Image Block Augmentation for One-Shot Learning

Zitian Chen[1], Yanwei Fu[23], Kaiyu Chen[1], Yu-Gang Jiang[123]
[1]School of Computer Science, Fudan University; [2] School of Data Science, Fudan University;
[3] Shanghai Key Lab of Intelligent Information Processing, Fudan University;
Email:{chenzt15, yanweifu,kychen15, ygj}@fudan.edu.cn

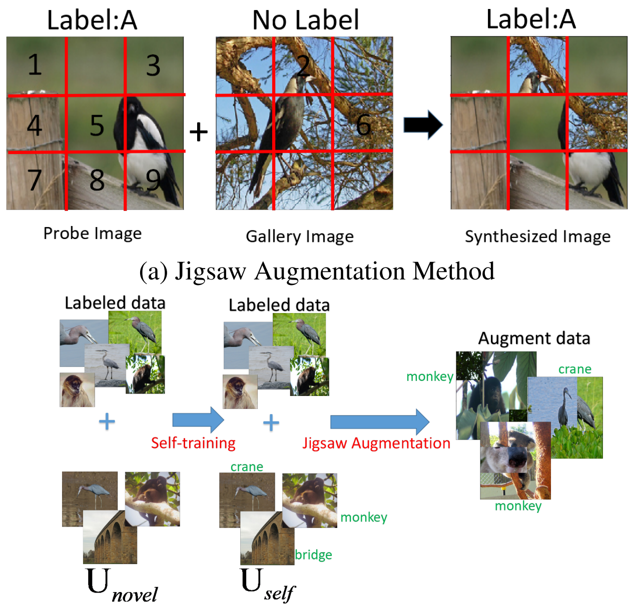
Abstract

Given one or a few training instances of novel classes, one-shot learning task requires that the classifier generalizes to these novel classes. Directly training one-shot classifier may suffer from insufficient training instances in one-shot learning. Previous one-shot learning works investigate the meta-learning or metric-based algorithms; in contrast, this paper proposes a Self-Training Jigsaw Augmentation (Self-Jig) method for one-shot learning. Particularly, we solve one-shot learning by directly augmenting the training images through leveraging the vast unlabeled instances. Precisely our proposed Self-Jig algorithm can synthesize new images from the labeled probe and unlabeled gallery images. The labels of gallery images are predicted to help the augmentation process, which can be taken as a self-training scheme. Intrinsically, we argue that we provide a very useful way of directly generating massive amounts of training images for novel classes. Extensive experiments and ablation study not only evaluate the efficacy but also reveal the insights, of the proposed Self-Jig method.

Introduction

Inspired by the ability of people to acquire a new concept from a handful of examples, one-shot learning is recently studied with the goal of learning classifiers that generalize to novel classes which only have one, or a few training instances of each class. While this problem is quite difficult, the main obstacle is lacking enough training images in the novel classes. To address this task, previous efforts explored transferring knowledge from source base classes to help learn a classifier on target novel classes, by the ways of meta-learner (Vinyals et al. 2016; Snell, Swersky, and Zemeln 2017) or metric space(Finn, Abbeel, and Levine 2017; Zhou, Wu, and Li 2018; Li et al. 2017), and to a lesser extent, data augmentation in novel classes (Wang et al. 2018; Hariharan and Girshick 2017).

In term of Gestalt principles of perceptual grouping (Desolneux, Moisan, and Morel 2004), humans can naturally perceive objects as the organized patterns and objects. People have been entertained by solving Jigsaw puzzles by assembling the pieces into a complete picture. In this game, even though some pieces are missing, we are still able to infer the content drawn in Jigsaw. This game thus inspires us a totally different way of addressing one-shot learning by replacing some pieces of labeled images and producing the deformed additional images to help train the one-shot classifier.



(a) Jigsaw Augmentation Method

(b) Self-training Jigsaw Augmentation

Figure 1: Illustration of our augmentation method. The synthesized image is visually consistent in each block (*i.e.,* local consistent), but not in all blocks (*i.e.,* not global consistent).

Formally, to address one-shot learning task, we propose a Self-Training Jigsaw Augmentation (Self-Jig) model to generate synthesized training images of novel classes. In particular, our fundamental idea is the proposed Jigsaw augmentation algorithm that can create a new image from the probe and gallery images. In the source domain, we first train a base network; the synthesized images are composed by the labeled probe and labeled gallery images which are randomly sampled from training data. In the target domain, we learn to generalize the one-shot classifier. Particularly, the Self-Jig algorithm learns to synthesize new images from the labeled probe image, and unlabeled gallery image which must have the same class label predicted as the probe image by one-shot classifier. As illustrated in Fig. 1(a), our Jigsaw augmentation method can synthesize new image by replacing some blocks. The synthesized images can then be utilized to update the classifiers in the target domain. Critically, we present a very effective way of directly generating a large number of training images in the novel classes. The whole pipeline of our method is illustrated in Fig. 1(b). Remarkably, our framework is orthogonal and complementary to all previous one-shot methods (Finn, Abbeel, and Levine 2017; Zhou, Wu, and Li 2018; Li et al. 2017; Vinyals et al. 2016); and the augmented training images, in principle, can be utilized to train classifiers in the target domain. Extensive experiments show that our framework can greatly improve the one-shot learning performance.

Related Work

We briefly review the connections and differences between our method with several related works. Note that the literature on these topics is vast, only works that most relevant are summarized here.

One-Shot Learning. Previous works explored it by metric-learning and meta-learning. The former one aims at learning a metric space in which the one-shot classification can be performed. The recent typical works on metric learning include Deep Siamese Networks(Koch, Zemel, and Salakhutdinov 2015), Matching Nets (Vinyals et al. 2016), PROTO-NET (Snell, Swersky, and Zemeln 2017), RELATION NET (Sung et al. 2018) and MACO (Hilliard et al. 2018). The meta-learning algorithms (Finn, Abbeel, and Levine 2017; Li et al. 2017; Zhou, Wu, and Li 2018; Ravi and Larochelle 2017; Munkhdalai and Yu 2017; Wang and Hebert 2016) train a "high-level meta-structure" – meta-learner to fast optimize and adapt the weights of low-level networks. Additionally, other strategies have also been investigated in one-shot learning, such as attention (Wang et al. 2017; Guttenberg and Kanai 2018), graph CNN (Garcia and Bruna 2018), and memory network (Santoro et al. 2016; Cai et al. 2018). Different from all these works, our model directly generates synthesized images for one-shot classes; thus is orthogonal and complementary to all these previous works.

Data Augmentation. In standard supervised classification, researchers usually utilize many data augmentation methods, such as flipping, rotating, adding noise and randomly cropping images, to train deep networks (Krizhevsky, Sutskever, and Hinton 2012; Chen et al. 2018; Zeiler and Fergus 2014). More advanced augmentation method is to synthesize or edit images directly, by hallucinating (Wang et al. 2018), shrinking and hallucinating features (Hariharan and Girshick 2017), mixing paring samples (Inoue 2018), or randomly erasing parts of images (Zhong et al. 2017). Quite differently, our Jigsaw augmentation dynamically synthesizes the new image by integrating two images: one is the labeled image, the other one is selected from the unlabeled image set by self-training scheme. Thus visually our new image can be both similar and yet significant different to the labeled image.

Semi-supervised Learning. It focuses on facilitating unlabeled data to help learn classifiers, particularly by deep architectures (Rasmus et al. 2015; Laine and Aila 2016). These works nevertheless still require an amount of labeled data to initially train a deep network, thus limited in the one-shot learning scenario. Recently, Ren *et al.* (Ren et al. 2018) addressed the semi-supervised few-shot learning tasks by extending PROTO-NET (Snell, Swersky, and Zemeln 2017). In contrast, our model primarily targets at augmenting training data by the proposed self-training Jigsaw augmentation model, which is also very different from standard self-training scheme (Yarowsky 1995; Chuck Rosenberg 2005). That is, our framework utilizes the self-training scheme to select unlabeled images as the gallery images to help synthesize new images by Jigsaw augmentation method.

Image Block Augmentation

In one-shot learning, we are given the base categories C_{base}, and novel categories C_{novel} in the source/target domain respectively. The C_{base} have sufficient labeled image data $D_{base} = \{\mathbf{I}_i, y_i\}$, $y_i \in C_{base}$. We conduct one-shot learning on the C_{novel} which only have a small amount of labeled data $D_{novel} = \{\mathbf{I}_i, y_i\}$, $y_i \in C_{novel}$; additionally, we have large amounts of unlabeled data of novel categories $U_{novel} = \{\mathbf{I}_u\}$ in the target domain.

Our framework has two steps. We firstly use base classes to train the base network, which is further fine-tuned as the robust feature extractor by the augmented data, generated by our Jigsaw augmentation method. We further employ the augmentation method on novel classes by using unlabeled data to help train the classifier for one-shot recognition.

Jigsaw Augmentation Method to Synthesize Images

We introduce a novel Jigsaw data augmentation method. As Fig. 1(a), all the images are resized to the same fixed size, and divided into nine blocks. Some blocks of *probe image* $\mathbf{I}_i$ are randomly chosen and replaced by m blocks ($m \leq 4$) of corresponding positions from *gallery image* $\mathbf{I}_u$.

Generally, we require $\mathbf{I}_i$ and $\mathbf{I}_u$ to have the same class label. However, in target domain C_{novel}, the gallery image $\mathbf{I}_u$ is unlabeled; thus we use the predicted label of $\mathbf{I}_u$ by self-training scheme. Such a way has several benefits.

First, when the class label of $\mathbf{I}_u$ is very different from that of $\mathbf{I}_i$, (*i.e.*, the labels of $\mathbf{I}_u$ is wrongly predicted), the synthesized image $\widetilde{\mathbf{I}}_i$ can be taken as the randomly erased version of $\mathbf{I}_i$. Interestingly such synthesized image $\widetilde{\mathbf{I}}_i$ has been shown in (Zhong et al. 2017) to be able to improve the generalization ability of the deep network, and complementary to the classical augmentation techniques, *e.g.*, random cropping, and flipping.

Second, when $\mathbf{I}_u$ is very related to $\mathbf{I}_i$, *i.e.*, they may share the same or similar class label, the $\widetilde{\mathbf{I}}_i$ can be taken as the semantic composition of $\mathbf{I}_u$ and $\mathbf{I}_i$. That means, the features extracted from $\mathbf{I}_u$ and $\mathbf{I}_i$, should in principle, be close to each other, or distributed on the same class manifold/cluster. And the $\widetilde{\mathbf{I}}_i$ that is visually similar to both $\mathbf{I}_u$ and $\mathbf{I}_i$, will also be located in the same class manifold/cluster, as visualized in Fig. 2. Note that visually $\widetilde{\mathbf{I}}_i$ is only locally, but not globally consistent, since some blocks are replaced (in Fig.

1(a)). Thus we need to learn a feature extractor that can still robustly understand such images.

Learning Robust Feature Extractor of Base Classes

To learn a feature extractor, we firstly train base network on D_{base}, and then fine-tune the base network with synthesized images generated by Jigsaw augmentation method, which replaces some blocks $m \leq 4$. Particularly, we use the base categories C_{base} to train a supervised classification network $f\left(\mathbf{I};\theta\right)$; θ indicates the parameter set. The $f\left(\mathbf{I};\theta\right)$ can be a classification network, *e.g.*, ResNet (He et al. 2015). In D_{novel}, we can use the final layer output of $f\left(\mathbf{I};\theta\right)$ as the extracted features of the image $\mathbf{I}$.

The $f\left(\mathbf{I};\theta\right)$ should be trained to robustly extract the features of the synthesized image set $\left\{\widetilde{\mathbf{I}}_i\right\}$ which is locally consistent, but not globally. Thus, we fine-tune the $f\left(\mathbf{I};\theta\right)$ by the synthesized images from synthesized image set $\widetilde{D}_{base}$ until converged. To construct the synthesized image set $\widetilde{D}_{base}$, the probe $\mathbf{I}_i$ and gallery $\mathbf{I}_u$ images are randomly sampled from D_{base}. We want to highlight several issues with this fine-tuning step.

Firstly, rather than using only $\widetilde{D}_{base}$ to fine-tune $f\left(\mathbf{I};\theta\right)$, we actually mix $\widetilde{D}_{base}$ and D_{base} to avoid the "*catastrophic forgetting*" of the network (MichaelMcCloskey and Cohen 1989). Otherwise, the fine-tuned network will be inclined to extract robust features of images in $\widetilde{D}_{base}$, but not D_{base}.

Secondly, training by the augmented set $\widetilde{D}_{base}$ does not improve the classification performance of $f\left(\mathbf{I};\theta\right)$ on the source domain. This is very reasonable, since the gallery images that generate $\widetilde{D}_{base}$ have already been used to train $f\left(\mathbf{I};\theta\right)$. In other words, any each block of newly synthesized image $\widetilde{\mathbf{I}}_i$ in $\widetilde{D}_{base}$ has been trained in $f\left(\mathbf{I};\theta\right)$; and there is no new information being learned. This motivates our semi-supervised framework of using unlabeled image set U_{novel} to help one-shot learning. Thus once $f\left(\mathbf{I};\theta\right)$ is trained, we use it as the generic feature extractor for one-shot recognition.

Thirdly, Noroozi *et al.* (Noroozi and Favaro 2016) took solving Jigsaw puzzles as a pretext task to train a context-free network in boosting the performance of object classification and detection. In contrast, our fine-tuning step introduced here can be used in any base network rather than the specially designed network as in (Noroozi and Favaro 2016).

Furthermore, the target of our fine-tuning step is only to teach the network to understand the synthesized images, rather than directly boosting the performance of one-shot classification.

Self-Training Jigsaw Augmentation Algorithm

We explain our self-training Jigsaw augmentation algorithm in details. Particularly, in novel categories, the we can extract the features $\mathbf{x} = f\left(\mathbf{I};\theta\right)$ of image $\mathbf{I} \in D_{novel}$; and we further train a classifier $g\left(\mathbf{x};\eta\right)$ with the parameter set η. The classifier $g\left(\cdot\right)$ can either be Support Vector Machine (SVM), Logistic Regression (LR) or Neural Network. The $g\left(\mathbf{x};\eta\right)$ is directly applied to U_{novel} to predict the class label of each image $\mathbf{I}_u$. Among these unlabeled images, the $g\left(\mathbf{x};\eta\right)$ selects the image subset $U_{self} = \left\{\mathbf{I}_u\right\} \subseteq U_{novel}$ of high prediction confidence, as well as the corresponding label set $\mathbf{y}_{self} = \left\{g\left(f\left(\mathbf{I}_u;\theta\right);\eta\right)\right\}$.

We then apply the Jigsaw augmentation method by taking D_{novel} and U_{self} as probe and gallery set respectively. We set the replacing block $m \leq 2$. Specifically, given one labeled image $\mathbf{I}_i$ as the probe image, we randomly select one gallery image $\mathbf{I}_u \in U_{self}$, conditioning that $y_i = g\left(f\left(\mathbf{I}_u;\theta\right);\eta\right)$. We can thus generate the augmented dataset $\widetilde{D}_{novel} = \left\{\widetilde{\mathbf{I}}_i, y_i\right\}$ by Jigsaw augmentation method. In one-shot recognition, we use the network $f\left(\mathbf{I};\theta\right)$ to extract the image features of D_{novel} and $\widetilde{D}_{novel}$ and train the classifier $\widetilde{g}\left(\mathbf{x};\eta\right)$, which can be applied to categorize the testing instances.

Intrinsically, our algorithm can indeed use the self-training scheme (Chuck Rosenberg 2005) of updating $g\left(\mathbf{x};\eta\right)$, which is one of the simplest semi-supervised algorithms. Note that traditional self-training algorithm may suffer from the semantic drift by reinforcing poor predictions. In contrast, the synthesized images generated by our Jigsaw augmentation algorithm may have better performance in learning the one-shot learning models. This is because the synthesized images can be either the cropped labeled probe images or semantically composed images. Note that it is useless to synthesizing new images by simply exchanging blocks between labeled images due to that does not bring in any new information. This point is also evaluated in our ablation study of the experiments.

Experiments

Extensive experiments are conducted on *mini*ImageNet and ImageNet1k challenge datasets. The codes and models will be released upon acceptance.

The *mini*ImageNet proposed in (Vinyals et al. 2016), is one of the most widely used benchmark dataset on one-shot learning. It has 60,000 images from 100 classes with 600 images per class. The data split setup is used by (Ravi and Larochelle 2017) with 64, 16, 20 classes as training validation, and testing set individually. Only the 64 classes of training set serve as the C_{base} to train feature extractor. As in (Vinyals et al. 2016; Ravi and Larochelle 2017), we consider 1-shot and 5-shot classification for five classes as C_{novel} randomly chosen from testing class set; and 15 examples per class for evaluation in each test round. The results are averaged and reported over multiple rounds. Additionally, in each round, we randomly select, 30 unlabeled images per class (U_{novel}) that have not used for training and testing. We employ a self-training scheme to select half of the images from U_{novel} as U_{self}.

We also conduct the experiments on the large-scale dataset – ImageNet1k challenge dataset. We use the same split C_{base} and C_{novel} as proposed in (Hariharan and Girshick 2017). The Top-1 and Top-5 accuracy are reported. The results are averaged over 5 repeated runs as (Hariharan and Girshick 2017). On each novel category, we randomly sample 50 images per class as the unlabelled set U_{novel}.

	Methods	n=1	2	5	10	20
Baselines	Softmax	–/14.1	–/33.3	–/56.2	–/66.2	–/71.5
	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
Competitors	Matching Network$^\sharp$ (Vinyals et al. 2016)	–/43.0	–/54.1	–/64.4	–/68.5	–/72.8
	Model Regression$^\sharp$ (Wang and Hebert 2016)	16.9/41.7	24.0/53.6	33.5/63.7	37.7/68.2	42.7/72.3
	Generation-SGM (Hariharan and Girshick 2017)	–/34.3	–/48.9	–/64.1	–/70.5	–/74.6
	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	Self-Jig (SVM)*	23.8/**51.9**	31.5/**62.1**	38.6/**69.6**	42.4/**73.2**	45.8/**75.7**
	Self-Jig(SVM)$^{+}$	17.7/39.1	24.9/51.3	34.1/64.9	39.1/70.6	44.2/74.9
	Self-Jig (LR)*	22.0/49.3	30.1/60.5	38.1/68.7	41.9/72.2	44.9/74.9
	Self-Jig (LR)$^{+}$	20.1/45.3	28.1/56.9	36.5/66.4	41.3/71.5	44.8/74.6
Ablation Study I	Rob (LR)	16.7/40.5	24.2/52.5	33.7/64.2	38.9/69.4	43.1/73.0
	Jigsaw (LR)	17.5/42.2	25.1/53.9	33.9/64.7	39.8/67.9	41.3/70.6
	Self-T (LR)	16.1/41.2	24.6/52.9	32.8/64.0	39.7/68.8	42.2/72.7
	Rob (SVM)	14.3/33.2	20.5/44.4	29.4/58.6	36.3/67.5	42.0/72.9
	Jigsaw (SVM)	14.2/33.9	20.1/45.2	29.2/58.4	34.1/65.1	40.0/68.4
	Self-T (SVM)	16.0/36.7	22.7/48.4	30.8/60.0	35.7/66.7	40.9/69.5
Ablation Study II	Rob+Jigsaw (LR)	22.1/47.2	29.3/57.0	36.1/66.3	41.0/71.1	44.5/74.1
	Rob+Self-T (LR)	16.3/41.5	24.4/52.7	33.1/64.1	39.9/69.2	42.6/72.5
	Self-T+Jigsaw (LR)	15.6/40.3	23.6/51.8	32.0/63.5	39.1/68.5	41.9/71.8
	Rob+Jigsaw (SVM)	17.9/37.0	23.9/48.3	29.5/58.8	35.1/65.7	40.3/68.8
	Rob+Self-T (SVM)	15.0/35.5	21.8/47.5	29.9/59.2	34.6/65.4	39.8/68.3
	Self-T+Jigsaw (SVM)	13.1/31.9	19.0/44.3	28.3/57.2	33.0/64.3	39.1/67.2

Table 1: **Top-1 / Top-5 results of Imagenet1K (ResNet-10).** *: our methods. $^\sharp$: results from (Hariharan and Girshick 2017). $^{+}$: Jigsaw is applied in D_{base} rather than U_{novel} (from D_{novel}).

Results on ImageNet1k Challenge Dataset

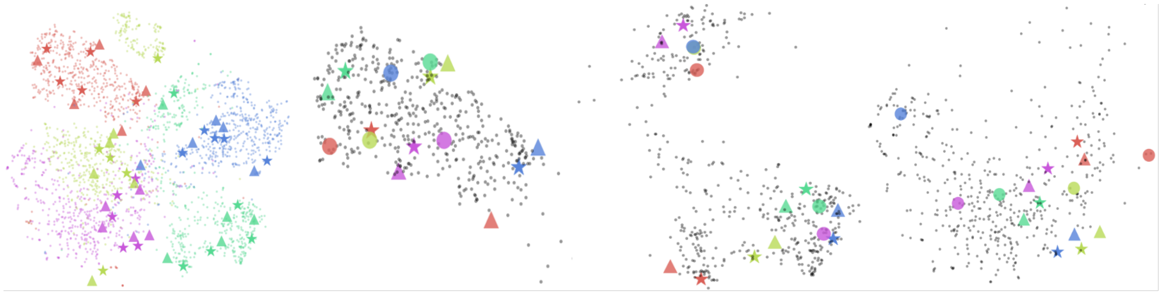
Setup. To make a comparison to (Hariharan and Girshick 2017), we choose both ResNet-10 and ResNet-50 residual network (He et al. 2015) as the base networks to train $f\left(\mathbf{I};\theta\right)$. For both networks, we use SGD to train networks which gets converged in 300 epochs: the learning rate is set to 1×10^{-1} and degraded by 10 every 30 epochs with the batch size 128. We augment each training instance up to five instances and train for 10 epochs.In fine-tuning, each probe image $\mathbf{I}_i$ helps produce 10 synthesize image $\widetilde{\mathbf{I}}_i$ and we trained for 10 epochs. The learning rate of the last layer and the other layers are set to 1×10^{-1}, 1×10^{-2} respectively in our experiments.

Baselines and competitors. As the naive baselines, we directly use C_{base} to train $f\left(\mathbf{I};\theta\right)$ which serves as the feature extractor to extract feature $\mathbf{x}$ of the image $\mathbf{I}$; and further train $g\left(\mathbf{x};\eta\right)$. The $g\left(\mathbf{x};\eta\right)$ can be used as Support Vector Machine (SVM), Logistic Regression (LR) or Neural Network of 1 fully connected layer and 1 Softmax classification layer (Softmax). In our experiments, SVM and LR perform consistently better than Softmax, and thus are further used in the ablation study. We also compare with recent models, such as Model Regression (Wang and Hebert 2016), Matching Network (Vinyals et al. 2016), Generation SGM (Hariharan and Girshick 2017). The standard data augmentation methods are also compared here: "Flipping": the same input image is flipped from left to right; "Gaussian Noise": we add Gaussian noise $\mathcal{N}\left(0,10\right)$ to each pixel of the input image; "Gaussian Noise (feature level)": the Gaussian noise $\mathcal{N}\left(0,0.3\right)$ is also added to each dimension of ResNet-18 feature extracted of each image. To make fair comparisons, all the augment methods utilize the SVM classifier as the one-shot classifier.

Variants of our framework. An ablation study is used to evaluate these three components of our framework: (1) Learning robust feature extractor ("Rob"): we only learn robust feature extractor, and directly apply the feature extractor to do one-shot classification on C_{novel}; (2) Self-training scheme ("Self-T"): we only use D_{base} to train $f\left(\mathbf{I};\theta\right)$; and on C_{novel}, we apply the self-training method (Chuck Rosenberg 2005) to update the classifier $g\left(\mathbf{x};\eta\right)$; (3) Jigsaw data augmentation ("Jigsaw"): we only use D_{base} to train $f\left(\mathbf{I};\theta\right)$; and we randomly select some unlabeled images as the gallery images in Jigsaw augmentation to get $\widetilde{D}_{novel}$ and $\widetilde{g}\left(\mathbf{x};\eta\right)$.

Results. The results of using ResNet-10 and ResNet-50 networks are compared in Tab. 1 and Tab. 2 respectively. We highlight several valuable points of the experiments.

(1) Our framework – Self-Jig (SVM) can achieve the best performance, significantly outperforming all the baselines and competitors. In particular, on Top-1 accuracy, the results of Self-Jig (SVM) are improved 8% over the SVM baseline. This validates the effectiveness of our self-training Jigsaw augmentation method.



(a)t-SNE visualization (b)Red class (c)Green class (d)Purple class

Figure 2: t-SNE Visualization. Stars, Circles and Triangles represent the labeled probe, unlabeled gallery, and synthesized image produced by our framework.

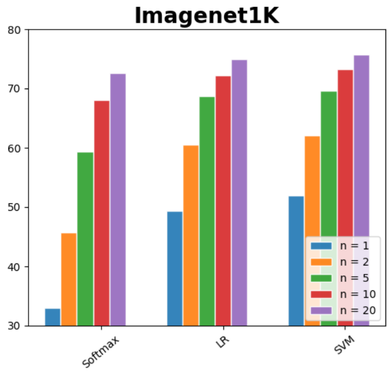


Figure 3: Top 5 accuracy(%) on Imagenet1K (Resnet-10).

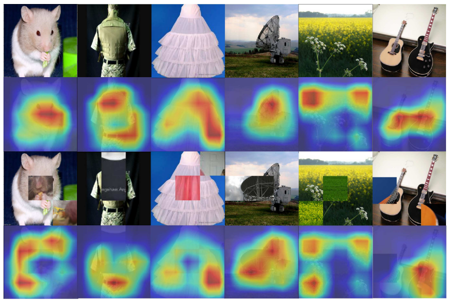


Figure 4: Class activation of probe and synthesized images.

Methods	n=1	2	5	10
Softmax	–/28.2	–/51.0	–/71.0	–/78.4
SVM	20.1/41.6	29.4/57.7	42.6/72.8	49.9/79.1
LR	22.9/47.9	32.3/61.3	44.3/73.6	50.9/78.8
Generation SGM	–/47.3	–/60.9	–/73.7	–/79.5
Self-Jig (SVM)	30.3/59.7	39.7/70.9	48.7/**78.7**	51.1/**80.3**
Self-Jig (LR)	32.8/**62.8**	41.2/**71.3**	49.2/77.9	51.5/79.3
Robust (SVM)	19.2/40.7	28.3/56.6	41.4/71.9	48.7/77.9
Robust (LR)	21.8/46.5	31.2/60.2	43.3/72.6	50.7/77.5

Table 2: **Top-1 / Top-5 accuracy on Imagenet1K Challenge Dataset (ResNet-50).** Self-Jig (LR) indicates that LR classifier is used as $g(\mathbf{I};\eta)$. Self-T: self-training; Jigsaw: Jigsaw Augmentation. Rob: Robust feature extractor. n indicates the number of training instances per class.

(2) On Top-5 accuracy, our Self-Jig (SVM) can beat the best competitors – Matching Network (Vinyals et al. 2016) by more than 10% in Tab. 1. The same conclusion still holds when we use the ResNet-50 in Tab. 2. We argue that much of our success and improvement comes from a more discriminative augmented training set $\widetilde{D}_{novel}$ by self-training Jigsaw augmentation method, rather than from the other factors, since the Self-Jig (SVM) can beat the corresponding baselines and competitors by a large margin both in Tab. 1 and Tab. 2.

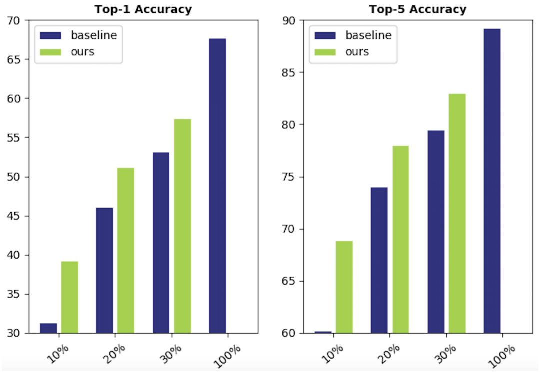
(3) We found that due to the small number of training instances on C_{novel} , the Softmax classifier has lower recognition accuracy than SVM and LR. Fig.3 clearly visualize the performances of three different classifiers when changing the number of training instances.

We further extend the experiments in few-shot settings by increasing the training instances from 1 to 20 as shown in Tab. 1 and Tab. 2 . We found that our methods can still beat the other baselines and competitors, due to the fact that our augmented training instance set can help train the classifier on C_{novel}. We also observe that the improvements achieved by our self-training Jigsaw augmentation method tend to somewhat diminish as the number of training instances substantially grows from 1, 2, 5 to 20 shots. This also makes sense. With sufficient training instances, the classifier $g(\mathbf{I};\eta)$ can be better learned and the effects of augmenting labeled images from the unlabeled set U_{self}, become less pronounced.

Ablation study. In Ablation Study I in Tab. 1, we evaluate the variants of our model. Here when we use Jigsaw without self-T, means we synthesized new images from the probe and randomly selected gallery images. We note that, (1) almost all the results of using each single component get degraded than the naive baselines. (2) Self-T (SVM) and Self-T (LR) are the naive semi-supervised baselines by the self-training method only. They add the most confidence instances from U_{novel} to update the corresponding classifier $g(\mathbf{x};\eta)$. Only in one-shot learning ($n = 1$), Self-T (SVM) can get slightly improved over SVM method, due to discriminative information learned from unlabeled data. Critically, this shows that the proposed methodology is a single unified framework; and the significant improvement of Self-Jig comes from the integration of all three components, rather

		“Self-T”				
		0	1	2	3	4
“Rob”	0	52.73	51.76	50.12	49.34	48.02
	1	52.64	55.99	56.24	55.79	55.55
	2	52.66	56.67	56.99	56.51	56.01
	3	52.59	57.45	57.78	57.66	57.22
	4	52.55	57.76	**58.45**	57.01	56.45
	5	52.23	56.06	56.71	56.36	56.21
	6	51.79	53.51	54.12	53.89	53.25
	7	51.6	53.21	53.41	52.97	52.76
	8	51.44	52.77	52.83	52.41	52.18

(a) Replacing blocks



(b) Augmentation in SSL

Figure 5: (a)Each row/column is corresponding to the number of replaced blocks in the “Rob”/”Self-T” component individually. The 1-shot classification accuracy on C_{novel} are reported. (b) Our Self-Jig is used in semi-supervised learning. The X-axis indicates the number of percentage of training instances used. Y-axis denotes the classification accuracy.

than each individual component.

In Ablation Study II in Tab. 1, the combinations of two components are compared. Interestingly, (1) we found that using two components can indeed help improve the performance over the corresponding baselines. For example, the “Rob+Jigsaw (LR)” and “Rob+Jigsaw (SVM)” have better classification accuracy over the LR and SVM baselines, but lower results than our Self-Jig (LR) and Self-Jig (SVM). This actually demonstrates that the importance of “self-training” component in selecting the U_{self} for data augmentation. (2) Additionally, the SVM classifier is more sensitive to the quality of training instances than LR. With the augmented data $\widetilde{D}_{novel}$ by our self-training Jigsaw method, our Self-Jig (SVM) have better performance than Self-Jig (LR). In contrast, Ablation study I and Ablation Study II show that the baseline methods by SVM classifiers are significantly lower than those corresponding variants by the LR classifier, due to the small number or low augmented data quality of training instances.

Without unlabeled data. Our framework still works without access to the extra unlabeled data. We could sample instances from D_{base} to serve as the gallery images. In this case, the Jigsaw Augmentation is conducted between images from different categories but likely to have a similar appearance. As shown in Tab. 1, we observed marginally improved (Self-Jig(SVM)$^{+}$,Self-Jig(LR)$^{+}$) over the naive baseline (SVM,LR).

Results on *mini*ImageNet

Setup. We employ ResNet-18 structure as the feature extractor $f\left(\mathbf{I};\theta\right)$. By default, we use the same hyperparameter and experimental settings as we train the network in ImageNet1k challenge dataset.

Competitors. As in Tab. 3, we mainly compare three groups of the competitors: Meta-learning algorithms, such as MAML (Finn, Abbeel, and Levine 2017), Meta-SGD (Li et al. 2017), DEML+Meta-SGD (Zhou, Wu, and Li 2018), META-LEARN LSTM (Ravi and Larochelle 2017) and Meta-Net (Munkhdalai and Yu 2017); Metric-learning algorithms, such as Matching Nets (Vinyals et al. 2016), PROTO-NET (Snell, Swersky, and Zemeln 2017), RELATION NET (Sung et al. 2018), and MACO (Hilliard et al. 2018)) ; and other semi-supervised methods, including Semi-supervised PROTO-NET (S.S. PROTO-NET) (Ren et al. 2018), Ladder Network (Rasmus et al. 2015), Graph Neural Networks(Garcia and Bruna 2018), Transductive Propagation Network(Liu et al. 2018), Semi-supervised Resnet PN(Boney and Ilin 2017). In particular, to make a fair comparison, we implement the ladder network by using ResNet-18 architectures and the Graph Neural networks under same settings. Random Erase (Zhong et al. 2017) is the method that randomly erases the images. We implement the method (Zhong et al. 2017) by ResNet-18 with SVM classifier.

Results. As shown in Tab. 3(a), our Self-Jig (SVM) achieves the best performance in the 1-shot and 5-shot classification settings. This validates the effectiveness of our framework in using the unlabeled images in data augmentation. Furthermore, we split the proposed framework, and evaluate each component/the combination of any two components in Tab. 3(b). We have two conclusive results: (1) Our frameworks, *i.e.*, Self-Jig (SVM), and Self-Jig (LR), get significantly improved over the corresponding baselines – SVM and LR, on 1-shot and 5-shot classification. This again shows the efficacy of our self-training Jigsaw data augmentation in helping one-shot classification. (2) The variants of only using each or any two components have no or very limited improvement over the corresponding baselines. That indicates that the proposed method is a unified single framework.

Compared with other semi-supervised few-shot learning methods. S.S.PROTO-NET(Ren et al. 2018), Ladder Network (Rasmus et al. 2015), GNN(Garcia and Bruna 2018), TPN(Liu et al. 2018), Resnet PN(Boney and Ilin 2017), Resnet PN^{+}(Boney and Ilin 2017) use 20,15,15,15,15,120 unlabeled images per category respectively, while our method use 15. As in Tab. 3(a), our frameworks easily overperform other methods in both 1-shot and 5-shot classification settings.

Visualization. We visualize the Class Activation Map (CAM) of images in Fig. 4. Specifically, the first and third rows are labeled probe and synthesized images produced by

Methods	*mini*ImageNet (%)	
	1-shot	5-shot
MAML (Finn, Abbeel, and Levine 2017)	48.70±1.84	63.11±0.92
Meta-SGD (Li et al. 2017)	50.47±1.87	64.03±0.94
DEML+Meta-SGD (Zhou, Wu, and Li 2018)	**58.49**±0.91	71.28±0.69
META-LEARN LSTM (Ravi and Larochelle 2017)	43.44±0.77	60.60±0.71
Meta-Net (Munkhdalai and Yu 2017)	49.21±0.96	–
Matching Nets (Vinyals et al. 2016)	43.56±0.84	55.31±0.73
PROTO-NET (Snell, Swersky, and Zemeln 2017)	49.42±0.78	68.20±0.66
RELATION NET (Sung et al. 2018)	57.02±0.92	71.07±0.69
MACO (Hilliard et al. 2018)	41.09±0.32	58.32±0.21
Random Erase (Zhong et al. 2017)	52.89±1.32	73.42±0.74
S.S. PROTO-NET (Ren et al. 2018)	50.41±0.31	64.39±0.24
Ladder Network (Rasmus et al. 2015)	52.82±1.49	73.37±0.79
GNN(Garcia and Bruna 2018)	53.46±0.48	68.70±0.81
TPN(Liu et al. 2018)	54.72±0.84	69.25±0.67
Resnet PN(Boney and Ilin 2017)	54.07±0.47	70.92±0.66
Resnet PN$^+$(Boney and Ilin 2017)	55.67±0.45	72.55±0.52
Self-Jig (SVM)*	**58.80**±1.36	**76.71**±0.72
Self-Jig (LR)*	**58.45**±1.22	**76.31**±0.64

(a) Main Results

Methods	*mini*ImageNet (%)	
	1-shot	5-shot
SVM	52.73±1.44	73.31±0.81
LR	53.06±1.37	73.48±0.86
Rob (SVM)	52.55±1.23	72.89±0.69
Rob (LR)	52.95±1.19	73.01±0.77
Self-T (SVM)	49.81±1.09	71.57±0.70
Self-T (LR)	51.89±1.52	72.58±0.87
Jigsaw (SVM)	54.76±1.25	73.25±0.88
Jigsaw (LR)	55.09±1.21	73.39±0.76
Rob+Self-T (SVM)	51.22±1.41	71.98±0.77
Rob+Self-T (LR)	50.02±1.11	71.02±0.65
Rob+Jigsaw (SVM)	55.54±1.29	73.40±0.69
Self-T+Jigsaw (SVM)	49.89±1.51	71.21±0.92
Self-Jig (SVM)*	**58.80**±1.36	**76.71**±0.72
Self-Jig (LR)*	**58.45**±1.22	**76.31**±0.64

(b) Ablation Study

Table 3: Results on *mini*ImageNet. The "±" indicates 95% confidence intervals over tasks. *: indicates our method.

our framework, respectively. The second and fourth rows are the activation map computed from the first and third rows by (Zhou et al. 2015) individually. In particular, the highest predicted class scores of each image is projected back to highlight its class–specific discriminative regions. We can show that if the replaced blocks of the synthesized images are highly related to its class label, the class-specific region would be enlarged, such as the "radar" image in the fourth column; otherwise, this region will be reduced.

To give more insights we visualize five classes in Fig. 2(a). The Stars, Circles, and Triangles represent the labeled probe, the unlabeled gallery, and the synthesized image generated by our framework. The instances of each class are denoted by the same color. We can see in Fig. 2(a) that most of our synthesized images (*i.e.*, Triangles) are still in the same class manifold. This explains why our synthesized images can help a lot in one-shot classification. We show the instances distribution of the red (Fig. 2(b)), green (Fig. 2(c)) and purple class (Fig. 2(d)) of Fig. 2(a). Each pair of labeled probe $\mathbf{I}_i$, unlabeled gallery $\mathbf{I}_u$ and synthesized image $\widetilde{\mathbf{I}}_i$ is drawn with the same color. We can observe that when the gallery image is on the same manifold as probe image, the synthesized image is also on the manifold, *e.g.*, the blue pair in Green class in Fig. 2(c) .

Ablation Study

We conduct extensive further ablation study on *mini*ImageNet to reveal the insights of our framework. In particular, we answer several questions as follows,

Replacing blocks. We use Jigsaw augmentation to fine-tune the base network with $m \leq 4$; The "Self-T" component employs the Jigsaw augmentation ($m \leq 2$) in a self-training manner. We vary the parameter m in our framework and compare the results in Tab. 5. We can see that by changing the number of replaced blocks, there may be a slight variance in one-shot classification accuracy; but, our experimental conclusions still hold.

The Number of synthesized images. We choose to generate 10 synthesized images in our experiments. We also compare the results of generating 0, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000 synthesized images, while all the other parameters are kept the same. Thus the corresponding one-shot classification accuracies are 52.55%, 55.84%, 57.14%, 57.9%, 58.2%, 58.02%, 57.96%, 58.42%, 58.24%, 58.11% and 58.19%, respectively. Thus, we found that changing this parameter may lead to a slight variance of the final performance. But our final results are still significantly better than the baselines.

Performance in Semi-Supervised Learning (SSL). Our Self-Jig is tested in the SSL setting on C_{base} classes: we use D_{base} as training data, and have unlabeled images to help learn the SSL classifier. As in Tab. 5(b), we train the ResNet-18 classifier on C_{base}, and conduct the Self-Jig method by using the unlabeled images. The results show that our framework can improve classification performance in such a setting, and we will take it as a future work to fully explore these cases.

Conclusion

This work proposes a self-training Jigsaw data augmentation method for one-shot learning. Extensive experiments show the efficacy of our framework in synthesizing new instances to boost the recognition performance.

Acknowledgments

This work was supported in part by National Key R&D Program of China (#2017YFC0803700), NSF China (#U1611461, #61622204, #61572138), and STCSM (#16JC1420400, #17JC1401600). Dr. Yanwei Fu is the corresponding authour.

References

Boney, R., and Ilin, A. 2017. Semi-supervised few-shot learning with prototypical networks. *CoRR* abs/1711.10856.

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

Chen, Z.; Fu, Y.; Zhang, Y.; Jiang, Y.-G.; Xue, X.; and Sigal, L. 2018. Semantic Feature Augmentation in Few-shot Learning. *ArXiv e-prints*.

Chuck Rosenberg, Martial Hebert, H. S. 2005. Semi-supervised self-training of object detection models. In *IEEE workshop on Motion and Video Computing*.

Desolneux, A.; Moisan, L.; and Morel, J.-M. 2004. Gestalt theory and computer vision. In *Theory and Decision Library A:*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Garcia, V., and Bruna, J. 2018. Few-Shot Learning with Graph Neural Networks. In *ICLR*.

Guttenberg, N., and Kanai, R. 2018. Learning to generate classifiers. *ArXiv e-prints*.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. In *CVPR*.

Hilliard, N.; Phillips, L.; Howland, S.; Yankov, A.; Corley, C. D.; and Hodas, N. O. 2018. Few-Shot Learning with Metric-Agnostic Conditional Embeddings. *ArXiv e-prints*.

Inoue, H. 2018. Data Augmentation by Pairing Samples for Images Classification. *ArXiv e-prints*.

Koch, G.; Zemel, R.; and Salakhutdinov, R. 2015. Siamese neural networks for one-shot image recognition. In *ICML – Deep Learning Workshok*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Laine, S., and Aila, T. 2016. Temporal Ensembling for Semi-Supervised Learning.

Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017. Meta-sgd: Learning to learn quickly for few shot learning. In *arxiv:1707.09835*.

Liu, Y.; Lee, J.; Park, M.; Kim, S.; and Yang, Y. 2018. Transductive propagation network for few-shot learning. *CoRR* abs/1805.10002.

MichaelMcCloskey, and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation*.

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

Noroozi, M., and Favaro, P. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*.

Rasmus, A.; Valpola, H.; Honkala, M.; Berglund, M.; and Raiko, T. 2015. Semi-supervised learning with ladder networks. In *NIPS*.

Ravi, S., and Larochelle, H. 2017. Optimization as a model for few-shot learning. In *ICLR*.

Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J. B.; Larochelle, H.; and Zemel, R. S. 2018. Meta-learning for semi-supervised few-shot classification. In *Proceedings of 6th International Conference on Learning Representations ICLR*.

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

Snell, J.; Swersky, K.; and Zemeln, R. S. 2017. Prototypical networks for few-shot learning. In *NIPS*.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching networks for one shot learning. In *NIPS*.

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

Wang, P.; Liu, L.; Shen, C.; Huang, Z.; Hengel, A.; and Tao Shen, H. 2017. Multi-attention network for one shot learning. In *CVPR*, 6212–6220.

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

Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, 189–196. Association for Computational Linguistics.

Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *ECCV*.

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2017. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2015. Learning deep features for discriminative localization. *CVPR*.

Zhou, F.; Wu, B.; and Li, Z. 2018. Deep meta-learning: Learning to learn in the concept space. In *arxiv:1802.03596*.