learning in a few-shot generation to generate samples from other models Antoniou et al. (2017). In addition to these two types of studies, the data augmentation technology most closed to the new proposed approach is applied to

Omniglot dataset, which consists of handwritten words Lake et al. (2015). They created the novel classes by rotating the original images 90, 180 and 270 degrees Santoro et al. (2016). However, this approach cannot be applied for the natural color image directly, and we will explain the reasons and the solutions in Section 3.

3 METHOD

3.1 PROBLEM DEFINITION

We adopt the formulation purposed by Vinyals et al. (2016) to describe the N-way K-shot task. A few-shot task contains many task instances (denoted by $\mathcal{T}_i$), each instance is a classification problem consisting of the data sampled from N classes. The classes are randomly selected from a classes set. The classes set are split into M^{tr}, M^{val} and M^{test} for a training class set $\mathcal{C}^{tr}$, a validation classes set $\mathcal{C}^{val}$, and a test classes set $\mathcal{C}^{test}$. In particular, each class cannot overlap others (i.e., the classes used during testing are unseen classes during training). Data is randomly sampled from $\mathcal{C}^{tr}$, $\mathcal{C}^{val}$ and $\mathcal{C}^{test}$, so as to create task instances for training meta-set S^{tr}, validation meta-set S^{val}, and test meta-set S^{test}, respectively. The validation and testing meta-sets are used for model selection and final evaluation, respectively.

The data in each task instance, $\mathcal{T}_i$, are divided into training examples D^{tr} and validation examples D^{val}. Both of them only contains the data from N classes which sampled from the appropriate classes set randomly (for a task instance applied during training, the classes form a subset of the training classes set $\mathcal{C}^{tr}$). In most settings, the training set $\mathcal{D}^{tr} = \{(\mathrm{x}_n^k, \mathrm{y}_n^k)|n = 1 \ldots N; k = 1 \ldots K\}$ consists of K data instances from each class, this processing usually called as a "shot". The validation set, $\mathcal{D}^{val}$, consists of several other data instances from the same classes, this processing is usually called as a "query". An evaluation is provided for generalization performance on the N classification task instance $\mathcal{D}^{tr}$. Note that: the validation set of a task instance $\mathcal{D}^{val}$ (for optimizing model during "outer loop") is different from the held-out validation classes set $\mathcal{C}^{val}$ and meta-set $\mathcal{S}^{val}$ (for model selection).

3.2 CREATING CLASSES BY ROTATING CLASSES

This work is to increase the size of the training classes set, M^{tr}, by rotating all images within the training classes set with 90, 180, 270 degrees. The size, M^{tr}, is increased for three times. In the Omniglot dataset consisting of handwritten words Santoro et al. (2016), this approach works well, since it can rotate a handwritten word multiple of 90 degrees and treat the new one as another word; in addition, it is really possible that the novel word is similar to some words, which are not included in the training classes but existed. However, for natural images, it is not the same cause. For examples, the images in the third line of Figure 1 are difficult to identify which images are rotated. Moreover, the images are rarely rotated in the photos taken by humans.

Despite the two problems, the fundamental features of the novel images can provide useful information. We assign novel classes that contain smaller weights than the original classes, so as to make models prioritize learning the features of the original classes, and make the features of the novel classes as a supplement to prevent the augmented data from taking up large capacity in the model.

The smaller weights are implemented in two ways, 1) lower probability and 2) delay selecting the novel classes. For a class in a task instance, the probability of the class coming from the novel classes is p, and the probability coming from the original classes is $1 - p$. Besides, The initial p is set to 0, then linearly rises from 0 to p_{max} after generating T task instances. The max probability p_{max} is set lower than the proportion of the novel classes in all classes to make each novel class have a lower probability than each original class. The whole process of Task Aug on a classes set is summarized in Algorithm 1 and Figure 2.

3.3 ENSEMBLE

In this work, we also compare the methods with the training protocol with ensemble method Huang et al. (2017) in addition to the standard training protocol, which choosing a model by the validation set. The training protocol with an ensemble method use the models with different training epoch to

Algorithm 1 Task Level Data Augmentation.

```
Require: Classes set $\mathcal{C} = \{c_1, c_2, \ldots, c_M\}$; Max possibility for Task Aug $p_{max}$; The delay to Task Aug T; The current count $t$; The number of ways, shots and queries $N$, $K$, $H$
$t \leftarrow t + 1$
$p \leftarrow p_{max} * \min\{1, \frac{t}{\mathrm{T}}\}$
$n \sim Binomial(N, p)$
$\mathcal{D}^{tr}, \mathcal{D}^{val} \leftarrow \{\}, \{\}$
$\mathcal{V} \leftarrow$ Sample $N - n$ from $\{1, 2, \cdots, M\}$
for all $v \in \mathcal{V}$ do
    $\mathcal{D} \leftarrow$ Sample $K + H$ from $c_v$
    $\mathcal{D}^{tr} \leftarrow \mathcal{D}^{tr} \cup$ First $K$ of $\mathcal{D}$
    $\mathcal{D}^{val} \leftarrow \mathcal{D}^{val} \cup$ Last $H$ of $\mathcal{D}$
end for
$\mathcal{U} \leftarrow$ Sample $n$ from $\{M, M+1, \cdots, 4M\}$
for all $u \in \mathcal{U}$ do
    $v \leftarrow (u \mod M) + 1$
    $\mathcal{D} \leftarrow$ Sample $K + H$ from $c_v$
    $r \leftarrow \lfloor \frac{u}{M} \rfloor$
    Rotate all $\mathrm{x} \in \{\mathrm{x} | (\mathrm{x}, \mathrm{y}) \in \mathcal{D}\}$ $90r$ degrees
    $\mathcal{D}^{tr} \leftarrow \mathcal{D}^{tr} \cup$ First $K$ of $\mathcal{D}$
    $\mathcal{D}^{val} \leftarrow \mathcal{D}^{val} \cup$ Last $H$ of $\mathcal{D}$
end for
return $(\mathcal{D}^{tr}, \mathcal{D}^{val})$
```

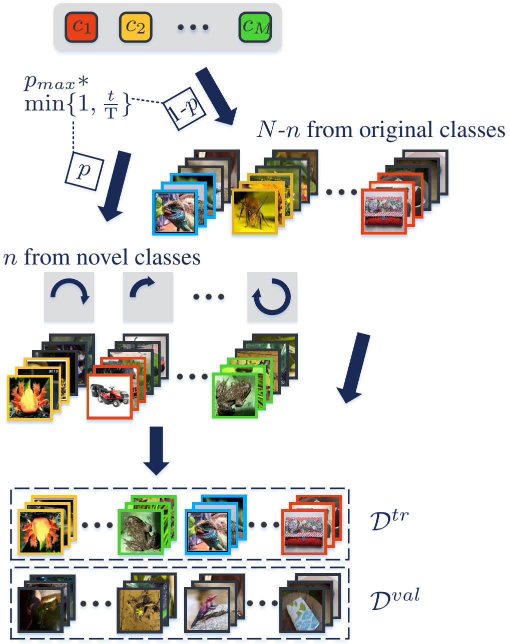

Figure 2: The process of generating a task instance with Task Aug.

an ensemble model, in order to better use the models obtained in a single training process, and this approach has been proved to be valid for meta-learning by experiments Liu et al. (2018). We adopt this ensemble method. However, unlike Huang et al. (2017) and Liu et al. (2018) that we did not use cyclic annealing for learning rate and any methods to select models. We directly took the average of the prediction of all models, which are saved according to an interval of 1 epoch. In Section 4, the methods with this ensemble approach are marked by "+ens".

4 Experiments

We evaluate the proposed method on few-shot learning tasks. In order to ensure fair, both the results of baseline and Task Aug were run in our own environment. The comparative experiment is designed to answer the following questions: (1) Is Task Aug able to improve the performance of meta-learning? (2) How much should the probably for the novel classes be set? (3) Will converting the novel data into the classes of the original data cause worse results, which are generated by being rotated 90, 180, 270 degrees?

4.1 Experimental Configuration

4.1.1 Backbone

Following Lee et al. (2019); Oreshkin et al. (2018); Mishra et al. (2017), we used ResNet-12 network in our experiments. The ResNet-12 network had four residual blocks which contains three 3×3 convolution, batch normalization and Leaky ReLU with 0.1 negative slope. One 2×2 max-pooling layer is used for reducing the size of the feature map. The numbers of the network channels were 64, 160, 320 and 640, respectively. DropBlock regularization Ghiasi et al. (2018) is used in the last two residual blocks, the conventional dropout Hinton et al. (2012) is used in the first two residual blocks. The block sizes of DropBlock were set to 2 and 5 for CIFAR derivatives and ImageNet derivatives, respectively. In all experiments, the dropout possibility was set to 0.1. The global average pooling was not used for the final output of the last residual block.

Table 1: The average accuracies (%) with 95% confidence intervals. [*]CIFAR-FS results from Bertinetto et al. (2018). [†]Result from Lee et al. (2019). The best results are highlighted.

Method	**CIFAR-FS 5-way**		**FC100 5-way**	
	1-shot	**5-shot**	**1-shot**	**5-shot**
MAML[*] Finn et al. (2017)	58.9±1.9	71.5±1.0	-	-
R2-D2 Bertinetto et al. (2018)	65.3±0.2	79.4±0.1	-	-
TADAM Oreshkin et al. (2018)	-	-	40.1±0.4	56.1±0.4
ProtoNets[†] Snell et al. (2017)	72.2±0.7	83.5±0.5	37.5±0.6	52.5±0.6
MTL Sun et al. (2019)	-	-	45.1±1.8	57.6±0.9
M-SVM Lee et al. (2019)	72.8±0.7	85.0±0.5	47.2±0.6	62.5±0.6
M-SVM (+ens+val) (our)	76.75±0.46	**88.38±0.33**	49.83±0.45	67.14±0.42
R2-D2 (+ens+val) (our)	**77.66±0.46**	88.33±0.33	**50.91±0.45**	**67.46±0.42**

4.1.2 Base Leaners

We used ProtoNets Snell et al. (2017), MetaOptNet-SVM Lee et al. (2019) (we write it as M-SVM) and Ridge Regression Differentiable Discriminator (R2-D2) Bertinetto et al. (2018) as basic methods to verify the effective of Task Aug.

For ProtoNets, we did not use a higher way for training than testing like Snell et al. (2017). Instead, the equal number of shot and way were used in both training and evaluation, and its output multiplied by a learnable scale before the softmax following Oreshkin et al. (2018); Lee et al. (2019),

For M-SVM, we set training shot to 5 for CIFAR-FS; 15 for FC100; and 15 for miniImageNet; regularization parameter of SVM was set to 0.1; and a learnable scale was used following Lee et al. (2019). We did not use label smoothing like Lee et al. (2019), because we did not find that label smoothing can improve the performance in our environment. This was also affirmed from the Lee et al. (2019) author's message on GitHub, that Program language packages and environment might affect results of the meta-learning method.

For R2-D2, we set the same training shot as for M-SVM, and used a learnable scale and bias following Bertinetto et al. (2018). It was different from Bertinetto et al. (2018) we used a fixed regularization parameter of ridge regression which was set to 50 because Bertinetto et al. (2018) has confirmed that making it learnable might not be helpful.

Last, for all methods, each class in a task instance contained 6 test (query) examples during training and 15 test (query) examples during testing.

4.1.3 Training Configuration

Stochastic gradient descent (SGD) was used. Following Sutskever et al. (2013), we set weight decay and Nesterov momentum to 0.0005 and 0.9, respectively. Each mini-batch contained 8 task instances. The meta-learning model was trained for 60 epochs, and 1000 mini-batchs for each epoch. We set the initial learning rate to 0.1, then multiplied it by 0.06, 0.012, and 0.0024 at epochs 20, 40 and 50, respectively, as in Gidaris & Komodakis (2018). The results, which are marked by "ens" were used the 60 models saved after each epoch to become an ensemble model. For the final epoch, the training classes set was augmented by the validation classes set. We chose the model at the epoch where we got the best model during training on the training classes set only. The results of the final run are marked by "+val" in this subsection.

For data augmentation, we adopted random random crop, horizontal flip, and color (brightness, saturation, and contrast) jitter data augmentation following the work of Gidaris & Komodakis (2018); Qiao et al. (2018). We set p_{max} to 0.5 for CIFAR-FS and FC100; 0.25 for miniImageNet; and T was set to 80000 for all experiments.

Table 2: Comparison to the average accuracies (%) with 95% confidence intervals between the methods with and without Task Aug on CIFAR-FS 5-way. The results are better than its compared results are highlighted.

Method	**1-shot**		**5-shot**	
	Baseline	Task Aug	Baseline	Task Aug
ProtoNets Snell et al. (2017)	71.88±0.52	**74.15±0.50**	84.14±0.36	**85.37±0.35**
ProtoNets (+ens)	73.95±0.51	**75.89±0.48**	85.72±0.35	**87.33±0.33**
ProtoNets (+val)	73.20±0.51	**75.10±0.49**	85.29±0.35	**86.53±0.34**
ProtoNets (+ens+val)	76.05±0.49	**77.28±0.47**	86.88±0.34	**88.24±0.33**
M-SVM Lee et al. (2019)	71.52±0.51	**72.95±0.48**	84.01±0.36	**85.91±0.36**
M-SVM (+ens)	74.12±0.50	**75.85±0.47**	85.85±0.34	**87.73±0.33**
M-SVM (+val)	72.42±0.50	**73.13±0.47**	84.94±0.36	**86.94±0.34**
M-SVM (+ens+val)	75.91±0.48	**76.75±0.46**	87.15±0.34	**88.38±0.33**
R2-D2 Bertinetto et al. (2018)	72.27±0.51	**74.42±0.48**	84.60±0.36	**86.02±0.35**
R2-D2 (+ens)	75.06±0.50	**76.51±0.47**	86.11±0.34	**87.63±0.34**
R2-D2 (+val)	73.52±0.50	**76.02±0.47**	85.39±0.36	**86.73±0.34**
R2-D2 (+ens+val)	76.40±0.49	**77.66±0.46**	87.04±0.34	**88.33±0.33**

Table 3: Comparison to the average accuracies (%) with 95% confidence intervals between the methods with and without Task Aug on FC100 5-way. The results are better than its compared results are highlighted.

Method	**1-shot**		**5-shot**	
	Baseline	Task Aug	Baseline	Task Aug
ProtoNets Snell et al. (2017)	37.53±0.40	**39.07±0.40**	51.43±0.39	**54.49±0.40**
ProtoNets (+ens)	40.04±0.41	**42.23±0.42**	54.24±0.40	**57.47±0.40**
ProtoNets (+val)	43.63±0.43	**45.85±0.45**	**61.16±0.42**	61.05±0.41
ProtoNets (+ens+val)	47.16±0.46	**48.05±0.46**	63.64±0.43	**65.63±0.42**
M-SVM Lee et al. (2019)	40.50±0.39	**40.54±0.39**	54.83±0.40	**56.88±0.41**
M-SVM (+ens)	43.24±0.42	**44.00±0.42**	58.49±0.41	**60.01±0.41**
M-SVM (+val)	46.72±0.45	**47.13±0.44**	62.99±0.42	**63.52±0.41**
M-SVM (+ens+val)	49.50±0.46	**49.83±0.45**	66.37±0.42	**67.14±0.42**
R2-D2 Bertinetto et al. (2018)	40.66±0.41	**41.34±0.41**	**55.85±0.39**	55.83±0.39
R2-D2 (+ens)	43.27±0.42	**44.45±0.43**	58.01±0.40	**59.82±0.41**
R2-D2 (+val)	47.12±0.44	**47.78±0.44**	63.32±0.40	**63.73±0.43**
R2-D2 (+ens+val)	49.92±0.45	**50.91±0.45**	65.58±0.42	**67.46±0.42**

4.2 Evaluation on CIFAR derivatives

The **CIFAR-FS** Bertinetto et al. (2018) containing all 100 classes from CIFAR-100 Krizhevsky et al. (2010) is proposed as few-shot classification benchmark recently. These classes are randomly divided into training classes, validation classes and test classes. The three types contain 64, 16 and 20 classes, respectively. There are 600 nature color images of size 32 × 32 in each class.

The **FC100** Oreshkin et al. (2018) are also derived from CIFAR-100 Krizhevsky et al. (2010), and the 100 classes are grouped into 20 superclasses. The training, validation, and testing classes contain 60 classes from 12 superclasses, 20 classes from 4 superclasses, and 20 classes from 4 superclasses, respectively. The target is to minimize the information overlap between classes to make it more challenging than current few-shot classification tasks. Same as CIFAR-FS, there are 600 nature color images of size 32 × 32 in each class.

Results. In Table 1, we compare our results with the previous studies, and the table shows that the highest accuracies of our experiments exceeded the current state-of-art accuracies from 3% to 5%. Besides, Table 2 and Table 3 summarize the results on the CIFAR-FS and FC100 5-way tasks, and in most cases our method rises accuracy by 0.5%-3%.

Table 4: The average accuracies (%) with 95% confidence intervals on miniImageNet. *Result from Lee et al. (2019). The best results are highlighted. Here only list the best results of previous works due to the shortage of space.

Method	**1-shot**	**5-shot**
Gidaris & Komodakis (2018)	56.20±0.86	73.00±0.64
TADAM Oreshkin et al. (2018)	58.50±0.30	76.70±0.30
LEO Rusu et al. (2018)	61.76±0.08	77.59±0.12
ProtoNets* Snell et al. (2017)	59.25±0.64	75.60±0.48
M-SVM Lee et al. (2019)	64.09±0.62	80.00±0.45
M-SVM (+ens+val) (our)	65.38±0.45	**82.13±0.31**
R2-D2 (+ens+val) (our)	**65.95±0.45**	81.96±0.32

Table 5: Comparison to the average accuracies (%) with 95% confidence intervals between the methods with and without Task Aug on miniImageNet 5-way. The results are better than its compared results are highlighted.

Method	**1-shot**		**5-shot**	
	Baseline	Task Aug	Baseline	Task Aug
ProtoNets Snell et al. (2017)	58.67±0.48	**60.52±0.48**	75.24±0.37	**77.00±0.36**
ProtoNets (+ens)	62.12±0.48	**63.69±0.47**	78.11±0.34	**79.77±0.34**
ProtoNets (+val)	60.13±0.48	**62.22±0.49**	76.98±0.36	**77.59±0.37**
ProtoNets (+ens+val)	63.84±0.48	**65.04±0.48**	79.54±0.35	**80.60±0.34**
M-SVM Lee et al. (2019)	60.02±0.45	**62.12±0.44**	77.85±0.34	**78.90±0.34**
M-SVM (+ens)	63.44±0.45	**64.56±0.44**	80.18±0.32	**81.35±0.32**
M-SVM (+val)	61.58±0.45	**63.14±0.45**	78.65±0.34	**79.97±0.33**
M-SVM (+ens+val)	64.74±0.45	**65.38±0.45**	81.39±0.32	**82.13±0.31**
R2-D2 Bertinetto et al. (2018)	60.57±0.44	**62.32±0.45**	77.44±0.34	**78.81±0.34**
R2-D2 (+ens)	63.72±0.44	**64.79±0.45**	79.90±0.33	**81.08±0.32**
R2-D2 (+val)	**62.82±0.45**	62.64±0.44	78.61±0.35	**79.58±0.33**
R2-D2 (+ens+val)	65.50±0.45	**65.95±0.45**	81.34±0.32	**81.96±0.32**

We can observe that: some results without the ensemble approach Huang et al. (2017) of baseline and Task Aug are close, but the advantage of Task Aug is still obvious on the results with the ensemble approach. We suspect that the scale of backbone limits the performance of the best model. A larger scale backbone is needed for the training process with Task Aug. For the results of ensemble approach, since Task Aug reduces the over-fitting, more models during the training process have good performance, which provide ensemble with models of higher quality.

4.3 Evaluation on miniImageNet

The **miniImageNet** Vinyals et al. (2016) is one of the most popular benchmark for few-shot classification, which contains 100 classes randomly selected from ILSVRC-2012 Russakovsky et al. (2015). The classes are randomly divided into training classes, validation classes and test classes, and them contain 64, 16 and 20 classes, respectively. There are 600 nature color images of size 84×84 in each class. Since Vinyals et al. (2016) did not release the class splits, we use the more common split proposed by Ravi & Larochelle (2017).

Results. Table 4 shows that the highest accuracies of our experiments exceeded the current state-of-art accuracies from 1% to 2%. Table 5 summarizes the results on the miniImageNet 5-way tasks, and in most cases our method rises accuracy by 0.5%-2%.

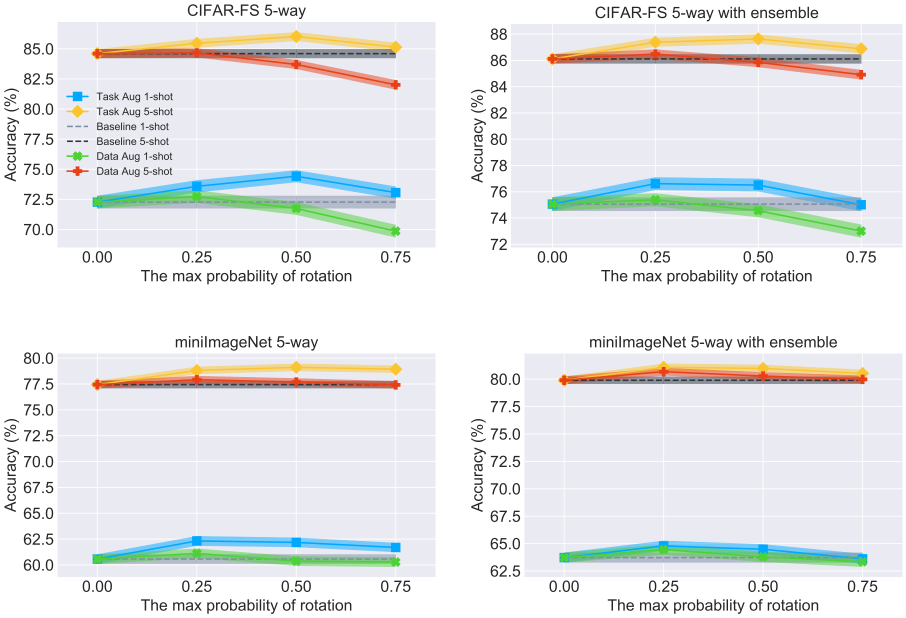

Figure 3: The accuracies (%) on meta-test sets with varying probability p_{max} for the novel classes.The 95% confidence interval is denoted by the shaded region. In general, the performance of Task Aug on most of the regimes is better than Data Aug and baseline.

4.4 Efficiency of the probability for Task Aug

To identify whether the rotation multi 90 degrees for Task Aug is better than that for Data Aug, we analyzed the experiment on CIFAR-FS and miniImageNet. The linear rising of p was also used for Data Aug, and $\mathrm{T} = 80000$ for both Task Aug and Data Aug. In the analysis, the training classes set was not augmented by the validation classes set. As shown in Figure 3, we observed that: with p_{max}, the accuracy rises at first, reaches the peaks between 0.25 and 0.5, then declines and reaches baseline when $p_{max} = 0.75$ at the end, which is the proportion of the novel classes in all classes. On the other hand, the rotation multi 90 degrees for Data Aug can not improve or even cause worse performance.

5 Conclusion

We proposed a Task Level Data Augmentation (Task Aug), a data augmentation technique that increased the number of training classes to provide more diverse few-show task instances for meta-learning. We proved that Task Aug was valid for CIFAR-FS, FC100, and miniImageNet, and exceeded the result of the previous works. Task Aug achieved the performance by rotating the images 90, 180 and 270 degrees. This method is simple and cost-effective. With the ensemble method, we exceeded the state-of-the-art result over a large margin.

Future work will focus on searching different network structures for meta-learning, since the training with Task Aug would require more large model. Besides, we will try to apply Task Aug to other few-shot learning tasks to verify its effectiveness. Another interesting topic is to build other approaches for Task Aug, such as swapping channel order, picture blend or even auto augmentation.

References

Maruan Al-Shedivat, Trapit Bansal, Yuri Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. *arXiv preprint arXiv:1710.03641*, 2017.

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 10727–10737, 2018.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375, 2018.

Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3018–3027, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 8, 2010.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, 2019.

Jinchao Liu, Stuart J Gibson, and Margarita Osadchy. Learning to support: Exploiting structure information in support sets for one-shot learning. *arXiv preprint arXiv:1808.07270*, 2018.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

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*, pp. 721–731, 2018.

Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7229–7238, 2018.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2017.

Danilo Jimenez Rezende, Shakir Mohamed, Ivo Danihelka, Karol Gregor, and Daan Wierstra. One-shot generalization in deep generative models. *arXiv preprint arXiv:1603.05106*, 2016.

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*, 115(3):211–252, 2015.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pp. 1842–1850, 2016.

Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogerio Feris, Raja Giryes, and Alex Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *Advances in Neural Information Processing Systems*, pp. 2845–2855, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 403–412, 2019.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pp. 1139–1147, 2013.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pp. 3630–3638, 2016.

Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 2365–2374, 2018.