FSIL: Few-shot and Incremental Learning for Image Classification

Shivam Saboo,[1] Anbumani Subramanian[2]
[1]Indian Institute of Information Technology [2]Intel Corporation
ssaboo.cs@gmail.com
anbumani.subramanian@intel.com

Abstract

The success of deep learning can be largely attributed to availability of large datasets. However creating large annotated datasets is often expensive and sometimes even impossible. Also, in many applications the entire dataset might not be available at once, and new data may be observed sequentially. Few-shot learning addresses the issue of requirement of large datasets by learning and generalizing to new classes given only a few annotated examples, and incremental learning aims to learn novel class categories in a sequential manner without forgetting the old classes. In this paper, we propose a novel method which combines both - using a few-shot adaptation stage that learns to assign importance to each feature, and embed incremental learning in this few-shot setup. We demonstrate that simply adding new classes to few-shot framework is sub-optimal due to interference between the prototypes of the old and new classes. To this end, we propose a conceptually simple divergence penalty based on cosine similarity that prevents such interference. Further, our method is memoryless as it does not require storing data from the old classes. We conduct the experiments on CUB dataset and also in cross-domain setting using *mini*ImageNet and India Driving Dataset to demonstrate the efficacy of our method. We achieve nearly 7% improvement in 5-shot and nearly 4% improvement in 10-shot cross domain incremental setup, compared to previous baselines. Our method can also complement several other few-shot learning frameworks.

Introduction

Current progress in deep learning has led to breakthroughs in computer vision for various challenging tasks such as image classification (He, Zhang, et al. 2016), object detection (Wu et al. 2019; Redmon and Farhadi 2018), segmentation (He, Gkioxari, et al. 2017), etc. However the current supervised learning based techniques have several demerits. Firstly, massive datasets are often required to achieve desired results. This is often impractical in real world scenarios as obtaining large amounts of annotated datasets is expensive, or just not possible. In such scenarios, the need for algorithms that can learn from a limited number of examples – as low as 5-10 (few-shot) – becomes inevitable. And secondly, conventional learning techniques are incapable of learning in an incremental manner. This means that to accommodate new knowledge into a neural network, it needs to be trained from scratch by combining old and new data. Naturally, this becomes infeasible because of the need to store all the old data and retraining every time new data is added. To this end, we propose a technique which enables incremental learning in few-shot setup. Below, we briefly introduce few-shot and incremental learning.

Few-shot learning setup aims to recognize novel classes using just a few (typically 1-10) annotated examples. Typically few-shot learning algorithms consists of two training stages. First stage is the base training stage where sufficient number of examples for each class is available. In the second stage however, novel classes are encountered with only a few examples for each class. Broadly few-shot learning can be categorized into initialization based, and metric-learning based techniques (Chen et al. 2019). Initialization based methods aim to learn good initialization such that novel classes can be quickly learnt using few-shot data. Metric learning based methods (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017) build a comparison classifier based on distance metrics such as cosine distance, Euclidean distance, etc, and aim to learn representation for each class such that inter-class variance is high.

Incremental learning (or lifelong learning) is a learning paradigm which allows learning of new tasks without forgetting the previously learnt tasks and without needing the data from the previous tasks. Transfer learning (Tan et al. 2018) based methods, that finetune a pretrained model on a new task, usually exhibit forgetting of the previously learnt task. Therefore, the need for algorithms that can learn incrementally becomes inevitable in many real world scenarios. Broadly incremental learning methods are based on regularization (Kirkpatrick et al. 2017), distillation (Shmelkov, Schmid, and Alahari 2017) and memory (Ye et al. 2018). Regularization based methods prevent the update of weights that were important for the previous task. Distillation aims at retaining the knowledge from old model by constraining the logits of new model to be close to the logits of the old model. Memory based methods store a few example from previous tasks and use it when learning new tasks to avoid forgetting.

In this paper, we propose and demonstrate a novel setup that combines few-shot and incremental learning. Such a setup is appealing in the real world scenarios where a model needs to continually learn new classes but only a few examples for each class are available. To this end, following are the key contributions in our paper:

- we introduce a novel few-shot adaptation stage which learns to assign importance to each feature in the prototype, using the few-shot dataset
- we demonstrate that simply adding new classes in a few-shot learning framework is sub-optimal and causes interference between the old and new class prototypes, and
- we propose a simple cosine similarity based divergence penalty that reduces interference between the prototypes of the old classes and new classes when learning incrementally.

Related Work

In this section, we provide a brief overview of the previous work done in: i) few-shot learning and ii) incremental learning.

Few-shot learning

Few-shot learning has been a widely studied topic in machine learning research (Wang and Yao 2019; Snell, Swersky, and Zemel 2017; Vinyals et al. 2016) and they broadly fall under meta learning and metric learning based approaches. Meta-Learning (Finn, Abbeel, and Levine 2017; Ravi and Larochelle 2016) by design focuses on learning initialization or optimization rule which can help adapt to new task quickly. Model Agnostic Meta Learning (MAML) (Finn, Abbeel, and Levine 2017) learns good initialization such that few iterations of stochastic gradient descent (SGD) is sufficient to adapt to new classes. A first order approximation to MAML was proposed by Nichol (Nichol, Achiam, and Schulman 2018), to make it computationally less expensive. Ravi (Ravi and Larochelle 2016) use an LSTM based optimizer has been proposed, which is learnt to enable quick adaptation in few steps.

Metric Learning based approaches are relatively simpler yet achieve competitive results with meta learning based methods. The principle behind metric learning is to learn good feature representations - ideally having low intra-class and high inter-class variance. Various metrics have been used - Snell (Snell, Swersky, and Zemel 2017) is based on Euclidean distance while Chen (Chen et al. 2019) uses cosine distance based classifiers. Self supervision has shown to improve the results of few shot learning (Gidaris, Bursuc, et al. 2019). RepMet (Karlinsky et al. 2019) propose an end to end method for learning multi-modal distribution over the training categories.

Incremental Learning

Incremental Learning is a setup in which new tasks arrive sequentially and the learning algorithm learns new tasks without forgetting the previous ones. Regularization based methods (Kirkpatrick et al. 2017; Aljundi et al. 2018) prevent the update of weights that are important for the previous tasks. Memory based methods (Hsu et al. 2018) store and replay the previous examples. Some memory based methods (Lopez-Paz and Ranzato 2017; Chaudhry et al. 2018) propose to prevent conflicting gradients between the new tasks and the old tasks by storing examples from the previous tasks. Javed (Javed and White 2019) use gradient based meta-learning and frame the problem of continual learning as inner optimization problem. Further, Hsu (Hsu et al. 2018) categorize incremental learning into three categories, i) incremental domain learning, ii) incremental task learning and iii) incremental class learning.

Incremental domain learning aims to perform domain adaptation in an incremental manner - i.e., performance on the old *domain* must not deteriorate when learning a new domain.

Incremental task learning (Kirkpatrick et al. 2017; Aljundi et al. 2018) operates in multi-head setup meaning, there is a separate classification layer for each task and a task descriptor is available during test time that can be used to select the appropriate layer.

Incremental class learning is the most challenging among the three categories as it operates in single head mode and no task descriptor is available during the test time. Distillation based (Li and Hoiem 2017; Shmelkov, Schmid, and Alahari 2017) and memory based (Hsu et al. 2018) approaches perform better for incremental class learning than regularization (Kirkpatrick et al. 2017; Aljundi et al. 2018) based methods.

All the above methods however either focus on few-shot learning or incremental learning. Gidaris (Gidaris and Komodakis 2018) proposed method for few-shot learning without forgetting the base classes. They propose a meta learning based framework that can generate set of weights for the novel classes. This is learned through episodic training which resembles the process of incrementally learning novel classes. Xiang et al. 2019 builld on top of this and use a multi layered perceptron as their meta network which provides more discriminative trasformations and better performance. The evaluation is done in a multi label classification setup for pedestrain attribute recognition. RepMet (Karlinsky et al. 2019) proposed a state of the art few-shot classifier, however owing to its end to end training, it is not well suited for incrementally learning new classes. Recently, Attention Attractor Network (Ren et al. 2019) proposed a meta learning based approach to perform few-shot and incremental learning. This method also uses episodic training during the meta learning stage to enforce quickly learning novel classes and remembering the base classes. More recently neural gas network (Tao et al. 2020) has been proposed to perform few-shot and class incremental learning which can learn and preserve the topology of manifold formed by different classes. Another similar direction to incremental few-shot learning is online continual learning (M. Caccia et al. 2020) that enable fast

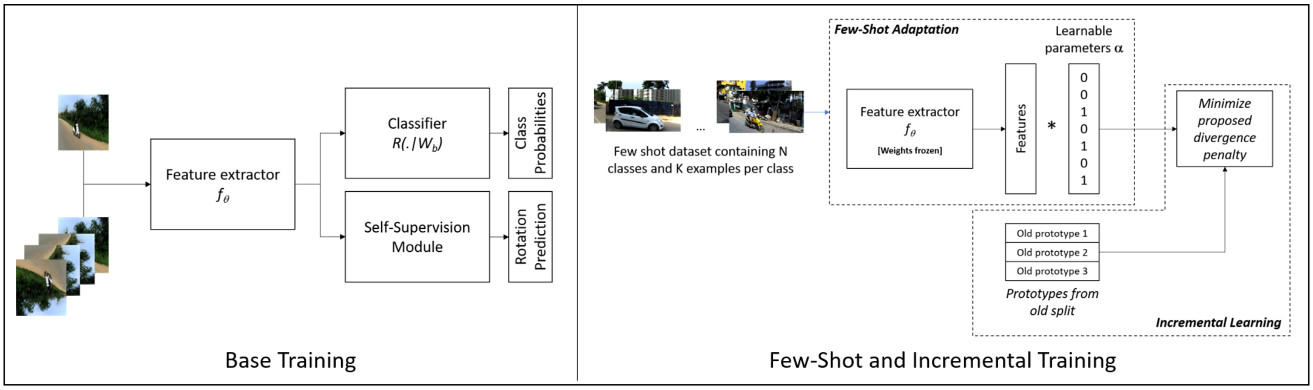

Figure 1: Base Training: We use cosine distance classifier $R(.|W_b)$ and self-supervision module that predicts angle of rotation of input image. Few-shot adaptation: In this stage we learn novel classes in few-shot setup using our proposed Single Model Enhanced (SME). The adaptation stage learns the feature importance vector α using SGD. Incremental Learning: Novel classes are learnt incrementally in few-shot setup using our proposed divergence penalty which reduce interference between the prototypes of the old and new classes

adaptation to new tasks. The setup differs from the few-shot incremental learning setup because the goal is to learn an agent that can quickly adapt to new tasks continually and perform well on that task while retaining the capability to switch to older or different tasks quickly.

Our Method

In our method, few-shot training occurs in two stages. For the first stage (base-training), consider an annotated dataset D_b containing m classes, $C_b = \{c_{b1}, c_{b2}, ..., c_{bm}\}$. The second stage is the few-shot adaptation stage over the dataset D_f containing n classes, $C_f = \{c_{f1}, c_{f2}, ..., c_{fn}\}$. D_f is a k-shot dataset which means that $\forall c_{fi} \in C_f$, there are k-labeled datapoints $\{(x_i, y_i)\}_{i=\{1...k\}}$. Following the conventional few-shot learning setup (Vinyals et al. 2016), the class sets from both the training stages are non-overlapping, that is $C_b \cap C_f = \phi$. For the incremental learning setup, we have a set of such k-shot datasets $T = \{D_{f1}, D_{f2}, ..., D_{ft}\}$. The feature extractor parameterized by weights θ is denoted by f_θ. We denote the classifier by $R(.|W_b)$, parameterized by weight matrix $W_b \in^{d\times n}$. Note that the classifier $R(.|W_b)$ is used only once for the base training and is later discarded. Fig 1 shows high level overview of our proposed method.

Base Training

Most metric-learning and meta-learning based methods (Snell, Swersky, and Zemel 2017; Vinyals et al. 2016; Finn, Abbeel, and Levine 2017), conduct base training by simulating few shot scenario (by creating K-shot N-way tasks). However unlike these, we adhere to conventional training setup for the entire dataset D_b. This is motivated by the results demonstrated by Chen (Chen et al. 2019) that class prototypes obtained from feature extractor trained using cross entropy loss on D_b gives competitive results compared to various meta-learning and metric learning based methods. To reduce the intra-class variance, we make the classifier $R(.|W_b)$ as cosine distance based classifier (Chen et al. 2019; Gidaris and Komodakis 2018; Mensink et al. 2012). Cosine distance based classifier operate by computing cosine similarity between the features extracted from f_θ and weight vector $w_j \in W_b$ corresponding to each class as shown in equation 1

$$s_j = \frac{f_\theta(x)^T w_j}{\|f_\theta(x)\| \|w_j\|} \tag{1}$$

where s_j is similarity score for j^{th} class. The scores vector s is then normalized and converted to probability scores using softmax operator.

To make the learnt representations more generalizable and robust, following (Gidaris, Bursuc, et al. 2019), we incorporate self-supervision as auxiliary task during base training. Specifically, we use rotation prediction (Gidaris, Singh, and Komodakis 2018) for self-supervision. Thus the overall loss function for base training is,

$$L = L_{CE}(\sigma(s), y) + \lambda \Sigma_{y\in\{0,90,180,270\}} L_{CE}(t_{\hat{y}}(x), \hat{y}) \tag{2}$$

where $t_{\hat{y}}$ is transformation function that rotates input x with $\hat{y}$ degrees and L_{CE} is the cross entropy loss. λ controls strength of the self supervision loss. For base training we minimize loss L using SGD.

Few-Shot Adaptation

In this section, we describe methods that we explore for few-shot adaptation given a model trained on base classes. We explore three baselines – Single Model (SM) (Chen et al. 2019) and Ensemble Model (Dvornik, Schmid, and Mairal 2019), and propose a modification to the SM, which we call Single Model Enhanced (SME).

Single Model (SM) In this method, following the baseline proposed by Chen (Chen et al. 2019), we take the feature extractor f_θ trained on the base classes and use the few-shot dataset D_f to compute feature embeddings for each image of a given class. The class prototype c_j is then simply the mean of all such embeddings for that class,

$$c_j = \frac{1}{K}\Sigma_{k=1}^{K} f_\theta(x_k) \tag{3}$$

During the test time, the new input is classified into the category corresponding to nearest prototype.

Ensemble Model This method proposed by Nikita (Dvornik, Schmid, and Mairal 2019) uses an ensemble of multiple CNNs in a joint training. This helps to reduce the variance typically observed in the distance based classifiers. The class prototype is computed by averaging the prototypes from each model in the ensemble.

Single Model Enhanced (SME) In this method we modify the SM method to learn importance weights for each feature during the few-shot adaptation stage. The importance weights, denoted by α are used to scale the class prototypes element-wise,

$$\hat{c_j} = \sigma(\alpha) \circ c_j \tag{4}$$

where σ is the sigmoid operator. The importance weight vector α, is updated by maximizing the likelihood of the following probabilistic model over the few shot dataset D_f,

$$p_\alpha(y = k|x) = \frac{exp(-d(\sigma(\alpha) \circ f_\theta(x), c_k))}{\Sigma_{k'} exp(-d(\sigma(\alpha) \circ f_\theta(x), c_{k'}))} \tag{5}$$

where $d(.,.)$ denotes distance function. The above model is optimized using gradient descent over parameters α. The above model looks similar to the models proposed in (Dvornik, Schmid, and Mairal 2019; Dvornik, Schmid, and Mairal 2020), however there is a subtle difference. In (Dvornik, Schmid, and Mairal 2019), the weights α are applied as scalar multiplication to the prototypes from all examples in the few shot dataset whereas (Dvornik, Schmid, and Mairal 2020) apply such scalar transformation to the universal representations from multiple domains. On the other hand, we apply element-wise scaling at an individual feature level. Intuitively, we want to learn an importance score for each feature which is analogous to gating or attention mechanism. Not all features will be important while adapting from base classes to the novel classes. In fact, as the base training happens over a large number of classes and examples, the representations are bound to have a lot of noise. The importance parameters helps in "switching off" the features of lower importance and learning simpler and more robust representations. Additionally, this also increases the capacity to learn more number of classes incrementally as the representation of each class becomes more sparse.

Extension to Incremental Learning

In this section, we describe how we extend the methods proposed above for incremental learning in few-shot setup. In the incremental learning setup, we observe few shot datasets $D_{fi} \in T$, sequentially. When the dataset D_{fi} arrives, we do not have access to datasets $D_{fj} \forall j < i$. However we assume that the model trained on base classes and the learnt prototypes for all classes from previous few shot datasets $D_{fj} \forall j < i$ are available.

Few-shot learning approaches usually do not exhibit catastrophic forgetting when learning incrementally due to less or no parameter update during few-shot adaptation stage. However, there is a drop in performance for the previously learnt classes due to interference from the new class prototypes. This implies that simply adding new classes without any constraint is sub-optimal and we need a better approach which can reduce the interference among the old and the new class prototypes.

To this end, we propose a simple divergence penalty when learning new prototypes that encourage them to be *different* from the old class prototypes. Specifically we introduce cosine similarity as a divergence penalty while performing the few shot adaptation, with

$$L_{div} = \Sigma_{p_n \in D_{fi}} \Sigma_{p_o \in \{D_{fj}\}_{j<i}} \frac{p_n^T p_o}{\|p_n\| \|p_o\|} \tag{6}$$

where p_n and p_o are the prototypes for classes from new and old datasets respectively. The divergence penalty and the probabilistic model defined in the above section are jointly optimized for when learning new set of classes.

Experimental Details

We conduct experiments to demonstrate two different scenarios: 1) same domain and 2) cross domain. For the same domain, we conduct experiments on Caltech UCSD Birds (CUB) (Welinder et al. 2010) dataset. For the cross domain, we conduct experiments using *mini*ImageNet (Russakovsky et al. 2015) and the India Driving Dataset (IDD) (Varma et al. 2019). The CUB dataset consists of images of objects in same domain - 200 bird species, whereas *mini*ImageNet contains natural images of various objects, and the IDD dataset contains images of Indian roads with objects of various categories.

Same Domain

Experiments are conducted on CUB (Welinder et al. 2010) dataset, which has 11,788 images and 200 classes. The train, validation and test set contains 100, 50 and 50 classes respectively (Ye et al. 2018; Dvornik, Schmid, and Mairal 2019).

The base training is done on the 100 classes from training set and we demonstrate the few-shot results on the 50 classes from the test set.

Cross-Domain

In this setup, we conduct base training on the training classset of *mini*ImageNet (Russakovsky et al. 2015). To demonstrate the efficacy of our proposed method in cross-domain setup, we randomly select 64 classes from *mini*ImageNet that do not overlap with the IDD (Varma et al. 2019) classes and conduct few-shot evaluation on the IDD dataset.

We create IDD Classification dataset by cropping objects whose size is greater than 224 × 224. After this preprocessing, we obtain 9 classes – car, bus, autorickshaw, truck, motorcycle, rider, person, bicycle and traffic sign. For incremental learning, we create 5-4 splits from the 9 classes.

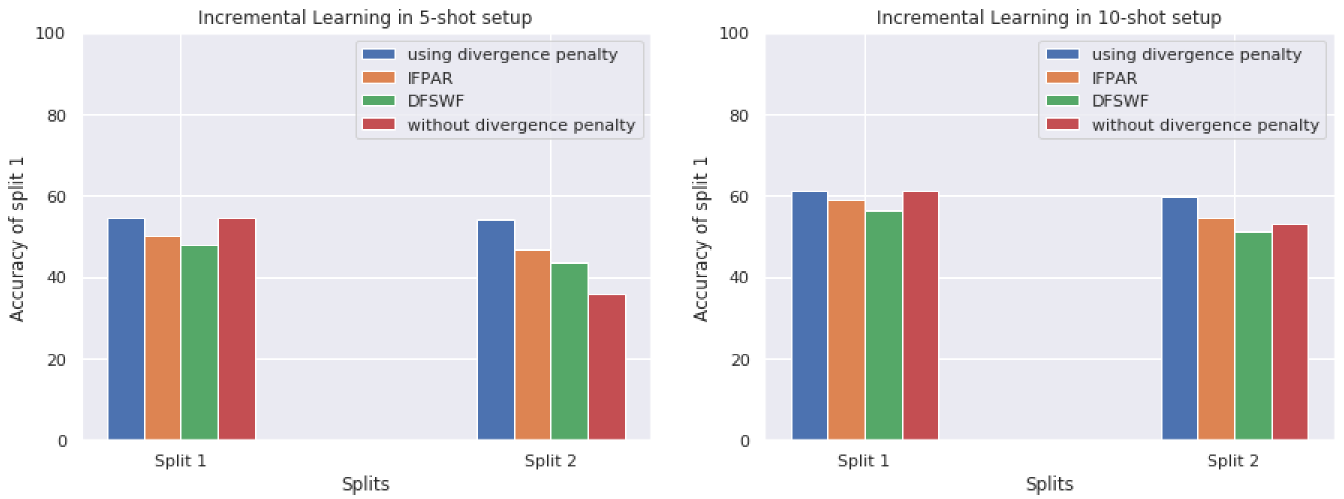

Figure 2: Demonstration of effect of our proposed divergence penalty on the accuracy of the first split as new split is added in i) 5-shot incremental learning setup and ii) 10-shot incremental learning setup. In the 5-shot setup the divergence penalty improves the performance on the first split by 18% and 6.5% in the 10-shot setup compared to when divergence penalty is not used. Comparison with IFPAR (Xiang et al. 2019) and DFSWF (Gidaris and Komodakis 2018) is also shown.

Method	**5-shot**	**10-shot**
MatchingNet	72.86	80.06
MAML	72.09	79.60
ProtoNet	70.77	78.14
RelationNet	76.11	81.17
Single Model (SM)	79.61	83.82
SM + Cosine	78.89	84.08
SM + Cosine + Self-supervision	81.33	84.27
Ensemble	**84.20**	86.14
SME + Cosine + Self-supervision	83.73	**86.91**

Table 1: Evaluation of different few-shot learning methods under same domain on CUB dataset in 5-shot and 10-shot setups. For each task, the best performing method is highlighted.

Experiments are conducted in 5-way and 5-shot and 10-shot for few-shot learning. For incremental learning, we report results using full-way evaluation – i.e., when learning from dataset D_{fi} all classes combined from all splits $D_{fj} \in T \forall j \leq i$ will be considered.

Results

We categorize our findings into three parts i) few-shot in same domain, ii) few-shot in cross domain and iii) few-shot and incremental learning.

Few-shot in same domain

In the same domain setup, the models are evaluated on the same dataset that is used for training. In our experiments, we demonstrate the results on CUB dataset. We use splits of 100, 50 and 50 classes for training, validation and testing according to Ye (Ye et al. 2018).

Table 1 summarizes the performance of different methods in the same domain. Note that we have evaluated the method in both 5-shot and 10-shot setup. The vanilla SM method (Chen et al. 2019) outperforms several baseline methods namely MatchingNet (Vinyals et al. 2016), MAML (Finn, Abbeel, and Levine 2017), ProtoNet (Snell, Swersky, and Zemel 2017) and RelationNet (Sung et al. 2018). Hence for the further experiments on cross domain setup and few-shot and incremental learning setup we consider the SM method as our baseline and continue to build on top of it.

First, we use a cosine distance based classifier instead of the conventional dot product based classifier because it has empirically shown to decrease the intraclass variance. In our experiments we gained slight improvements over the SM baseline in the 10-shot setup.

Then we incorporated self supervision into the base training to learn more robust features. By predicting the angle of rotation the the image, the neural network is forced to learn representation that robustly captures the semantic meaning of the object in the image. Hence, this improves the performance of the network in both 5-shot and 10-shot setup. We also evaluate the ensemble method (Dvornik, Schmid, and Mairal 2019). We set the number of models in the ensemble to 10. The ensemble method outperforms our baselines in the 5 shot setup by a small margin and is competitive with our best baseline in the 10-shot setup. We trained the ensemble with the diversity penalty proposed by Nikita (Dvornik, Schmid, and Mairal 2019).

Finally we introduce the learnable importance parameters in the SM enhanced which intuitively learns the importance of each feature element while optimizing over the prototypi-

Method	5-shot	10-shot
SM	51.05	57.15
SM + Cosine	58.04	64.09
SM + Cosine + Self-supervision	59.09	**66.31**
Ensemble	55.07	60.18
SME + Cosine + Self-supervision	**60.97**	66.06

Table 2: Evaluation of different few-shot learning methods under cross domain setup where the training occurs on the miniImageNet dataset and the testing is done on the IDD dataset. Experiments are done in both 5-shot and 10-shot setup. For each task the best performing method is highlighted.

cal loss as defined in equation 5. This achieves competitive performance with the ensemble method and outperforms all the other baselines with an added advantage of being simple to implement and requiring much less computation for both training and inference.

Few-shot in cross-domain

In cross domain setup, the models are evaluated on different dataset than that was used for training. In our experiments we conduct the training on the miniImageNet dataset and test on the IDD dataset. The cross domain evaluation helps us gauge the generalization efficacy of our method. IDD is representative of challenging driving conditions in India such as the diversity of object shape, color, lighting and viewpoint and hence we propose that generalizing to IDD from the miniImageNet dataset is a good benchmark.

In Table 2 we show the ablative study of our baselines and compare with the ensemble (Dvornik, Schmid, and Mairal 2019) approach (which was the most competitive in the same domain setup). The efficacy of our baselines stands out in the cross domain setup and outperforms the ensemble approach by a significant margin in both 5-shot and 10-shot experiments. We argue that self supervision is the most important aspect of this improvement and enables learning of highly robust features when compared to ensemble method. This suggests that the self supervision task in a single model outweighs the variations incorporated in the ensemble learning that is augmentations, different initialization and dropout, especially for the cross domain setup. Note that even though the performance of the model without importance parameters is slightly better in the 10-shot setup, the importance parameters (single++) outperforms everything in the 5-shot setup. This shows that in the low data regime, our importance parameters help in learning better representations.

Few-shot and incremental learning

Here we discuss the results in few-shot and incremental learning setup. Specifically, we demonstrate incrementally learning the classes of IDD in a few-shot setup and in a cross domain setup (that is the base training is done on the miniImageNet dataset). This is a particularly challenging setup as during the base training not only we do not see the classes of IDD but data domain is also different. In the few-shot and incremental adaptation stage, the model has to learn novel unseen classes with limited data and in an incremental manner meaning not all classes will be available at once and we do not have access to the data of old classes while learning new classes. This requires the model to generate representations that do not interfere with the representations of the old class, otherwise it will cause forgetting of the old classes.

Accuracy (%)	5-shot	10-shot
No incremental	46.94	57.12
First split before	**54.48**	**61.31**
First split after	**54.26**	**59.76**
Combined	**53.70**	**60.80**

Table 3: Incremental on IDD dataset (Varma et al. 2019). i) Without using the divergence penalty and simply adding new classes, ii) Accuracy after training on the first split, iii) Accuracy on the first split after learning the second split iv) Combined Accuracy for both first and second split. Our proposed penalty prevents performance degradation while learning incrementally.

Table 3 shows the result of incremental few-shot learning with and without using our proposed divergence penalty. Our cosine similarity based divergence penalty explicitly forces the representations of new classes to be different from the representations of the old classes, thus reducing the interference and the forgetting. Using our proposed loss, we gain significant improvement of 7% in the 5-shot setup and around 4% in the 10-shot setup (improvements on the combined splits, after learning both splits incrementally). Note that this improvement is achieved on the full-way evaluation, specifically 9-way in case of IDD dataset.

To demonstrate the the forgetting of old classes when learning new classes incrementally, we plot the accuracy over the first split as the new splits are added. Fig. 2 shows the plots for both 5-shot and 10-shot experiments. Using our proposed divergence penalty while learning the new classes significantly reduce the forgetting of the old classes. In the 5-shot setup, the divergence penalty helps in retaining the performance on the first split after adding the second split incrementally and achieves a significant improvement of 18% over not using the divergence penalty. It also gains a significant improvement of 6.5% on the first split in the 10-shot setup. This demonstrates the ability of our proposed loss function in preventing catastrophic forgetting in the few-shot learning scenario. We also compare the performance with DFSWF (Gidaris and Komodakis 2018) and IFPAR (Xiang et al. 2019) which use meta networks trained episodically to enable few-shot learning without forgetting. Both the methods do not exhibit catastropic forgetting on the first split and the performance drop is not significant when new split is added. However, the absolute performance of them drops in the cross domain setup. Recall that we conduct base training on miniImageNet dataset and evaluate the performance on more challenging IDD. We provide two possible explanations for this phenomenon - i) Our method has learnable

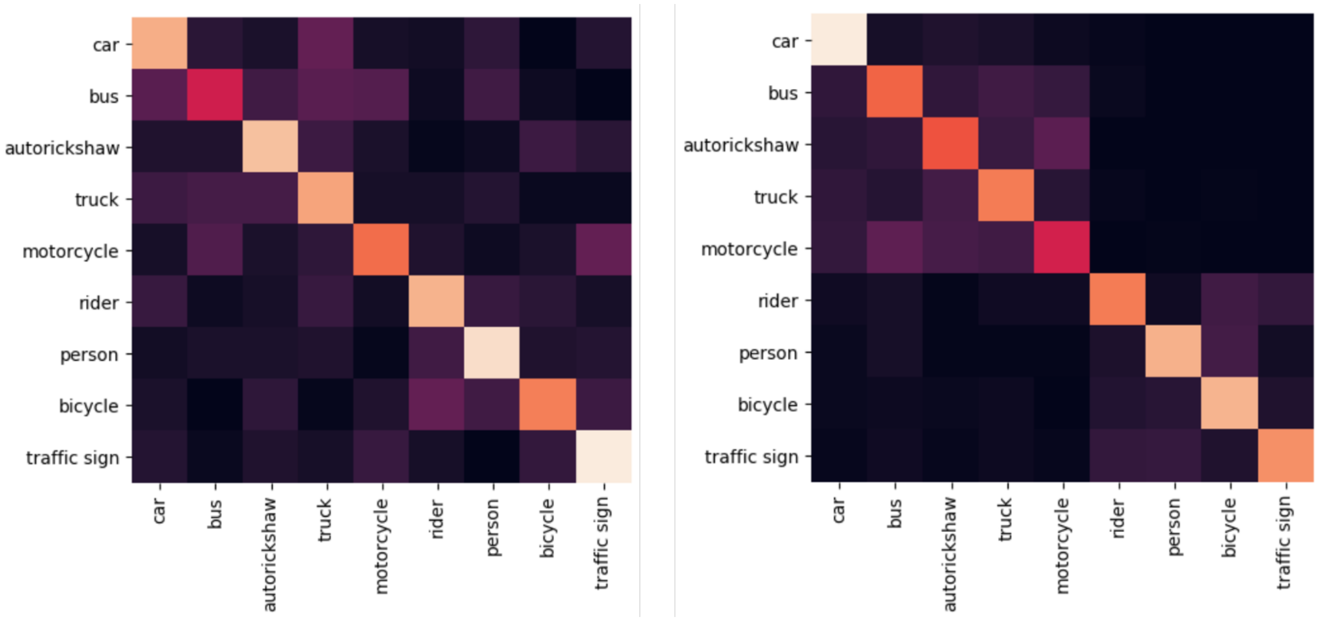

Figure 3: Confusion matrix over all 9 classes in IDD (Varma et al. 2019) classes. The first matrix represents all classes being simply added without any constraint. The second matrix demonstrates the effect of our proposed divergence penalty. Our proposed penalty clearly reduce interference between the two class splits while learning incrementally.

parameters when each new split is encountered while in DFSWF and IFPAR, the networks are fully frozen once meta training has happened, hence there is less degree of freedom for the features to adapt to the new dataset and ii) In both DFSWF and IFPAR, the episodic training is done on the base classes itself to simulate the novel few-shot classes, thus it is prone to overfit to the base dataset. We have left comparison to the neural gas network (NG) (Tao et al. 2020) method as future work as the code hasn't been fully released at the time of submission of this draft [1].

To further highlight the issue of interference among the representations and how our divergence penalty helps in resolving it, we plot two confusion matrices - one when we do not use any constraint and second when we use our divergence penalty during the incremental learning. Fig 3 shows the confusion matrix plot. Note that we split the classes in IDD for incremental learning as described in the experimental section. The plot clearly shows the effect of our divergence penalty as it significantly reduce the interference between the two splits when learnt incrementally, and thus obtain better performance.

Conclusion

In this paper, we explored a combination of few-shot and incremental learning. We proposed a novel few-shot adaptation stage which learns the importance of features using the few-shot data. We also demonstrate that simply adding classes incrementally to few-shot frameworks is sub-optimal due to interference between the old and new class prototypes and to this end, proposed a cosine similarity based distance penalty to reduces the interference. We conducted experiments in both single-domain and more challenging, cross-domain setup. Our method consistently outperforms both single model and ensemble model baselines and also more sophisticated meta learning based methods in the cross domain setup. We achieve nearly 7% improvement in 5-shot and nearly 4% improvement in 10-shot cross domain incremental setup despite simplicity and flexibility of our method.

References

Welinder, Peter et al. (2010). “Caltech-UCSD birds 200”. In:

Mensink, Thomas et al. (2012). “Metric learning for large scale image classification: Generalizing to new classes at near-zero cost”. In: *European Conference on Computer Vision*. Springer, pp. 488–501.

Russakovsky, Olga et al. (2015). “Imagenet large scale visual recognition challenge”. In: *International journal of computer vision* 115.3, pp. 211–252.

He, Kaiming, Xiangyu Zhang, et al. (2016). “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Ravi, Sachin and Hugo Larochelle (2016). “Optimization as a model for few-shot learning”. In:

Vinyals, Oriol et al. (2016). “Matching networks for one shot learning”. In: *Advances in neural information processing systems*, pp. 3630–3638.

Finn, Chelsea, Pieter Abbeel, and Sergey Levine (2017). “Model-agnostic meta-learning for fast adaptation of deep networks”. In: *Proceedings of the 34th International Con-*

[1] https://github.com/xyutao/fscil

ference on Machine Learning-Volume 70. JMLR. org, pp. 1126–1135.

He, Kaiming, Georgia Gkioxari, et al. (2017). “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

Kirkpatrick, James et al. (2017). “Overcoming catastrophic forgetting in neural networks”. In: *Proceedings of the national academy of sciences* 114.13, pp. 3521–3526.

Li, Zhizhong and Derek Hoiem (2017). “Learning without forgetting”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.12, pp. 2935–2947.

Lopez-Paz, David and Marc’Aurelio Ranzato (2017). “Gradient episodic memory for continual learning”. In: *Advances in Neural Information Processing Systems*, pp. 6467–6476.

Shmelkov, Konstantin, Cordelia Schmid, and Karteek Alahari (2017). “Incremental learning of object detectors without catastrophic forgetting”. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3400–3409.

Snell, Jake, Kevin Swersky, and Richard Zemel (2017). “Prototypical networks for few-shot learning”. In: *Advances in neural information processing systems*, pp. 4077–4087.

Aljundi, Rahaf et al. (2018). “Memory aware synapses: Learning what (not) to forget”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154.

Chaudhry, Arslan et al. (2018). “Efficient lifelong learning with a-gem”. In: *arXiv preprint arXiv:1812.00420*.

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

Gidaris, Spyros, Praveer Singh, and Nikos Komodakis (2018). “Unsupervised representation learning by predicting image rotations”. In: *arXiv preprint arXiv:1803.07728*.

Hsu, Yen-Chang et al. (2018). “Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines”. In: *NeurIPS Continual learning Workshop*. URL: https://arxiv.org/abs/1810.12488.

Nichol, Alex, Joshua Achiam, and John Schulman (2018). “On first-order meta-learning algorithms”. In: *arXiv preprint arXiv:1803.02999*.

Redmon, Joseph and Ali Farhadi (2018). “Yolov3: An incremental improvement”. In: *arXiv preprint arXiv:1804.02767*.

Sung, Flood et al. (2018). “Learning to compare: Relation network for few-shot learning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208.

Tan, Chuanqi et al. (2018). “A survey on deep transfer learning”. In: *International conference on artificial neural networks*. Springer, pp. 270–279.

Ye, Han-Jia et al. (2018). “Learning embedding adaptation for few-shot learning”. In: *arXiv preprint arXiv:1812.03664*.

Chen, Wei-Yu et al. (2019). “A closer look at few-shot classification”. In: *arXiv preprint arXiv:1904.04232*.

Dvornik, Nikita, Cordelia Schmid, and Julien Mairal (2019). “Diversity with cooperation: Ensemble methods for few-shot classification”. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3723–3731.

Gidaris, Spyros, Andrei Bursuc, et al. (2019). “Boosting few-shot visual learning with self-supervision”. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8059–8068.

Javed, Khurram and Martha White (2019). “Meta-learning representations for continual learning”. In: *Advances in Neural Information Processing Systems*, pp. 1818–1828.

Karlinsky, Leonid et al. (2019). “Repmet: Representative-based metric learning for classification and few-shot object detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5197–5206.

Ren, Mengye et al. (2019). “Incremental few-shot learning with attention attractor networks”. In: *Advances in Neural Information Processing Systems*, pp. 5275–5285.

Varma, Girish et al. (2019). “IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments”. In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pp. 1743–1751.

Wang, Yaqing and Quanming Yao (2019). “Few-shot learning: A survey”. In: *arXiv preprint arXiv:1904.05046*.

Wu, Yuxin et al. (2019). *Detectron2*. https://github.com/facebookresearch/detectron2.

Xiang, Liuyu et al. (2019). “Incremental few-shot learning for pedestrian attribute recognition”. In: *arXiv preprint arXiv:1906.00330*.

Caccia, Massimo et al. (2020). “Online Fast Adaptation and Knowledge Accumulation: a New Approach to Continual Learning”. In: *arXiv preprint arXiv:2003.05856*.

Dvornik, Nikita, Cordelia Schmid, and Julien Mairal (2020). *Selecting Relevant Features from a Universal Representation for Few-shot Classification*. arXiv: 2003.09338 [cs.CV].

Tao, Xiaoyu et al. (2020). “Few-Shot Class-Incremental Learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12183–12192.