# An Embarrassingly Simple Baseline to One-shot Learning

Chen Liu[1*] Chengming Xu[1*] Yikai Wang[1,4] Li Zhang[2] Yanwei Fu[1,3,4†]
[1]School of Data Science, Fudan University
[2]Department of Engineering Science, University of Oxford
[3]MOE Frontiers Center for Brain Science, Fudan University
[4]Shanghai Key Lab of Intelligent Information Processing, Fudan University
{chenliu18, cmxu18, yikaiwang19, yanweifu}@fudan.edu.cn, lz@robots.ox.ac.uk

## Abstract

*In this paper, we propose an embarrassingly simple approach for one-shot learning. Our insight is that the one-shot tasks have domain gap to the network pretrained tasks and thus some features from the pretrained network are not relevant, or harmful to the specific one-shot task. Therefore, we propose to directly prune the features from the pretrained network for a specific one-shot task rather than update it via an optimized scheme with complex network structure. Without bells and whistles, our simple yet effective method achieves leading performances on miniImageNet (60.63%) and tieredImageNet (69.02%) for 5-way one-shot setting. The best trial can hit to 66.83% on miniImageNet and 74.04% on tieredImageNet, establishing a new state-of-the-art. We strongly advocate that our method can serve as a strong baseline for one-shot learning. The codes and trained models will be released at* http://github.com/corwinliu9669/embarrassingly-simple-baseline.

## 1. Introduction

Deep learning (DL) has achieved great success in many computer vision tasks such as image recognition [8, 24, 13], image generation [7, 2, 34] and segmentation [33, 9, 31, 17, 28]. Despite the achievement, deep learning based approached still struggle to obtain the ability to adapt to a totally unseen environment. For instance, they can hardly recognize images that does not share training categories with no-further information [32] or limited training example provided [26].

One-shot learning [26, 29] aims to solve this problem, assuming only the existence of one or a few labeled data for each category. One-shot learning, as a special case, restricts the number of label data for each category to 1. Such an assumption breaks one of the necessary conditions of traditional neural network models and requires the capacity of rapidly learning from few-shot instances without overfitting. Though such capacity seems hard for machine learning algorithms, it is observed that humans have one-shot learning ability. That is, one can handle a new concept with only one or a few corresponding instances are provided. For example, child can recognize the concept of "panda" after seeing several pandas' pictures. Humans' few-shot learning capacity comes from the relative learned knowledge as well as rapid learning. To simulate human's learning process, one-shot learning has access to a base class dataset with a large amount of labeled images to learn the necessary knowledge. Then we focus on learning from few-shot labeled instances on a novel class dataset whose classes are disjoint (but more or less relevant) from the base dataset.

One-shot learning has been studied for a long time, [5] proposes to tackle this problem via a probabilistic framework. With the revolution of deep learning, a resurgence of research in one-shot learning comes. At the beginning stage, a general procedure is to train a network using multi-class classification loss as an extractor and classify images from novel class using this extractor. For instance [12] utilizes a siamese structure to construct a similarity evaluation network. However, with the rise of meta-learning [22], this branch seems to be overlooked. For meta-learning, the key idea lies in simulating the test phase so that the trained network can have a better ability to adapt to novel class data. Follow [22], plenty of great works have been proposed such as MAML [6]. Afterwards more and more works tend to combine meta-learning and metric learning. ProtoNet [25] selects a meta-learning strategy and utilize a nearest neighbour classifier. Based on ProtoNet, TADAM [21] and RelationNet [26] try to improve previous method in terms of metric. In spite of the success of these methods, only using meta-learning leads to a drawback as mentioned in [3]. Meta-learning based method tends to focus on local features

*Equal contribution.
†Corresponding author.

which can cause a problem called mean shift. Additionally, we notice that some work [10, 18] imports the global label as additional loss which has significant overhead over purely using meta-leaning. These phenomenon enlightens us to revisit the pretrain-based method.

By revisiting the pretrain-based method, we propose a very simple baseline. As first stage, we utilize the base class data to train network. The pretrained network is then frozen. During test phase, we try different classifiers such as logistic regression (LR), support vector machine (SVM) and nearest neighbour (NN) for classification. Furthermore, we compare these procedures with ones in machine learning. We find that one important step is overlooked. To guarantee that features have a similar scale, we should normalize the output of pretrained network. Empirically we observe significant enhancement via normalization.

Furthermore, inspired by human learning process, we then propose an inspiring way to select the features extracted by the deep neural networks based on the logic for humans when recognizing objects. In fact, we know a lot of features such as shape, color, size. But when we face a specific problem such as distinguish between lemon and lime, we will not use all these features. Instead, we know that one significant difference is that they have different colors, thus dropping others and focusing on color. This intuition can be also useful for one-shot learning. Currently we find that some meta-learning based methods attempt to enhance the one-shot learning performance through modifying the embedding or finding some combination of them. In fact the embedding of deep network always has a high dimension, we argue that not all of them are useful for classification of novel class data. The network is trained based on base class data, so embedding may contain some domain-specific information that degenerates one-shot learning. According to this intuition, we try to investigate the contribution of these embedding dimensions via a random pruning strategy. Interestingly we observe that by cutting negative dimensions, a large boost can be achieved.

To summarize, our contributions are two-fold: (i) We revisit one-shot learning using a common supervised training paradigm. Without the assistance of meta-learning, we can get accuracy of 60.63% for *mini*ImageNet and 69.02% for *tiered*ImageNet under one-shot 5-way setting. (ii) We explore the redundancy of the output feature of pretrained network, and find an interesting phenomenon that offers us a new insight for improving one-shot learning. The best trials can achieve 66.83% on *mini*ImageNet and 74.04% on *tiered*ImageNet.

# 2. Methodology

## 2.1. Problem setup

Different setups are used in train and test by us. When training, we use many-shot recognition manner. Denote the images and annotations as $X$ and $Y$, respectively.

**Training.** During training stage, we have $\{C\}_{i=1}^{N}$ categories with sufficient annotations. For each class $C_i$ from the $N$ categories, we have $n_i$ image-label pairs as $\{(x_j^{C_i}, y_j^{C_i})\}_{j=1}^{n_i}$ for each class.

**Testing.** For testing phase, we follow a 5-way 1-shot setting. Suppose that we have $\{K\}_{i=1}^{N_t}$ categories. During test phase, we sample support set and query set from these data. First of all we sample 5 classes from the $N_t$ classes. Then for each of them we sample 1 image-annotation pair for $K_i$ as $(x^{K_i}, y^{K_i})$ as the support set. And for the query set, we sample $m$ samples that are assumed to have no annotation from them as $\{x_j^{K_i}\}_{j=1}^{m}$, here we set $m = 15$ as common.

## 2.2. Baseline

For our baseline, we train a ResNet-12[19]. Here we denote the layer before fc layer as $\phi(\cdot)$, it is composed of convolution layers, pooling layers and maxpooling layers while it ends with a global average pooling layer. Besides we denote the fc layer as $\Psi(\cdot)$ which acts as the classifier. The output embedding of the network is $\phi(x)$ and the logit is $\Psi(\phi(x))$. Here $\Psi(\cdot)$ has a softmax activation function.

For training phase, the output feature is a 512-dimension vector after the global average pooling. For this stage, we apply a cross-entropy loss which is widely used in image recognition [8]. We denote the loss function as $L$. For a batch $B$ of samples, the loss term is $\frac{1}{B}\sum_{i=1}^{B} L(\Psi(\phi(x)), y_i)$. After training the network, we remove the final fc layer for specific classification and the weights of $\phi(\cdot)$ are kept unchanged.

When testing, we only use extractor $\phi(\cdot)$. We adopt three different classifiers: LR, SVM and NN. For LR and SVM, we use the extractor to get the processed feature as 512-dimension vector. For the support set, we use them to train classifier. And the trained classifier is used to predict query set. For the NN classifier, we use support set to get template and classify query samples according to the euclidean distance between them and the template.

Additionally, we find that for a standard machine learning paradigm, normalization is a very important step. Though we use batch normalization [11], this operation is used to alleviate the problem of mean shift and get a better loss landscape. The scale of these features may be not suitable. For deep learning, in accompany with a fc layer can solve the problem by itself. Here we decide to add a normalization operation. For a output embedding $\phi(x)$, we

calculate its $l_2$ norm as $\|\phi(x)\|_2$, and normalize the embedding as $\frac{\phi(x)}{\|\phi(x)\|_2}$. The normalized features are then fed into classifiers during testing phase.

### 2.3. Random pruning

As is known to all, the model used for image recognition tends to have a relatively large capacity. Pruning some of them even leads to little changes in terms of model performance. Here we want to judge whether all of them are important for one-shot learning tasks. Inspired by SNIP [15], we try to explore whether there exists a subset of the embedding that is suitable for every specific task. In detail, for each task, we make several trials to find a $0-1$ mask. Suppose that we have a rate $p \in (0,1)$. For each mask $B$, we want that $p$ of the mask elements are 0 and the other elements are 1. After getting the mask, we multiply it to the embedding as $\phi'(x) = \phi(x)B$, here the operation is element-wise. Normalization is then applied to $\phi'(x)$. For each task, we pick the mask with the best accuracy. Note that we here just want to explore the property, so no training or other skills are used. A random trial is applied.

## 3. Experiments

### 3.1. Settings

**Dataset.** We perform experiments on two widely-used datasets, *mini*ImageNet and *tiered*ImageNet. Specifically there are 60000 images in *mini*ImageNet, belonging to 100 categories. 64 of them are for training, while 16 and 20 of them are used for validation and testing respectively. Each of the categories owns 600 images. The split follows the manner in [22]. *tiered*ImageNet dataset contains 779,165 images in all. There are 608 classes in total, where 351 classes are used for training, 97 for validation and 160 for testing. All these images are resized to 84×84. We apply random horizontal flip as augmentation when training.

**Implementation details.** We train the network using SGD [1] with initial learning rate of 0.1 which decays by 0.1 every 30 epochs. The whole training takes 100 epochs. The momentum term is set to be 0.9 and 5e-4 for weight decay. We validate the network using NN classification accuracy of validation set. During testing, we follow a 5-way 1-shot setting, and sample 15 queries for each class. We test the result for 1200 episodes.

### 3.2. Results

**Comparison with alternatives.** For comparison we pick some classical meta-learning based or metric-learning based method : ProtoNet[25], RelationNet[26], MatchingNet[27] and MAML [6] as mentioned in [4]. We directly use the results in [4]. The result is shown in Table 1.

| Method | *mini*ImageNet |
|---|---|
| MatchingNet | 54.49 |
| ProtoNet | 51.98 |
| MAML | 54.69 |
| RelationNet | 52.19 |
| Our Baseline | 60.63 |

Table 1. Comparison on *mini*ImageNet.

We further pick some newly-proposed method such as TADAM [21], SNAIL [19], CAN [10], MetaOptNet[14] and LEO [23]. We show results for *mini*ImageNet and *tiered*ImageNet in Table 2. Our baseline does have significant performance gap compared with these methods.

| Method | *mini*ImageNet | *tiered*ImageNet |
|---|---|---|
| TADAM [21] | 54.49 | – |
| SNAIL [19] | 55.71 | – |
| TapNet [30] | 61.65 | – |
| DC [18] | 62.53 | – |
| CAN [10] | 63.85 | 69.89 |
| MetaOptNet [14] | 62.64 | 65.99 |
| LEO [23] | 61.76 | 66.33 |
| CTM [16] | 64.12 | 68.41 |
| Our Baseline | 60.63 | 69.02 |
| Best Prune Trial | 66.83 | 74.04 |

Table 2. Comparison on *mini*ImageNet and *tiered*ImageNet with better methods.

**Effect of pretraining** We try different ways to choose pretrain model, whose results are shown in Tab. 3. In addition to that using validation set is better than using part of training set, there exists a gap between using many-shot metric and one-shot metric.

**Effect of normalization.** To verify the effect of normalization, we try to compare the situation with and without normalization. The result is shown in Table 4. It is clear that Logistic Regression performs the best. Besides, with the assistance of normalization, the performance boosts greatly. For NN the 5-way 1-shot on *mini*ImageNet increases from 55.59% to 60.06%. Besides, we find that before using normalization, NN has worse performance than other two counterparts. When normalization is added, their performance are at the same level. It is quite in accordance with our intuition. Based on these few information, there should be no significant difference for these classification method. Among these methods, LR outperforms other variants which is in accordance with [20].

| Method | *mini*ImageNet |
|---|---|
| Metric 1 | 60.06 |
| Metric 2 | 59.51 |
| Metric 3 | 60.59 |

Table 3. This table shows result using different metrics when selecting best moedl. Metric 1 uses validation part of train classes to get classification accuracy. Metric 2 uses validation part of train classes to get meta test accuracy. Metric 3 uses validation classes to get meta test accuracy.

| Method | *mini*ImageNet | *tiered*ImageNet |
|---|---|---|
| NN | 55.59 | 58.45 |
| SVM | 55.62 | 58.69 |
| LR | 57.41 | 63.15 |
| NN + normalization | 60.06 | 67.30 |
| SVM+ normalization | 59.95 | 67.27 |
| LR+ normalization | 60.63 | 69.03 |

Table 4. Results of different variants of our baseline.

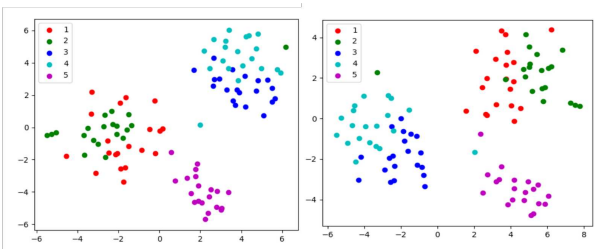

Figure 1. The left figure is tsne plot for the embedding of Res-12 and the right one shows the plot of a picked pruning. 50% of the out embedding dimensions are set to be zero. We pick 5 classes and 20 instances for each of them. It is obvious that the embedding space becomes more separable with a good pruning strategy.

**Discussion of the dataset.** We find that on *mini*ImageNet, our simple baseline is outperformed by some newly-proposed method. However, when it comes to *tiered*ImageNet, the gap reduces sharply as shown in Table 2. We wonder that when we discuss the one-shot learning problem, what size of base class data is suitable. For different settings, we should prefer different strategies.

### 3.3. Results of random pruning

As shown in Table 2, we pick a best trial for one-shot learning. And we find that if we can find the right mask for each task, the performance can be enhanced greatly from 60.63 to 66.83 for *mini*ImageNet. A right way to find this kind of mask can be very helpful. For better understanding, we add a visualization result shown in Figure 1. For an effective trial, we find that the red class and green class become more separable after pruning. It is in accordance with the increase of accuracy.

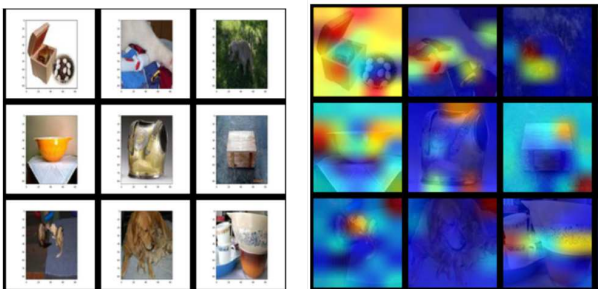

Figure 2. This figure shows the visualization result for some picked images. The left figure shows the input images, while the right one is the visualization of one of the pruned channels correspondingly. We find that by only pruning this channel can improve the result by 1 point. It is obvious that this channel puts emphasis on some wrong areas which can explain the mechanism to some extent.

To verify whether the boost is caused by reduction in redundancy, we compare the best trial with PCA. The results are illustrated in Table. 5. It is clear that using PCA does improve the performance slightly, but the extra enhancement is overwhelming. We think that this phenomenon is valuable for deeper investigation.

| Method | *mini*ImageNet |
|---|---|
| Our Baseline | 60.63 |
| PCA | 61.40 |
| Best Prune Trial | 66.83 |

Table 5. This table shows comparisons between PCA and best trial

Furthermore, we combine our prune method with MetaOptNet as shown in Table 6. For both *mini*ImageNet and *tiered*ImageNet, the best prune trial can improve 1-shot learning by a significant margin. It indicates that right prune mask exists not only for the simple baseline.

| Method | *mini*ImageNet | *tiered*ImageNet |
|---|---|---|
| MetaOptnet | 62.64 | 65.99 |
| Best Prune Trial | 66.36 | 69.71 |

Table 6. This table shows prune results on MetaOptNet.

## 4. Conclusion

In this paper, we propose an embarrassingly simple approach for one-shot learning. Without complex network structure or optimization scheme, we directly prune the features from the pretrained network for a specific one-shot recognition task. Extensive experiments show the effectiveness of our approach. Future work could leverage reinforcement learning to learn a better pruning mask.

## References


[1] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. 2010.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[3] Da Chen, Feng Mao, Mingli Song, Yuan He, Xiang Wu, Jinqiao Wang, Wenbin Li, Yongliang Yang, and Hui Xue. Class regularization: Improve few-shot image classification by reducing meta shift. *arXiv preprint arXiv:1912.08395*, 2019.

[4] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019.

[5] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *TPAMI*, 2006.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[9] Qinbin Hou, Li Zhang, Ming-Ming Cheng, and Jiashi Feng. Strip pooling: Rethinking spatial pooling for scene parsing. In *CVPR*, 2020.

[10] Ruibing Hou, Hong Chang, MA Bingpeng, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *NeurIPS*, 2019.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[12] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML workshops*, 2015.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[14] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019.

[15] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018.

[16] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *CVPR*, 2019.

[17] Xiangtai Li, Li Zhang, Ansheng You, Maoke Yang, Kuiyuan Yang, and Yunhai Tong. Global aggregation then local distribution in fully convolutional networks. In *BMVC*, 2019.

[18] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. Dense classification and implanting for few-shot learning. In *CVPR*, 2019.

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

[20] Andrew Y Ng and Michael I Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NeurIPS*, 2002.

[21] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018.

[22] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

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

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

[25] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

[26] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

[27] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016.

[28] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019.

[29] Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu. Instance credibility inference for few-shot learning. In *CVPR*, 2020.

[30] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. 2019.

[31] Li Zhang, Xiangtai Li, Anurag Arnab, Kuiyuan Yang, Yunhai Tong, and Philip HS Torr. Dual graph convolutional network for semantic segmentation. In *BMVC*, 2019.

[32] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2017.

[33] Li Zhang, Dan Xu, Anurag Arnab, and Philip HS Torr. Dynamic graph message passing network. In *CVPR*, 2020.

[34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.