IMAGE DEFORMATION META-NETWORKS FOR ONE-SHOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Humans can robustly learn novel visual concepts even when images undergo various deformations and loose certain information. Incorporating this ability to synthesize deformed instances of new concepts might help visual recognition systems perform better one-shot learning, *i.e.*, learning concepts from one or few examples. Our key insight is that, while the deformed images might not be visually realistic, they still maintain critical semantic information and contribute significantly in formulating classifier decision boundaries. Inspired by the recent progress on meta-learning, we combine a meta-learner with an image deformation network that produces additional training examples, and optimize both models in an end-to-end manner. The deformation network *learns to synthesize* images by fusing a pair of images — a probe image that keeps the visual content and a gallery image that diversifies the deformations. We demonstrate results on the widely used one-shot learning benchmarks (*mini*ImageNet and ImageNet 1K challenge datasets), which significantly outperform the previous state-of-the-art approaches.

1 INTRODUCTION

Deep architectures have made significant progress in various visual recognition tasks, such as image classification and object detection. Such a success typically relies on supervised learning from huge amounts of labeled examples. In real-world scenarios, however, one might not have enough resources to collect large training sets or need to deal with rare visual concepts. It is also unlike the human visual system, which can grasp a novel visual concept under very little supervision. One-shot or low/few-shot learning has thus attracted more and more attention, which aims to build a classifier for a new concept from one or very few labeled examples.

Recent efforts to address this problem have leveraged a *learning-to-learn* or *meta-learning* paradigm, which frames one-shot learning as an optimization problem (Finn et al. (2017); Li et al. (2017); Vinyals et al. (2016); Snell et al. (2017); Ravi & Larochelle (2017)). Meta-learning algorithms train a learning procedure (*i.e.*, learner), which is a parameterized function that maps labeled training sets to classifiers. Meta-learners are trained by sampling a collection of one-shot learning tasks and the corresponding datasets from a large universe of labeled examples of known (base) categories, feeding the sampled small training set to the learner to obtain a classifier, and then computing the loss of the classifier on the sampled test set. The hope is that the learner is able to tackle the recognition of unseen (novel) categories from few training examples.

Despite their noticeable performance improvement, these generic meta-learning algorithms typically treat images as black boxes and ignore the structure of the visual world. By contrast, our biological vision system is very robust and trustable in understanding images that undergo a variety of *deformations* (Vermaak et al. (2005); Boccolini et al. (2018)). For instance, we can easily recognize the concepts/objects in Fig. 1 as the ghost (Fig. 1(a, b)), stitched (Fig. 1(c)), montage (Fig. 1(d)), and partially erased images (Fig. 1(e)). While these deformed images might not be visually realistic, *our key insight* is that they still maintain critical semantic information and presumably serve as "hard examples" that contribute significantly in formulating classifier decision boundaries. Hence, by leveraging such modes of deformation shared across categories, the synthesized deformed images could be used as additional training data to build better classifiers.

A natural question then is how we could produce informative deformations. As shown in Fig. (2), we propose a simple parametrization that linearly combines a pair of images to generate the deformed

(a) (b) (c) (d) (e)

Figure 1: Deformation illustration. Left to right: ghost, stitched, montage, and partly erased images.

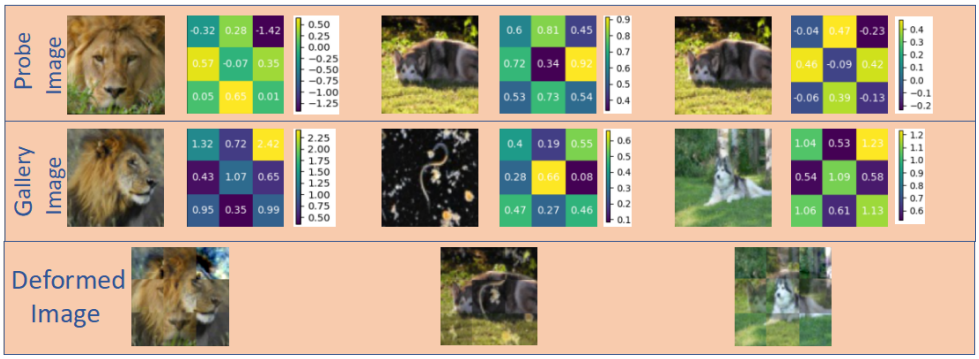


Figure 2: Examples of the deformed images generated by our image deformation network.

image[1]. We use a probe image to keep the visual content and overlay a gallery image at a patch level to introduce appearance variations, which could be attributed to semantic diversity, art effects, or even random noise. Importantly, inspired by Wang et al. (2018), we *learn to deform images* that are useful for classification by the end-to-end meta-optimization of a classification objective that includes image deformation in the model.

Our *Image Deformation Meta-network* (IDeMe-Net) thus consists of two components: a deformation subnetwork and an embedding subnetwork. The deformation network learns to generate the deformed image by linearly fusing the patches of probe and gallery images. Specifically, we treat the given small training set as the probe images and sample addition images from the base categories to form the gallery images. We evenly divide the probe and gallery images into nine patches, and the deformation network estimates the combination weight of each patch. The synthesized deformed images are used to augment the probe images and train the embedding network, which maps images to feature representations and performs one-shot classification. The entire network is trained in an end-to-end manner on base categories.

Our contributions are three-fold. (1) We propose a novel meta-learning based image deformation framework to address one-shot learning, which uses the rich structure of shared modes of deformation in the visual world. (2) Our deformation network learns to synthesize diverse deformed images, which effectively exploits the complementarity and interaction between the probe and gallery image patches. (3) By using the deformation network, we effectively augment and diversify the one-shot training images, leading to the significant performance boost in one-shot learning tasks. Remarkably, our approach achieves the state-of-the-art performances on both the challenging ImageNet1K and *mini*Imagenet datasets by large margins.

2 Related Work

Meta Learning. One research line of one-shot learning is the meta-learning Thrun (1996) in a learning to learn formulation. Generally, meta-learning (Finn et al. (2017); Li et al. (2017); Zhou et al. (2018); Ravi & Larochelle (2017); Munkhdalai & Yu (2017); Wang & Hebert (2016)) aims

[1]The two images could be the same image.

at training a parametrized mapping from a few training instances to some hidden parameters that impact the optimization procedure. Wang et al. (2018) employed meta-learner – "hallucinator" to produce additional training examples. Intrinsically, the hallucinator is a generative adversarial network to synthesize the *realistic* but *imaginary* images by using noise and training images. In contrast, also as a meta-learner, our IDeMe-Net dynamically learns to fuse the probe and gallery images as new images for one-shot learning. Our network aims at producing the *deformed* but *real* (*i.e*. not synthesized) images (as illustrated in Fig. 2). Furthermore, there are also other more general meta-learning strategies in one-shot learning, such as graph CNN (Garcia & Bruna (2018)), memory network (Santoro et al. (2016); Cai et al. (2018)) and relation network (Sung et al. (2018)). The attention mechanism is studied in Wang et al. (2017); Mishra et al. (2016): Wang et al. (2017) analyzed the relation between visual and semantic representations; Mishra et al. (2016) learned the combination of temporal convolutions and soft attention in the labeled training set. Quite different from Wang et al. (2017); Mishra et al. (2016), our IDeMe-Net concentrates on learning to use the complementarity and interaction among visual patches.

Metric Learning. This is another important research line in one-shot learning. The goal of metric-learning is to learn a metric space which can be optimized for one-shot learning. The recent works include Deep Siamese Networks (Koch et al. (2015)), Matching Nets (Vinyals et al. (2016)), PROTO-NET (Snell et al. (2017)) and Gidaris & Komodakis (2018).

Data Augmentation. The key limitation of one-shot learning is the lack of sufficient training images. Augmenting additional instances can help train supervised classifiers (Krizhevsky et al. (2012); Chatfield et al. (2014); Zeiler & Fergus (2014)). The standard techniques include flipping, rotating, adding noise and randomly cropping images. The classical data augmentation includes adding Gaussian perturbation, transforms, or rescales of training images. However, adding noise or jittering on the origin images are particular suspect to visual similarity with the origin images. Furthermore, previous works either seek additional training images by the semi-supervised manners (Ren et al. (2018); Rasmus et al. (2015)), augment new instances by transforms, rescales, or directly synthesize new instance in the feature domain (Wang et al. (2018); Hariharan & Girshick (2017)). Comparing with these models, our IDeMe-Net learns to dynamically fuse patches of two real images in an end-to-end manner. Our newly fused image has the best of two worlds: the IDeMe-Net can save the important patches of original images, while, the image is visually different from both images and thus help to train the one-shot classifier. Additionally, Wang et al. (2018) and Hariharan & Girshick (2017) both synthesize images in the feature domain while we can directly produce images in the image domain.

3 One-Shot Learning Setup

Following the recent work (Vinyals et al. (2016); Ravi & Larochelle (2017); Finn et al. (2017); Snell et al. (2017); Wang et al. (2018)), we frame one-shot learning in a *meta-learning* formulation: we have a base category set $\mathcal{C}_{base}$ and a novel category set $\mathcal{C}_{novel}$, in which $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \emptyset$; correspondingly, we have a base dataset $D_{base} = \{(\mathbf{I}_i, y_i)\}$, $y_i \in \mathcal{C}_{base}$, and a novel dataset $D_{novel} = \{(\mathbf{I}_i, y_i)\}$, $y_i \in \mathcal{C}_{novel}$. We aim to learn a classification algorithm on D_{base} that is able to generalize to unseen categories C_{novel} with one or few training examples per class.

To mimic the one-shot learning scenario, meta-learning algorithms learn from a collection of N-*way*-m-*shot* classification tasks/datasets sampled from D_{base} and are evaluated in a similar way on D_{novel}. Each of these sampled datasets is termed as *an episode*, and we thus have different meta-sets for meta-training and meta-testing. Specifically, we randomly sample N classes $L \sim C_k$ for meta-training (*i.e.*, $k = base$) and meta-testing episode (*i.e.*, $k = novel$). We then randomly sample m and q labeled images per class in L to construct the support set S and query set Q, respectively, *i.e.*, $\mid S \mid = N \times m$; and $\mid Q \mid = N \times q$. During meta-training, we sample S and Q to train our model. During meta-testing, we evaluate by averaging the classification accuracy on query sets Q of many meta-test episodes.

We view the support set as supervised prob images and different from the previous work, we introduce an additional gallery image set G that serves as an *unsupervised* image pool to help generate deformed images. To construct G, we randomly sample some images per base class from *the base dataset*, *i.e.*, $G \sim D_{base}$. The same G is used both in the meta-training and meta-testing episodes. Note that since it is purely sampled from D_{base}, the newly introduced G does not break the standard

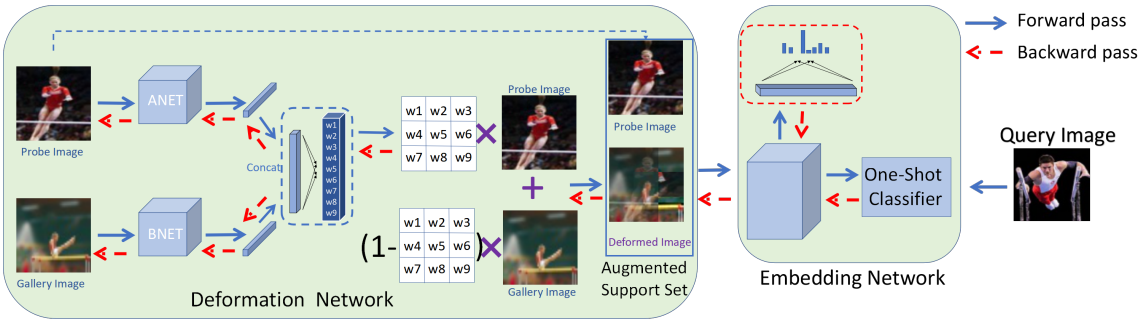


Figure 3: Overall architecture of our image deformation meta-network (IDeMe-Net).

one-shot setup as in (Xu et al. (2016); Snell et al. (2017); Finn et al. (2017); Ravi & Larochelle (2017)). We do not introduce any additional images from the novel categories $\mathcal{C}_{novel}$.

4 Image Deformation Meta-Networks

We now explain our image deformation meta-network (IDeMe-Net) for one-shot learning. Figure 3 shows the architecture of IDeMe-Net $f_\theta(\cdot)$ parametrized by θ. IDeMe-Net is composed of two modules — a deformation subnetwork and an embedding subnetwork. The deformation network adaptively fuses the prob and the gallery images to synthesize the deformed images. The embedding network maps the images to feature representations and construct the one-shot classifier. The entire meta-network is trained in an end-to-end manner.

4.1 Deformation Subnetwork

This subnetwork $f_{\theta_{def}}(\cdot)$ learns to explore the interaction and complementarity between the prob images $\mathbf{I}_{probe}$ ($\{\mathbf{I}_{probe}, y_{probe}\} \in S$) and the gallery images $\mathbf{I}_{gallery} \in G$, and fuses them to generate the synthesized images $\mathbf{I}_{syn}$, *i.e.*, $\mathbf{I}_{syn} = f_{\theta_{def}}(\mathbf{I}_{prob}, \mathbf{I}_{gallery})$. Our key insight is to synthesize meaningful deformed images such that $y_{syn} = y_{probe}$. This is achieved by using two strategies: (1) $y_{syn} = y_{probe}$ is explicitly enforced as a constraint during the end-to-end optimization of the network; (2) we propose an approach to sampling $\mathbf{I}_{gallery}$ that are visually or semantically similar to the images of y_{probe}. Specifically, for a prob image $\{\mathbf{I}_{probe}, y_{probe}\}$, we directly use the feature extractor and one-shot classifier learned in embedding network to select the top $\epsilon\%$ images from G that has the highest class probability as y_{probe} as the $I_{gallery}$. We randomly sampled $I_{gallery}$ from the selected set, which are visually or semantically similar to the probe image Iprobe.

Two branches, ANET and BNET, are used to parse $\mathbf{I}_{probe}$ and $\mathbf{I}_{gallery}$, respectively. Each of them is a residual network (He et al. (2015)) without fully-connected layers. The outputs of ANET and BNET are then concatenated to be fed into a fully-connected layer, which produces a 9-D weight vector w. We use w to construct a weight matrix W as shown in Fig. 3. The deformed image is thus produced as a simple linear weighted combination of $\mathbf{I}_{probe}$ and $\mathbf{I}_{gallery}$ as follows:

$$\mathbf{I}_{syn} = W \odot \mathbf{I}_{probe} + (\mathbf{1} - W) \odot \mathbf{I}_{gallery}, \tag{1}$$

where $\odot$ is the element-wise multiplication. We assign the class label y_{probe} to the synthesized image $\mathbf{I}_{syn}$. For any probe image $\mathbf{I}^i_{probe}$, we sample n_{aug} gallery images from the selected set and produce n_{aug} synthesized images $\{\mathbf{I}^i_{syn}\}$. We thus obtain an augmented support set $\tilde{S}_k = \left\{S_k, \left\{(\mathbf{I}^i_{syn}, y^i_{probe})\right\}_{i=1}^{N*m}\right\}$.

4.2 Embedding Subnetwork

As shown in Fig. 3, the embedding subnetwork $f_{\theta_{emb}}(\cdot)$ consists of a deep convolutional network for feature extraction and a non-parametric one-shot classifier which will be explained later. Given an input image $\mathbf{I}$, we use a residual network (He et al. (2015)) to produce the corresponding feature rep-

resentation $f_{\theta_{emb}}(\mathbf{I})$. To facilitate the training process, we introduce an additional fully-connected layer on top of the embedding network with cross-entropy loss (CELoss) that outputs $|C_{base}|$ scores.

4.3 ONE-SHOT CLASSIFIER

Due to its superior performance, we use the non-parametric prototypical classifier (Snell et al. (2017)) as the one-shot classifier. During each episode, given the sampled S, Q, and G, the deformation network produces the augmented support set $\tilde{S}$. Following Snell et al. (2017), we calculate the prototype vector p_θ^j for each class j in $\tilde{S}$ as

$$p_\theta^j = \frac{1}{Z} \sum_{(\mathbf{I}_i, y_i) \in \tilde{S}} f_{\theta_{emb}}(\mathbf{I}_i) \cdot [\![y_i = j]\!], \tag{2}$$

where $Z = \Sigma_{(\mathbf{I}_i, y_i) \in \tilde{S}} [\![y_i = j]\!]$ is the normalization factor and $f_{\theta_{emb}}(\mathbf{I}_i)$ is the feature extracted by the embedding network. $[\![\cdot]\!]$ is the Iverson's bracket notation: $[\![x]\!] = 1$ if x is true, and 0 otherwise. Given any query image $\mathbf{I}_i \in Q$, its probability of belonging to class k is compute as

$$P_\theta(y_i = k | \mathbf{I}_i) = \frac{\exp\left(\left\| f_\theta(\mathbf{I}_i), p_\theta^k \right\|\right)}{\sum_{j=1}^{N} \exp\left(\left\| f_\theta(\mathbf{I}_i), p_\theta^j \right\|\right)}, \tag{3}$$

where $\| \cdot \|$ indicates the Euclidean distance. The one-shot classifier thus predicts the class label of $\mathbf{I}_i$ as the highest probability over N classes.

5 TRAINING STRATEGY OF IDEME-NET

Algorithm 1 The training procedure of the IDeMe-Net. G is the fixed gallery. f_θ is our IDeMe-Net.

procedure TRAIN_EPISODE ▷ The procedure of one training episode
  $L \leftarrow$ randomly chosen n classes from C_{base}
  $S \leftarrow$ randomly sample instances belong L ▷ sample the support set
  $Q \leftarrow$ randomly sample instances belong L ▷ sample the query set
  learn the prototypical classifer P from $f_{\theta_{emb}}(S)$
  $\tilde{S} \leftarrow S$ ▷ initialize the augment support set
  for c in L **do** ▷ enumerate the choosing class
    $pool \leftarrow$use P to select $\epsilon\%$ images in G having highest class probability of c
    for (I_{prob}, c) in S_c **do** ▷ enumerate the instance of class c in S
      $I_{gallery} \leftarrow$ randomly sampled from $pool$
      $I_{syn} \leftarrow f_{\theta_{def}}(I_{prob}, I_{gallery})$
      $\tilde{S} \leftarrow \tilde{S} \cup (I_{syn}, c)$
    end for
  end for
  learn the prototypical classifer $\tilde{P}$ from $f_{\theta_{emb}}(\tilde{S})$
  use $\tilde{P}$ to classify $f_{\theta_{emb}}(Q)$ and get the prototypical Loss
  use $f_{\theta_{emb}}$ to classify $\tilde{S}$ and get the CELoss
  update θ_{emb} with the CELoss
  update θ_{def} with the prototypical Loss
end procedure

5.1 TRAINING LOSS

Training the entire IDeMe-Net includes two subtasks: (1) training the deformation network which maximally improves the one-shot classification accuracy; and (2) building the robust embedding network which effectively deals with various synthesized images. Note that our one-shot classifier has no parameters, which does not need to be trained. We use the prototypical loss and the cross-entropy loss to train these two subnetworks, respectively.

	Methods	$n = 1$	2	5	10	20
Baselines	Softmax	– /16.3	– /35.9	– /57.4	- / 67.3	– /72.1
	LR	18.3/42.8	26.0/54.7	35.8/66.1	41.1/71.3	44.9/74.8
	SVM	15.9/36.6	22.7/48.4	31.5/61.2	37.9/69.2	43.9/74.6
	Prototype Classifier	17.1/39.2	24.3/51.1	33.8/63.9	38.4/69.9	44.1/74.7
Competitors	Matching Network	– / 43.0	– / 54.1	– / 64.4	– / 68.5	– /72.8
	Prototypical Network	16.9/41.7	24.0/53.6	33.5/63.7	37.7/68.2	42.7/72.3
	Generation-SGM	– / 34.3	– / 48.9	– / 64.1	– / 70.5	– /74.6
	PMN	– / 43.3	– / 55.7	– / 68.4	– / 74.0	– / 77.0
	PMN w/ H	– / 45.8	– / 57.8	– / 69.0	– / 74.3	– / 77.4
	Cos & Att.	– / 46.0	– / 57.5	– / 69.1	– / **74.8**	– / **78.1**
Augmentation	Flipping	17.4/39.6	24.7/51.2	33.7/64.1	38.7/70.2	44.2/74.5
	Gaussian Noise	16.8/39.0	24.0/51.2	33.9/63.7	38.0/69.7	43.8/74.5
	Gaussian Noise(feature level)	16.7/39.1	24.2/51.4	33.4/63.3	38.2/69.5	44.0/74.2
Ours	IDeMe-Net	**23.1/51.0**	**30.1/60.9**	**39.0/69.8**	**42.7**/73.4	**45.0**/75.1

Table 1: **Top-1 / Top-5 accuracy on ImageNet1K Challenge Dataset (ResNet-10).** We use ResNet-10 as the embedding subnetwork. n indicates the number of training examples per class.

Update the deformation network. We optimize the following prototypical loss function to endow the deformation network with the desired one-shot classification ability:

$$\min_\theta \mathbb{E}_{L\sim D_{base}} \mathbb{E}_{S,G,Q\sim L} \left[\sum_{(\mathbf{I}_i, y_i)\in Q} -\log P_\theta \left(y_i \mid \mathbf{I}_i \right) \right], \tag{4}$$

where $P_\theta \left(y_i \mid \mathbf{I}_i \right)$ is the one-shot classifier in Eq. (3). Using the prototypical loss to update the deformation network encourages to generate diverse instances to augment the support set.

Update the embedding network. We use the cross-entropy loss to train the embedding network to directly classify the augmented support set $\tilde{S}$. Note that with the augmented support set $\tilde{S}$, we have enough training instances to train this subnetwork and the cross-entropy loss is the standard loss function in training a supervised classification network. Empirically, we found that using the additional cross-entropy loss speeds up the convergence and improves the recognition performance than using the prototypical loss solely.

5.2 Training Strategy

We summarize the entire training procedure of our IDeMe-Net on the base dataset D_{base} in Alg. 1. During meta-training, we have the gallery G and sample the N-way-m-shot training episode to produce S and Q. The embedding subnetwork learns an initial one-shot classifier $g\left(\cdot\right)$ on the original support set S using Eq. (3). Given a probe image $\mathbf{I}_{probe}$, we then sample the gallery images $\mathbf{I}_{gallery} \sim G$ and train the deformation subnetwork to produce the augmented support set $\tilde{S}$ using Eq. (1). $\tilde{S}$ is further used to update the embedding subnetwork and learn a better one-shot classifier. We then conduct the final one-shot classification on the query set Q and backpropagate the prediction error to update the entire network. During meta-testing, we sample the N-way-m-shot testing episode to produce S and Q from the novel dataset D_{novel}.

6 Experiments

Our IDeMe-Net is evaluated on two standard benchmarks: *mini*ImageNet and ImageNet 1K challenge datasets. The codes and models will be released upon acceptance.

The *mini*ImageNet dataset proposed by Vinyals et al. (2016) is a widely used benchmark in one-shot learning, which includes 600 images per class and has 100 classes in total. Following the data split in Ravi & Larochelle (2017), we use 64, 16, 20 classes as the base, validation, and novel category set, respectively. The hyper-parameters are cross-validated on the validation set. Consistent with Vinyals et al. (2016) and Ravi & Larochelle (2017), we evaluate our model in 5-way-5-shot and 5-way-1-shot settings.

Methods	n=1	2	5	10	20
Softmax	– /28.2	– /51.0	– / 71.0	– /78.4	– /82.3
SVM	20.1/41.6	29.4/57.7	42.6/72.8	49.9/79.1	55.8/83.2
LR	22.9/47.9	32.3/61.3	44.3/73.6	50.9/78.8	56.2/82.4
Prototype Classifier	20.8/43.1	29.9/58.1	42.4/72.3	49.5/79.0	56.0/83.0
Generation SGM Hariharan & Girshick (2017)	– /47.3	– /60.9	– /73.7	– /79.5	– /83.3
IDeMe-Net	**30.3/60.1**	**39.7/69.6**	**47.5/77.4**	**51.3/80.2**	**56.5/83.6**

Table 2: **Top-1 / Top-5 results on ImageNet1K Challenge Dataset (ResNet-50).** We use ResNet-50 as the embedding subnetwork. n indicates the number of training examples per class.

Our model is also evaluated on the large-scale ImageNet 1K dataset. Following the data split in Hariharan & Girshick (2017), we divide the original 1K categories into 389 base (D_{base}) and 611 novel (D_{novel}) categories. The base categories are further divided into two disjoint subsets: base validation set D_{base}^{cv} (193 classes) and base evaluation set D_{base}^{fin} (196 classes) and the novel categories are divided into two subsets as well: novel validation set D_{novel}^{cv} (300 classes) and novel evaluation set D_{novel}^{fin} (311 classes). We use the base/novel validation set D^{cv} for cross-validating hyper-parameters and use the base/novel evaluation set D^{fin} to conduct the final experiments. The same experimental setup is used as in Hariharan & Girshick (2017) and the reported results are averaged over 5 trails. Here we focus on synthesizing novel instances and we thus evaluate the performance primarily on novel categories, *i.e.*, 331-way-n-shot settings, which is also consistent with most of the contemporary work (Vinyals et al. (2016); Snell et al. (2017); Ravi & Larochelle (2017)).

6.1 Results on ImageNet1K challenge dataset

Setup. We use ResNet-10 architectures for ANET and BNET. For a fair comparison with Hariharan & Girshick (2017), we evaluate the performance of using ResNet-10 (Table 1) and ResNet-50 (Table 2) for the embedding network. Stochastic gradient descent (SGD) is used to train IDeMe-Net in an end-to-end manner. It gets converged over 300 epochs. The initial learning rates of ANET, BNET, and the embedding network are set as 3×10^{-3}, 3×10^{-3} and 10^{-1}, respectively and decreased by $1/10$ every 30 epochs. The batch size is set as 32. We randomly sample 10 images per base category to construct the gallery G and we set ϵ as 2. Note that G is *fixed* during the entire experiments. ANET, BNET, and the embedding network are trained from scratch on D_{base}. Our model is evaluated on D_{novel}. n_{aug} is cross-validated as 8, which balances between the computational cost and the augmented training data scale.

Baselines and competitors. We compare against several baselines and competitors as follows. (1) We directly train a ResNet-18 feature extractor on D_{base} and use it to compute image features on D_{novel}. We then train standard supervised classifiers on D_{novel}, including neural network(Softmax), support vector machine (SVM), logistic regression (LR), and prototype classifier. The neural network classifier consists of 1 fully-connected layer and 1 softmax classification layer. (2) We also compare with state-of-the-art approaches to one-shot learning, such as matching network (Vinyals et al. (2016)), generation SGM (Hariharan & Girshick (2017)), prototypical network (Snell et al. (2017)), Cosine Classifier & Att. Weight Gen(Cos & Att.) (Gidaris & Komodakis (2018)),PMN and PMN w/H (Wang et al. (2018)) . (3) The standard data augmentation methods are also compared here: "flipping": the input image is flipped from left to right; "Gaussian noise": cross-validated Gaussian noise $\mathcal{N}(0, 10)$ is added to each pixel of the input image; "Gaussian noise (feature level)": cross-validated Gaussian noise $\mathcal{N}(0, 0.3)$ is added to each dimension of the ResNet feature of each image. For fair comparisons, all theses augmentation methods use the prototype classifier as the one-shot classifier.

Results. Tables 1 and 2 summarize the results of using ResNet-10 and ResNet-50 as the embedding sub-network, respectively. Fig. 4(a) further highlights that our IDeMe-Net consistently outperforms all the baselines and competitors by large margins. For example, using ResNet-10, the top-1 accuracy of IDeMe-Net in Table 1 is superior to prototypical network by 6% when $n = 1, 2, 5$, showing the sample efficiency of IDeMe-Net for one-shot learning. The top-5 accuracy in Table 1 demonstrates the similar trend, and our IDeMe-Net beats prototypical network, the second best competitors,

	Methods	$n = 1$	2	5	10	20
Baselines	LR	18.3/42.8	26.0/54.7	35.8/66.1	41.1/71.3	44.9/74.8
	Prototype Classifier	17.1/39.2	24.3/51.1	33.8/63.9	38.4/69.9	44.1/74.7
Variants	IDeMe-Net - CE Loss	21.3/50.0	28.0/58.3	37.7/69.4	41.3/71.6	44.3/74.3
	IDeMe-Net - Proto Loss	15.3/36.7	21.4/50.4	31.7/62.0	37.9/69.0	43.7/73.7
	IDeMe-Net - $g\left(\mathbf{I}_{gallery}\right)$	17.0/39.3	24.0/50.7	33.6/63.5	38.0/69.2	43.7/73.8
	IDeMe-Net - Aug. Testing	17.0/39.1	24.30/51.3	33.5/63.8	38.0/69.1	43.8/74.5
	IDeMe-Net - Def. Network	15.9/38.0	24.1/50.1	32.6/63.3	38.2/68.9	42.4/73.1
	IDeMe-Net - Gallery	17.5/39.4	24.2/51.4	33.5/63.7	38.7/70.3	44.5/74.5
	Gallery Baseline	15.7/37.8	22.7/49.8	31.9/62.6	38.0/68.7	43.5/73.8
Block Size	IDeMe-Net (1×1)	16.2/39.3	24.4/52.1	32.9/63.0	38.8/69.5	42.7/73.2
	IDeMe-Net (5×5)	**24.1**/51.7	30.3/61.2	**39.6/70.4**	42.4/73.2	44.3/74.6
	IDeMe-Net (7×7)	23.8/**52.1**	30.2/**61.3**	39.1/70.2	**42.7**/73.1	44.5/74.7
	IDeMe-Net (pixel level)	17.3/39.0	23.8/51.2	34.1/63.7	38.5/70.2	43.9/74.5
Ours	IDeMe-Net	23.1/51.0	**30.4**/60.9	39.0/69.8	**42.7/73.4**	**45.0/75.1**

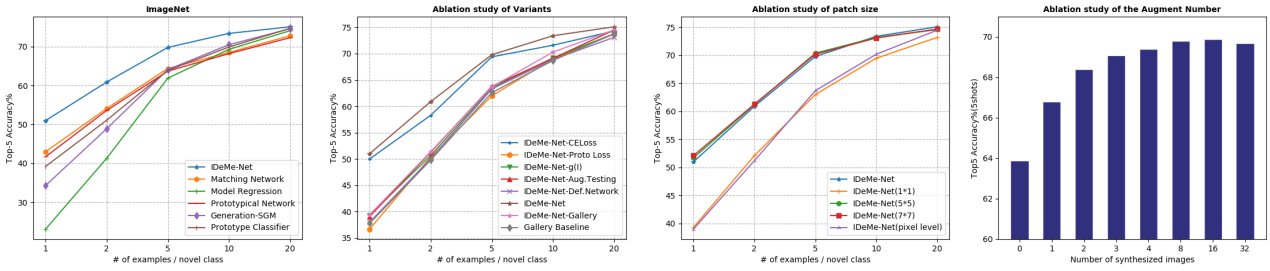
Table 3: **Top-1 / Top-5 accuracy of the ablation study on ImageNet1k challenge dataset (ResNet-10).** We use ResNet-10 as the embedding subnetwork.

by more than 9% when $n = 1$. Using ResNet-50 as the embedding sub-network, the performance of all the approaches improves and our IDeMe-Net consistently achieves the best performance, as shown in Table 2.

6.2 Ablation Study on ImageNet1k

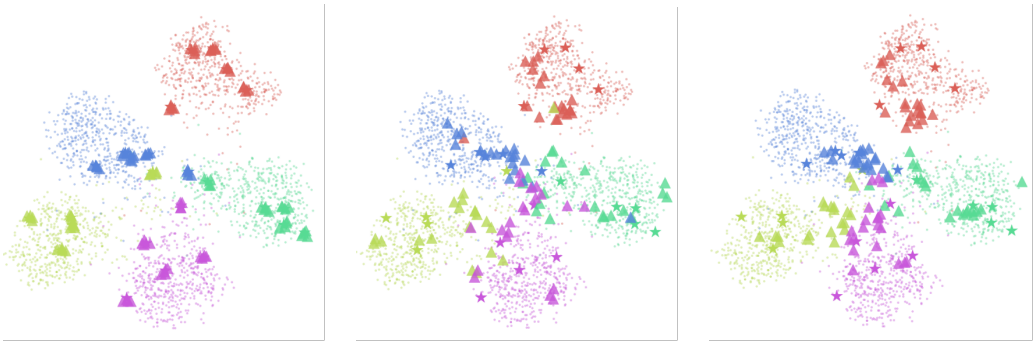
We conduct extensive ablation study to evaluate the contribution of each component in our model.

Variants of IDeMe-Net. We consider seven different variants of our IDeMe-Net, as shown in Fig. 4(b) and Table 3. (1) IDeMe-Net-CE Loss: the embedding sub-network is trained using only the prototypical loss *without* the cross-entropy loss (CE loss). (2) IDeMe-Net - Proto Loss: the embedding sub-network is trained using only the cross-entropy loss *without* the prototypical loss. (3) IDeMe-Net - $g\left(\mathbf{I}_{gallery}\right)$: the gallery images are randomly chosen in IDeMe-Net *without* predicting their class probability. (4) IDeMe-Net - Aug. Testing: the deformed images are not produced in the testing phase. (5) IDeMe-Net - Def. Network: the weight matrix W in Eq. (1) is randomly generated instead of using the learned deformation sub-network. (6) IDeMe-Net - Gallery: the gallery images are directly sampled from the support set instead of constructing an additional Gallery. (7) Gallery Baseline: we simply use the gallery images to serve as the deformed images. As shown in Fig. 4(b), our full IDeMe-Net model outperforms all these seven variants, showing that each component is essential and complementary to each other. We note that (1) **Using CELoss and prototypical loss to update the embedding sub-network and the deformation sub-network, respectively, achieves the best result.** As shown in Fig. 4(b), the accuracy of IDeMe-Net - CELoss is marginally lower than IDeMe-Net but still higher than the prototypical classifier baseline, while IDeMe-Net - Proto Loss underperforms the baseline. (2) **Our strategy for selecting the gallery images is the key to diversify the deformed images.** As we can see, randomly choosing the gallery image (IDeMe-Net - $g\left(\mathbf{I}_{gallery}\right)$) or sampling the gallery images from the support set (IDeMe-Net - Gallery) obtains no performance improvement. One reasonable explanation is that they only introduce noise or redundancy and do not bring in efficient information. (3) **Our improved performance comes from the diversified deformed images, rather than the embedding sub-network.** Without producing the deformed images in the testing phase (IDeMe-Net - Aug. Testing), the performance is close to the baseline, suggesting that training on the deformed images does not obviously benefit from the embedding sub-network. That is, the performance gain of our IDeMe-Net mainly results from the deformed images generated in the testing phase. (4) **Our meta-learned deformation sub-network effectively exploits the complementarity and interaction between the probe and gallery image patches, producing the key information in the deformed images.** To show this point, we investigate two deformation strategies: randomly generating the weight vector w (IDeMe-Net - Def. Network) and setting all the weights to be 0 (Gallery Baseline); in the latter case, it is equivalent to purely using the gallery images to serve as the deformed images. Both strategies perform worse than the prototypical classifier baseline, indicating the importance of meta-learning a deformation strategy.



(a) (b) (c) (d)

Figure 4: **Ablation study on the ImageNet1k challenge dataset:** (a) highlights the comparison with several competitors; (b) shows the impact of different components in our IDeMe-Net; (c) analyzes the impact of different division schemes; (d) shows how the performance changes with respect to the number of synthesized images. **Best viewed in color with zoom.**



(a) Gaussian Baseline (b) Gallery Baseline (c)IDeMe-Net

Figure 5: t-SNE visualization. Dots, stars, and triangles represent the real examples, the probe images, and the synthesized images, respectively. (a) synthesis by adding Gaussian noises. (b) synthesis by directly using the gallery images. (c) synthesis by our IDeMe-Net.

Different division schemes. In the deformation sub-network and Eq. (1), we equally split the image into 3×3 patches. Some alternative division schemes are compared in Table 1 and Fig. 4(c). Specifically, we consider the 1×1, 5×5, 7×7, and pixel level division schemes in Eq. (1) and report the results as IDeMe-Net (1×1), IDeMe-Net (5×5), IDeMe-Net (7×7), and IDeMe-Net (pixel level), respectively. The experimental results suggest the patch-level fusion, rather than image-level or pixel-level fusion in our IDeMe-Net. The image-level division (1×1) ignores the local combination and deforms through a global combination, thus decreasing the diversity. The pixel-level division is particularly suspect to the disarray of the local information while region-level division (3×3, 5×5, 7×7) considers the region as the basic unit to maintain some local information. In addition, the results show that using a fine-grained patch size (*i.e.*, 5×5 division and 7×7 division) may achieve slightly better results than our 3×3 division. This also makes senses since fine-grained patch deformed information is learned. In brief, our region-level division not only maintains the significant region information but also deforms it to increase diversity.

Number of synthesized images. We also show how the top-1 accuracy changes with respect to the number of synthesized images in Fig. 4(d). Specifically, we change the number of synthesized images n_{aug} in the deformation sub-network, and visualize the 5-shot top-5 accuracy on the Imagenet-1K dataset. It shows that when n_{aug} is changed from 0 to 8, the performance of our IDeMe-Net is gradually improved. The performance saturates when enough synthesized images are generated ($n_{aug} > 8$).

Visualization. Fig. 5 shows t-SNE van der Maaten & Hinton (2008) visualizations for novel classes from our IDeMe-Net, the Gaussian baseline, and the Gallery baseline. For the Gaussian baseline, the synthesized images are heavily clustered and close to the prob images. By contrast, the synthesized

Methods	*mini*ImageNet (%)	
	1-shot	5-shot
MAML	48.70±1.84	63.11±0.92
Meta-SGD	50.47±1.87	64.03±0.94
Matching Nets	43.56±0.84	55.31±0.73
Prototypical Network	49.42±0.78	68.20±0.66
RELATION NET	57.02±0.92	71.07±0.69
SNAIL	55.71±0.99	68.88±0.92
Delta-encoder	**58.7**	73.6
Cos & Att.	55.45±0.89	70.13 ±0.68
Prototype Classifier	52.54±0.81	72.71±0.73
IDeMe-Net	57.71±0.89	**74.34**±0.78

Table 4: **Top-1 accuracy on *mini*ImageNet.** The "±" indicates 95% confidence intervals over tasks.

images of our IDeMe-Net scatter widely in the class manifold and tend to locate more around the class boundaries. For the Gallery baseline, the synthesized images are the same as the gallery images and occasionally fall into manifolds of other classes. Interesting, comparing Fig. 5(b) and Fig. 5(c), our IDeMe-Net effectively deforms those misleading gallery images back to the correct class manifold.

Deformed Images. Here we show some examples of our deformed images in the Fig.2. In the first example, IDeMe-Net synthesizes a ghost image combining two lions with different head poses. In the second and the third samples, we have the same prob image but with different gallery images. Our IDeMe-Net has different react according to the contents of the gallery image: ignore most patches in the misleading gallery image and utilize the significant patch in the highly relevant gallery image. We can observe that the weight(first row) in the second sample is much larger than in the third sample. This intuitively shows why our model work.

6.3 Results on *mini*ImageNet

Setup and competitors. We use a ResNet-18 architecture as the embedding sub-network. We use the same experimental setting as the ImageNet1k challenge dataset. As summarized in Table 4, we mainly focus on three groups of the competitors: (1) meta-learning algorithms, such as MAML (Finn et al. (2017)) and Meta-SGD (Li et al. (2017)); (2) metric-learning algorithms, including Matching Nets (Vinyals et al. (2016)), Prototypical Network (Snell et al. (2017)), RELATION NET (Sung et al. (2018)), SNAIL(Mishra et al. (2016)), Delta-encoder(Schwartz et al. (2018)) and Cosine Classifier & Att. Weight Gen(Cos & Att.) (Gidaris & Komodakis (2018)).

Results. We report the results in Table 4 (a). Impressively, our IDeMe-Net consistently outperforms all these state-of-the-art competitors by large margins in 5-shot classification scenarios. This further validates the general effectiveness of our proposed model in addressing one-shot learning tasks.

7 Conclusions

In this paper, we proposed a conceptually simple but powerful approach to one-shot learning that uses a trained image deformation network to generate additional examples. Our deformation network leverages unsupervised gallery images to synthesize deformed images and is trained end-to-end with meta-learning. Extensive experiments demonstrate that our approach achieves the state-of-the-art performance on multiple one-shot learning benchmarks by large margins.

References

Alessandro Boccolini, Alessandro Fedrizzi, and Daniele Faccio. Ghost imaging with the human eye. 2018. 1

Q. Cai, Y. Pan, T. Yao, C. Yan, and T. Mei. Memory Matching Networks for One-Shot Image Recognition. 2018. 2

Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 2

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 1, 2, 3, 6.3

V. Garcia and J. Bruna. Few-Shot Learning with Graph Neural Networks. 2018. 2

Spyros Gidaris and Nikos Komodakis. Dynamic Few-Shot Visual Learning without Forgetting. In *CVPR*, 2018. 2, 6.1, 6.3

B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 2017. 2, 6, 6.1

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2015. 4.1, 4.2

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML – Deep Learning Workshok*, 2015. 2

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. In *arxiv:1707.09835*, 2017. 1, 2, 6.3

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *ICLR*, 2016. 2, 6.3

T. Munkhdalai and H. Yu. Meta networks. In *ICML*, 2017. 2

Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder networks. In *NIPS*, 2015. 2

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 1, 2, 3, 6

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *Proceedings of 6th International Conference on Learning Representations ICLR*, 2018. 2

Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. One-shot learning with memory-augmented neural networks. In *arx*, 2016. 2

E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, R. Feris, A. Kumar, R. Giryes, and A. M. Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *NIPS*, 2018. 6.3

Jake Snell, Kevin Swersky, and Richard S. Zemeln. Prototypical networks for few-shot learning. In *NIPS*, 2017. 1, 2, 3, 4.3, 6, 6.1, 6.3

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 2, 6.3

S. Thrun. *Learning To Learn: Introduction*. Kluwer Academic Publishers, 1996. 2

Laurens van der Maaten and Geoffrey Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 2008. 6.2

J. Vermaak, S. Maskell, and M. Briers. Online sensor registration. In *Aerospace, 2005 IEEE Conference*, pp. 2117–2125, 5-12 March 2005. doi: 10.1109/AERO.2005.1559503. 1

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NIPS*, 2016. 1, 2, 3, 6, 6.1, 6.3

Peng Wang, Lingqiao Liu, Chunhua Shen, Zi Huang, Anton Hengel, and Heng Tao Shen. Multi-attention network for one shot learning. In *CVPR*, pp. 6212–6220, 07 2017. 2

Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan. Low-Shot Learning from Imaginary Data. In *CVPR*, 2018. 1, 2, 3, 6.1

Yuxiong Wang and Martial Hebert. Learning to learn: model regression networks for easy small sample learning. In *ECCV*, 2016. 2

Zhongwen Xu, Linchao Zhu, and Yi Yang. Few-shot object recognition from machine-labeled web images. *CoRR*, abs/1612.06152, 2016. URL `http://arxiv.org/abs/1612.06152`. 3

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 2

Fengwei Zhou, Bin Wu, and Zhenguo Li. Deep meta-learning: Learning to learn in the concept space. In *arxiv:1802.03596*, 2018. 2