Dynamic Visual Few-Shot Learning Through Parameter Prediction Network

Nikhil Sathya Kumar(✉), Manoj Ravindra Phirke, Anupriya Jayapal, and Vishnu Thangam

Imaging CoE, HCL Technologies Ltd., Bangalore, India
{nikhilsathyak,manoj.p,anupriya.j,vishnut}@hcl.com

Abstract. Though machine learning algorithms have achieved great performance when adequate amounts of labeled data is available, there has been growing interest in reducing the volume of data required. While humans tend to be highly effective in this context, it remains a challenge for machine learning approaches. The goal of our work is to develop a visual learning based few-shot system that achieves good performance on novel few shot classes (with less than 5 samples each for training) and does not degrade the performance on the pre-trained large scale base classes and has a fast inference with little or zero training for adding new classes to the existing model. In this paper, we propose a novel, computationally efficient, yet effective framework called Param-Net, which is a multi-layer transformation function to convert the activations of a particular class to its corresponding parameters. Param-Net is pre-trained on large-scale base classes, and at inference time it adapts to novel few shot classes with just a single forward pass and zero-training, as the network is category-agnostic. Two publicly available datasets: MiniImageNet and Pascal-VOC were used for evaluation and benchmarking. Extensive comparison with related works indicate that, Param-Net outperforms the current state-of-the-art on 1-shot and 5-shot object recognition tasks in terms of accuracy as well as faster convergence (zero training). We also propose to fine-tune Param-Net with base classes as well as few-shot classes to significantly improve the accuracy (by more than 10% over zero-training approach), at the cost of slightly slower convergence (138 s of training on a Tesla K80 GPU for addition of a set of novel classes).

Keywords: Param-Net · MiniImagenet · Pascal-VOC · Activations · Few-shot learning

1 Introduction

Current state of the art on semantic segmentation, object detection, image classification and most other learning based tasks rely on deep neural networks. Deep neural networks are high-capacity powerful models which require large amounts

Supported by HCL Technologies Ltd.

I. S. Kotsireas and P. M. Pardalos (Eds.): LION 14 2020, LNCS 12096, pp. 249–263, 2020.
https://doi.org/10.1007/978-3-030-53552-0_24

of annotated data and millions of parameters. Large amounts of supervised training data per concept is required for deep learning algorithms to achieve great performance, and the learning process could take multiple days to weeks using specialized expensive hardware like GPUs. Adapting a deep learning model to recognize a new class includes 2 major steps: 1) Collection of the large scale dataset, 2) Fine-tune the existing model to recognize the new class. Both of these steps are time, memory and resource intensive. If new classes are to be recognized, then typically thousands of training examples are required for training and fine-tuning the model. Sometimes, unfortunately this fine-tuning might result in the model forgetting the initial classes on which it was trained. One of the most important objectives of few-shot learning based algorithms is to adapt the existing models at real-time, to recognize novel classes which were unseen during the initial training phase. The major challenge is that, these novel classes have less than 5 visual examples each for training the model. The performance of state-of-the-art classification models deteriorates when the number of images per new class reduces to less than 10, whereas humans are capable of learning new visual concepts reliably and effortlessly with very few examples. This has inspired scientists to adapt deep learning algorithms to work on few-shot domain, where the main goal is to learn novel concepts using limited number of examples. The main advantage of solving the few-shot problem is that it relies only on very few examples and eliminates or restricts the need to formulate large amount of labeled training data, which is usually a cumbersome and costly process.

In this paper, we propose a novel, computationally efficient, yet effective framework called Param-Net, which is a fusion of the best practices of parameter generation, gradient descent and data augmentation methods. Two publicly available dataset: MiniImageNet and Pascal-VOC have been used in this paper for evaluation and bench-marking. MiniImageNet dataset is the most popular dataset for benchmarking few-shot learning algorithms. Pascal-VOC is the most popular dataset for object detection tasks. Using Pascal-VOC dataset, an attempt can be made to scale the few-shot classification task to few-shot detection task. The dataset is split into: (1) classes that contain adequate number of samples denoted as C_{Base}, this is considered as large scale dataset and (2) classes that contain 1–5 images each which are denoted as C_{Few}, these are the few shot classes. The goal of our work is to devise a visual learning based few-shot system that achieves good performance on novel few shot classes, C_{Few} (with less than 5 samples each) and does not degrade the performance on the large scale base classes C_{Base} and has a fast inference with little or zero training for adding new classes to the existing model.

In neural networks, parameters of a particular class and its activations share a strong relationship and this property is used by Param-Net to predict weights for novel classes. For fair comparison with state-of-the-art approaches, we use a Res-Net based model for extracting the most relevant features/activations of the input images. The activations which are determined prior to the final fully connected layer in the base model, is used as input to the Param-Net, which is a multi-layer transformation function. Param-Net is used to convert activations

of a particular class to its corresponding weights. Res-Net as well as Param-Net model is pre-trained on C_{Base}. Param-Net can adapt to novel few shot classes with just a forward pass and zero training as the network is category agnostic.

In this paper, the models are initially tested on MiniImageNet dataset and compared with state-of-the-art few shot algorithms. The proposed Param-Net model outperforms the state-of-the-art methods on few-shot classes, while also not compromising on the efficiency on base classes (C_{Base}). On MiniImageNet dataset, Param-Net achieves an accuracy of 62.69% for 5-way 1-shot learning and 86.14% for 5 shot learning. The inference time of the model to add novel few shot classes is also very low (because of zero-training time), it only takes around 23 ms for adding a novel class on a Tesla K80 GPU. Inspite of this state-of-the-art performance it has been observed that the accuracy of Param-Net on closely similar classes is slightly compromised. To account for this, we suggest fine-tuning the Param-Net with base classes as well as few-shot classes. This significantly improves the accuracy at the cost of slightly slower convergence. Fine-tuned Param-Net achieves an accuracy of: a) 94.23% for 5-way and 5 shot learning on MiniImageNet dataset and b) 87.26% on Pascal-VOC dataset for 20 way 5-shot settings. The fine-tunable version of Param-Net takes around 138 s of training on Tesla K80 GPU for the addition of a set of novel classes for MiniImageNet dataset. Hence with little fine-tuning, Param-Net can be used to adapt a deep learning model to add novel classes with just a single training image.

In "Sect. 2", we describe the techniques widely used and documented in literature to achieve current state-of-the-art results, most of the techniques described in this section are used to benchmark the Param-Net framework. Then in "Sect. 3" we elaborate the proposed Param-Net approach, in "Sect. 4", we discuss the experimental setup, results and benchmarks and in "Sect. 5", we discuss the conclusion and future scope.

2 Related Work

The ideas behind Param-Net has broad prior support in literature, but mostly appear in disjoint or in incompatible problem setting. Research literature on few-shot learning techniques exhibits great diversity. We adapt these concepts into a unified framework for recognition in real-world scenarios. In this section, we focus on methods using the supervised meta-learning paradigm [12], [51], [9] most relevant to ours and compared to in the experiments. We can divide these methods into 5 categories:

Data Generation and Data Augmentation Methods: In [9], a sampling method was proposed that extracts varying sequences of patches by decorrelating an image based on maximum entropy reinforcement learning. This is a form of "learned" data augmentation. In [19], GAN based approach was proposed to address the few shot learning, where GAN allows the few shot classifiers to learn sharper decision boundary, which could generalize better. In [30], a modified

auto-encoder was proposed to synthesize new samples for an unseen category just by seeing few examples from it.

Gradient Descent Based Methods: Meta-LSTM [12], treats the model parameters as its hidden states and uses LSTM as a model updater, that not only learns the initial model, but also the update rule. In contrast to Meta-LSTM, MAML [14] only learns an initial update. In MAML, the updating rule is fixed to a stochastic gradient descent (SGD). In [26], a variant of MAML was proposed where only first order gradients were used. In [21], MetaSGD was proposed as an extension of MAML, which learns weight initialization as well as learner's update step size. In [1,2], a modification to MAML was proposed to prevent overfitting. In [2] entropy based and inequality minimization measures were introduced and in [1], Meta-Transfer Learning approach was introduced where it leverages transfer learning and benefits from referencing neuron knowledge in pre-trained deep nets. A framework was proposed in [18] to unify meta learning and gradient based hyperparameter optimization. In [20], Neuron-level adaptation was suggested, to reduce the complexity of weight modification when the connections are dense.

Metric Learning Methods: Siamese Neural Network which uses a two stream convolutional neural network was originally utilized by Koch et al. [30], to learn powerful discriminative representations and then generalized them to unseen classes. Vinyals et al. [13] proposed Matching-Nets and introduced the episodic training mechanism into few-shot learning. Prototypical Network was proposed in [15], which is built upon the matching network [13], uses cosine similarity and 4-layer network. Here, query image is compared with support images using class centroids to eliminate outliers in support set. In [16], a variant of Matching network [13] was proposed and named the Relation-Net. It uses additional network to learn similarity between image through a deep non-linear metric. Relationship of every query-support pair is evaluated using a neural network. As an extension to prototypical network in [15], three light weight and parameter free improvements were proposed in [5]. In [10,25] modifications to Relation-Net was proposed. In [10], images were encoded into feature vectors by an encoding network. In [25], second order descriptors were proposed instead of first order descriptors. Given a new task with its few-shot support set, Garcia et al. [23] proposed to construct a graph where all examples of the support set and a query set are densely connected. There have been modifications proposed to [23], by [27] and [7]. In [27], transductive propagation network was proposed to propagate labels from known labeled instances to unlabeled test instances. In [7], Edge Labeling Graph Neural Network (EGNN) was proposed to predict edge-labels rather than node-labels, this is ideal for performing classification on various tasks without retraining. In [4], local descriptor based image-to-class measure was proposed which was obtained using deep local descriptors of convolutional feature maps.

Parameter Generation Methods: Using attention based mechanism to predict the classification weights of each novel class as a mixture of base classification weights of each novel class, Gidaris et al. in [22] proposed Dynamic-Net to capture class dependencies in the context of few-shot learning. But in Dynamic-Net [22], dependencies were considered between base classes and novel classes. In contrast, the dependencies were considered to exist between all the classes in [6], and these dependencies were proposed to be captured using GNN architectures. But this is computationally more expensive than the simple attention based mechanism proposed in [22]. The episodic formulation proposed in [13], was used by [6], to apply the Denoising Autoencoder framework in the context of few-shot learning, thereby improving the performance of parameter generation, by forcing it to reconstruct more discriminative classification weights. In GNN, input is the labeled training examples and unlabeled test examples of few shot problem and the model is trained to predict the label of test examples. But here, input to GNN is some initial estimate of classification weights of the classes that needs to be learnt and it is trained to reconstruct more discriminative classification weights.

Model Fine-Tuning Methods: Most of the ADAS based models, usually opt for fine-tuning the pre-trained model to add novel classes, but this method works well, only when there is sufficient number of novel class examples for training. The method that has been proposed in this paper is a fusion of the most effective features of Parameter generation, Model fine-tuning, data generation and gradient based methods. In the following section, we shall discuss the architecture of the model followed by the experimental results.

3 Methodology

The datasets are split into large-scale dataset (D_{Base}) and few-shot dataset (D_{Few}), where D_{Base} contains classes which have sufficient number of images for training, whereas D_{Few} contains classes with less than 5 images. C_{Base} refers to the classes present in D_{Base} and C_{Few} refers to the classes present in D_{Few}. There is no overlap between C_{Base} and C_{Few}. The distribution of the dataset into D_{Base} and D_{Few} is illustrated in Table 1.

Table 1. Random distribution of classes from public datasets into large scale and few-shot classes.

Datasets	Number of classes in D_{Base}	Number of classes in D_{Few}
MiniImageNet	80	20
Pascal-VOC	13	7

The goal of our work is to devise a visual learning based few-shot system that achieves good classification performance on novel few shot classes, C_{Few} (with

less than 5 samples each) and does not degrade the performance on the large scale base classes C_{Base} and has a fast inference with little or zero training for adding new classes to the existing model.

In a neural network, for a particular class: weights and activations are closely related. In this paper, we propose a novel, computationally efficient, yet effective framework called Param-Net, which is a multi-layer dense regression transformation function to convert the activations of a particular class to its corresponding weights.

Initially, Resnet-101 deep neural network was considered as the base model and was pre-trained on the large scale dataset (D_{Base}). In the base-model, the entire network prior to the final fully connected layer was considered as feature extractor. The final fully connected layer was considered as the classifier network. For an input image 'X_i', the feature extractor will output a "d"-dimensional feature vector $Z_{X_i} = F(X_i)$. The weights "w" of the classifier network consists of "N" classification weights, where "N" is the number of classes in D_{Base}:

$$w = [w_i]_{i=1}^{N} \tag{1}$$

where w_i is the d-dimensional classification weight vector for the i^{th} class. For input image 'X_i and for 'N' classes, the classifier will compute the classification scores as: For input image 'X_i',

$$[s_{1i}, s_{2i}, ..., s_{Ni}] = [Z_{xi}w_1, Z_{xi}w_2, .., Z_{xi}w_N], \tag{2}$$

For an image "X_e", belonging to class "k", the objective of the feature extractor and classifier is to maximize s_{ke}, where $s_{ke} = Z_{xe}w_k$, and minimize [s_{1e}, $s_{2e}, ..., s_{(k-1)e}$] and [$s_{(k+1)e}$, $s_{(k+2)e}, ..., s_{Ne}$]. The weights "w" are learnt through back-propagation using the loss function:

$$\frac{\sum_{i=0}^{M} L_i}{M} \tag{3}$$

where,

$$L_i = -log\frac{e^{Z_{X_i}W_{y_i}}}{\sum_{j=1}^{N} e^{Z_{X_i}W_j}} \tag{4}$$

where "M" is the number of images per epoch, for image "X_i": "L_i" is the loss and "y_i" is the annotation label. To adapt the base model to include novel few-shot classes, a transformation layer named Param-Net has been proposed in this paper. The objective of the Param-Net is to predict parameters of a particular class based on its corresponding activations. The activations which are used as input to the Param-Net are determined using the feature extractor network. Parameters of the original base-model classifier network is replaced by the Parameters estimated from the Param-Net which can be denoted as:

$$T(Z_{\mathrm{X_i}}) = we_{\mathrm{i}} \tag{5}$$

where T() is the transformation function or the Param-Net, $\mathrm{Z_{X_i}}$ is the activation of the image "$\mathrm{X_i}$" and $\mathrm{we_i}$ is the estimated parameters for the image "$\mathrm{X_i}$".

During the training phase of the Param-Net, initially a mini-batch of input images is formed from the $\mathrm{C_{Base}}$ classes, containing $\mathrm{M}^{'}$ images of each class. Hence the size of the mini-batch was N * $\mathrm{M}^{'}$, where N is the number of classes. Using the feature extractor network, d-dimensional activation vectors are extracted for the entire mini-batch, containing N classes and $\mathrm{M}^{'}$ images for each class. Mean activations are extracted for each class using:

$$Mz_{\mathrm{j}} = \frac{\sum_{k=0}^{M^{'}} Z_{\mathrm{X_{jk}}}}{M'} \tag{6}$$

where $\mathrm{Mz_j}$ is the mean activation of the batch of images belonging to $\mathrm{j^{th}}$ class, $\mathrm{M}^{'}$ is the number of images per class in the batch and $\mathrm{Z_{X_{jk}}}$ is the activation of the $\mathrm{k^{th}}$ image of the $\mathrm{j^{th}}$ class.

The d-dimensional mean activations of each of the class is input to the Param-Net to estimate the parameters of the corresponding classes, as illustrated in Fig. 1(a). The Param-Net is a regression network whose input and output dimensions are of the same size, but to reduce overfit, we have posed it as a classification task. After estimating parameters of all the classes individually, they are concatenated into a "d * N" vector.

A mini-batch of training images $\mathrm{X'_1, X'_2, ..., X'_N}$, containing an equal sample of all the classes is considered and features are extracted, where $\mathrm{X'_1}$, contains a subset of images belonging to class 1. These features are then convolved with the estimated parameters from the transformation layer and softmax activations are applied. The output is compared with the input annotated labels and the loss function is computed and the gradients are propagated. The loss function from Eq. 4 is modified to:

$$L_{\mathrm{i}} = -log \frac{e^{\mathrm{Z_{x_i'}[T(Mz)]_{y_i}}}}{\sum_{j=1}^{N} e^{\mathrm{Z_{x_i'}[T(Mz)]_j}}} \tag{7}$$

where, $\mathrm{L_i}$ is the loss for the image '$\mathrm{X_i}$', 'N' is the number of classes, $\mathrm{Z_{x_i'}}$ is the activations estimated or features extracted for the image '$\mathrm{X_i}$' using the feature extraction network, '$\mathrm{[M_z]_{y_i}}$' is the mean activation of the actual class that the image '$\mathrm{X_i}$' belongs to, '$\mathrm{[Mz]_j}$' is the mean activation of all the other classes ranging from 1 to N and T[] is the estimated weights from the Param-Net. The entire flow of the network has been illustrated in Fig. 1 (b).

There are three distinct phases in this approach: Training, Parameter predictor and inference. During the training phase, only images from $\mathrm{D_{Base}}$ are used

to train the transformation layer/Param-Net. None of the images from the D_{Few} are used. In the parameter predictor phase, the images 1–5 of a given class C_{Few} from D_{Few} are considered and passed through the feature extractor and mean activations of the extracted features are determined.

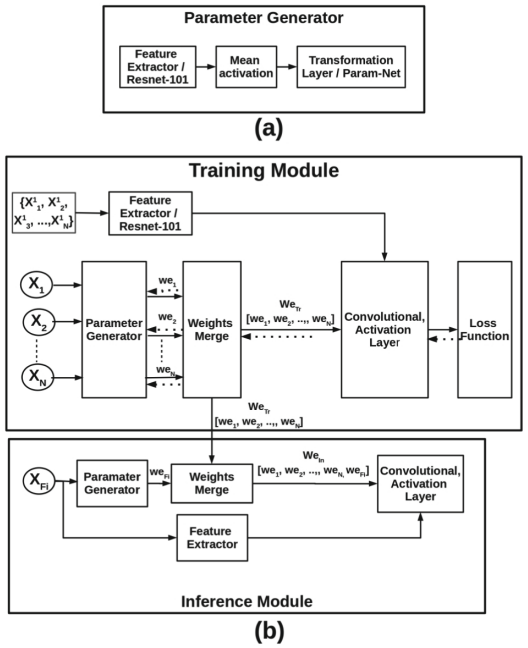

Fig. 1. Illustration of few-shot learning pipeline proposed in this paper using Param-Net based approach with zero-training. (a): Parameter Generator network (b): Training and inference pipeline for the proposed model.

These mean activations are input to the transformation layer to estimate the parameters for the few shot class C_{Few}. These estimated parameters for few-shot classes are concatenated with the estimated parameters of the other base classes. This way, a new class is added to the existing model with zero training. At inference time, for any input image, features are extracted and convolved with the estimated parameters to determine the class of the input image.

We also propose to fine-tune the Param-Net with base classes as well as few-shot classes to significantly improve the accuracy. For addition of a novel class to the existing model, Param-Net is fine-tuned with mean-activations from C_{Base} as well as from C_{Few}.

But the convergence of the fine-tuned model does not require too many epochs nor too many computation cycles because: (1) Param-Net is just a 2-layer dense network, (2) Input to the Param-Net is only mean activations which are of lower dimension, compared to the high-dimension raw input. Hence, with

little training, the efficiency of Param-Net can be considerably increased. The modified flow for the proposed Param-Net is illustrated in Fig. 2.

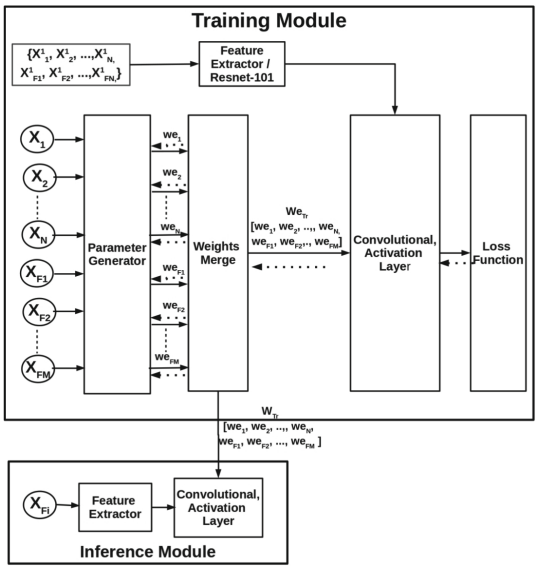

Fig. 2. Illustration of few-shot learning pipeline proposed in this paper using fine-tuning based Param-Net approach with little-training. Training and inference pipeline for the proposed model has been shown with the Param-Net being fine-tuned, every time a novel few-shot class is added to the model.

4 Results

We evaluate the proposed Param-Net extensively in terms of few-shot recognition accuracy and model convergence speed on MiniImageNet and Pascal-VOC datasets. In the following sections, we describe the datasets, detailed experimental setup, comparison to the state-of-the-art methods and an ablation study with respect to different model architectures.

4.1 Dataset

MiniImageNet was proposed by Vinyals et al. in [13], for evaluating few-shot learning based algorithms. Its complexity is high due to the use of images from ImageNet dataset, but requires less resources and cheaper infrastructure than running on the full ImageNet dataset. It is the most popular dataset for benchmarking few-shot learning algorithms. We used the same split proposed by [1] for fair comparison. We consider 5-way classification for both 1-shot and 5-shot.

Pascal-VOC dataset is one of the most popular datasets for object detection task. The logical extension of few-shot learning based algorithms is object detection based few-shot learning. Hence the use of Pascal-VOC helps us to make the model robust to scale to object detection tasks. Param-Net was evaluated on the Pascal VOC for 20-way 1-shot performance.

4.2 Model

For a fair comparison with the state-of-the-art we use pre-trained Resnet-101 as our base model for feature extraction, for MiniImageNet and Pascal-VOC datasets. Resnet-101 was pre-trained on ImageNet and the model was later fine-tuned on D_{Base} dataset. Model fine-tuning was performed by freezing the initial layers and only updating the weights of the final few layers. The layer prior to the final fully connected layer serves as a feature extractor for Param-Net. The dimension of the extracted features is 2048. The learning rate for fine-tuning Resnet-101 was 0.0001 using an Adam optimizer.

For MiniImageNet dataset, the D_{Base} dataset had 80 classes with 600 images each and these images were used for fine-tuning Resnet-101. Of the 600 images, 500 images were used for training and 50 images each for validation and testing. The same distribution of D_{Base} is used for training the Param-Net as well. One of the most important characteristics of Param-Net is that it is class-agnostic, which indicates that the network need not be trained on images from D_{Few}. Adam optimizer was used with a learning rate of 0.001 and it took around 187 s for the Param-Net to be trained on a Tesla K80 GPU. Once the Param-Net is trained, it has the capability to add novel classes with zero-training and the addition of novel classes can be done at real time. It takes only 18 ms to add novel classes to the existing base-classes because Param-Net is just a two layer dense regression network with a low-dimensional input.

In the following sections, we shall discuss the key role that a robust feature extractor plays, in improving the performance of few-shot learning algorithms. We shall also discuss the effect of the number of dense layers in the Param-Net on the quality of the weights generated, in the following sections.

Table 2 shows the comparison between: a) Accuracy of conventional fine-tuning based approach, where Resnet-101 model was fine-tuned and test only using the D_{Base} dataset b) Accuracy of the Param-Net based approach on the D_{Base} dataset, where similarly, the Param-Net was trained and tested only using the D_{Base} dataset and the weights of the classification layer of the Res-Net model was replaced by the weights estimated using the Param-Net. It is evident from the results in Table 2, that the Param-Net is able to achieve comparable performance on the D_{Base} dataset, while achieving state-of-the-art results on D_{Few} dataset as depicted in later sections.

Table 2. Comparative study between: a) ResNet-101 based feature extractor and classifier and b) ResNet-101 based feature extractor with Param-Net based classifier. Both are trained and tested only on D_{Base} dataset.

Dataset	ResNet (Feature extractor + classifier) (%)	Resnet feature extractor + Param-Net classifier (%)
Training Data	94.91	92.36
Validation Data	92.63	91.93
Testing Data	93.14	91.06

4.3 Evaluation Results and Comparisons

In Table 3, we compare Param-Net with state-of-the-art methods of different few-shot techniques. The accuracies of all the techniques have been reported on the test dataset.

Data generation techniques [9,11,19,30]: Different techniques like decorrelation [9], GAN [19], auto-encoder [30] and image deformation network [11] have been used. But these models need to be trained on more than just 5 examples for the generators to generate useful data, otherwise the generators under-perform on few-shot data. Hence the accuracy of the data augmentation based methods is lower than the most of the other approaches.

Metric learning techniques [4,5,7,10,13,15,16,23,25]: In [13], few-shot learning problem was addressed using cosine similarities between the features produced by CNN, which is a very simplistic metric to differentiate between the images. In [15] and [16], instead of cosine similarities, a non-linear similarity score was introduced using neural networks, but the descriptors were of first order. As an extension to [15], different approaches [5,10,25] were proposed like second order descriptors, encoding feature vectors using encoder network, batch folding, few shot localization (fsl) and covariance pooling (cp). But still they under-performed because of the inability of the feature extractors to extract meaningful features. GNN based approach, proposed by [22] under-performed because it uses node-labeling to model intra-cluster similarity and inter-cluster dissimilarity. This was addressed in [7] to achieve better performance. In [4], k-nearest neighbors metric was used, but one drawback of this approach was that it used Conv4 as a feature extractor, which is a weak feature extractor. Most of the mentioned metric learning techniques, learn a feature similarity metric between few-shot training examples and a test (query) example. But, these methods treat each training class independently from one another. Hence, the performance of metric learning frameworks is weakened.

Gradient descent techniques: In [12] and [14], element-wise fine tuning was used, hence inducing overfitting on the designed models, and in [12], LSTM was used to update the initial model as well as the update rule, hence it was time consuming as well, this was addressed in [14], by learning only the initial model. As an update to [14], different solutions were proposed like: using first order

gradients [26], joint learning of weight initialization as well as learner's step size [21], entropy based inequality minimization measures [2] and transfer learning and fine-tuning [1]. But the problem with gradient based approaches is that, they require many iterative steps over many examples to perform well.

Parameter generation: In [22], dependencies were considered between novel classes and base classes using simple attention based mechanism. In [6], GNN based techniques were used to differentiate between all the classes, novel and base.

The Param-Net that has been proposed here, is a combination of data augmentation, gradient descent and parameter generation methods. This has resulted in state-of-the-art performance on 1-shot and 5-shot settings for Mini-ImageNet, which is the most popular public dataset for benchmarking few-shot learning algorithms. The existing state-of-the-art methods mainly focus on making relation measure, concept representation and knowledge transfer, but do not pay enough attention to final classification. This issue has been addressed in this paper by posing a regression problem of the Param-Net into a classification task, thereby also reducing the overfit of the model onto the training dataset considerably.

Table 3. 5-way accuracy on MiniImageNet. Blue: Best accuracy.

Method	Algorithm	Models	1-shot (%)	5-shot (%)
Data augmentation	Decorrelation [9]	Conv4	51.03	67.96
	Meta-Gan [19]	Resnet-12	52.71	68.63
	Delta-encoder [30]	VGG-16 (pre)	58.7	73.6
	Image deformation Meta Network [11]	ResNet-18	57.71	74.34
Metric learning	Matching networks [13]	Conv4	43.44	55
	Protonets (PN) [15]	Conv4	49.42	68.2
	RelationNet [16]	Conv4	50.44	65.32
	2nd order similarity network [25]		52.96	68.63
	GNN [23]	Conv-256F	50.33	66.41
	Deep Nearest Neighbor neural network [4]	Conv4	51.24	71.02
	PN+ fsl + CP [5]	Res-Net50		69.45
	Salient-Network [10]	Conv4	57.45	72.01
	Edge-Labeling GNN [7]	Resnet		76.37
Gradient descent	Meta-learning LSTM [12]	Conv-32F	43.56	60
	MAML [14]	Conv-32F	48.7	63
	Reptile [26]	Conv-32F	49.97	65.99
	TAML [2]	Conv-32F	49.4	66.0
	Matasgd [21]	Conv-32F	50.47	64.03
	MTL [1]	Resnet-12	61.2	75.5
Parameter generation	DynamicNet [22]	Conv-4-64	55.45	70.13
	WDAE-ML [6]	WRN-28-10	60.61	76.56
	Our-Param-Net: 2-layer (Resnet)	ResNet-101	**63.31**	**82.29**

Table 4. Results of ablation study

Algorithm	1-shot (%)	5-shot (%)
Resnet-101 + 1 layer Param-Net	61.95	78.95
Resnet-101 + 2 layer Param-Net	63.31	82.29
Nasnet + 2 layer Param-Net	64.69	86.14
Resnet-101 + fine-tune(2 layer Param-Net)	71.18	94.23

Table 3 and Table 4, indicates the importance of a robust feature extractor. In Table 3, the techniques which used Resnet based architecture for feature extraction performed better than the approaches that used Conv-4 or Conv-32 based feature extractors. Similarly, we also experimented using Nas-Net (network based on neural-architecture search) instead of Resnet-101 as feature extractor as shown in Table 4. Nas-Net improved the performance of the algorithm on both 1-shot as well as 5-shot settings. We also conducted experiments to ascertain the contribution of the number of layers in the Param-Net, to the eventual performance of the algorithm on MiniImageNet dataset. It has been observed that a 2-layer dense network performs better than a 1-layer dense network, as has been indicated in Table 4.

In this paper, we also propose an approach where, Param-Net is finetuned with D_{Base} as well as D_{Few}. For every new class that needs to be added, the Param-Net needs to be finetuned. It significantly leads to an increase in the accuracy but with a little extra training time of 138 s for addition of a set of novel classes to the existing model, on a NVIDIA K80 GPU.

5 Conclusion

In this work, we contribute to few-shot learning by developing a novel, computationally efficient framework called Param-Net which achieves top performance in tackling few-shot learning problems.

The main objective of few-shot applications is to add novel classes at real time to the existing model in the presence of less than 5 visual examples. Hence Param-Net has been proposed in this paper. It is a dense transformation layer which converts the activations of a particular class to its corresponding weights. It is pre-trained on large-scale base classes and at inference time it adapts to novel few-shot classes with just a single forward pass and zero or little training as the network is class agnostic.

Extensive comparison with related works indicate that the Param-Net outperforms state-of-the-art algorithms in terms of accuracy (1-shot and 5-shot) and in terms of faster convergence (zero or very-little training). We evaluate the performance of Param-Net on two publicly available datasets: MiniImageNet and Pascal-VOC. MiniImageNet is the most popular dataset for benchmarking few-shot algorithms. Pascal-VOC dataset was used to verify the scalability of Param-Net from few-shot classification task to few-shot detection task.

The future scope of improvement for the proposed Param-Net would be to scale the algorithm to address few-shot detection rather than just few-shot classification problems. The first step to address this challenge has been successfully accomplished by testing the Param-Net on the Pascal-VOC dataset, which is the premier dataset for object detection tasks.

References

1. Sun, Q., Liu, Y., Chua, T.S., Schiele, B.: Meta-transfer learning for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 403–412 (2019)
2. Jamal, M.A., Qi, G.J.: Task agnostic meta-learning for few-shot learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2019)
3. Lifchitz, Y., Avrithis, Y., Picard, S., Bursuc, A.: Dense classification and implanting for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9258–9267 (2019)
4. Li, W., Wang, L., Xu, J., Huo, J., Gao, Y., Luo, J.: Revisiting local descriptor based image-to-class measure for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
5. Wertheimer, D., Hariharan, B.: Few-shot learning with localization in realistic settings. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6558–6567 (2019)
6. Gidaris, S., Komodakis, N.: Generating Classification Weights with GNN Denoising Autoencoders for Few-Shot Learning. arXiv preprint arXiv:1905.01102 (2019)
7. Kim, J., Kim, T., Kim, S., Yoo, C.D.: Edge-labeling graph neural network for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 11–20 (2019)
8. Li, H., Eigen, D., Dodge, S., Zeiler, M., Wang, X.: Finding task-relevant features for few-shot learning by category traversal. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
9. Chu, W.H., Li, Y.J., Chang, J.C., Wang, Y.C.F.: Spot and learn: a maximum-entropy patch sampler for few-shot image classification. In: Proceedings of Conference on Computer Vision and Pattern Recognition (2019)
10. Zhang, H., Zhang, J., Koniusz, P.: Few-shot learning via saliency-guided hallucination of samples. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2770–2779 (2019)
11. Chen, Z., Fu, Y., Wang, Y.X., Ma, L., Liu, W., Hebert, M.: Image deformation meta-networks for one-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
12. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning (2016)
13. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems (2016)
14. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1126–1135, August 2017
15. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, pp. 4077–4087 (2017)
16. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: relation network for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)

17. Mishra, N., Rohaninejad, M., Chen, X., Abbeel, P.: A simple neural attentive meta-learner. arXiv pre-print arXiv:1707.03141 (2017)
18. Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., Pontil, M.: Bilevel programming for hyperparameter optimization and meta-learning. arXiv preprint arXiv:1806.04910 (2018)
19. Zhang, R., Che, T., Ghahramani, Z., Bengio, Y., Song, Y.: MetaGAN: an adversarial approach to few-shot learning. In: Advances in Neural Information Processing Systems, pp. 2365–2374 (2018)
20. Munkhdalai, T., Yuan, X., Mehri, S., Trischler, A.: Rapid adaptation with conditionally shifted neurons. arXiv preprint arXiv:1712.09926 (2017)
21. Li, Z., Zhou, F., Chen, F., Li, H.: Meta-SGD: learning to learn quickly for few-shot learning. arXiv pre-print arXiv:1707.09835 (2017)
22. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4367–4375 (2018)
23. Garcia, V., Bruna, J.: Few-shot learning with graph neural networks. arXiv preprint arXiv:1711.04043 (2017)
24. Cai, Q., Pan, Y., Yao, T., Yan, C., Mei, T.: Memory matching networks for one-shot image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4080–4088 (2018)
25. Zhang, H., Koniusz, P.: Power normalizing second-order similarity network for few-shot learning. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1185–1193. IEEE, January 2019
26. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999 (2018)
27. Liu, Y., et al.: Learning to propagate labels: transductive propagation network for few-shot learning. arXiv preprint arXiv:1805.10002 (2018)
28. Jiang, X., Havaei, M., Varno, F., Chartrand, G., Chapados, N., Matwin, S.: Learning to learn with conditional class dependencies (2018)
29. Allen, K.R., Shin, H., Shelhamer, E., Tenenbaum, J.B.: Variadic meta-learning by Bayesian nonparametric deep embedding (2018)
30. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML Deep Learning Workshop, vol. 2, July 2015