Mutual Guidance: A New Approach to Train Auto-Encoder for Self-supervised Learning

Dou ZhaoPeng
2019310715
Department of Eletronic Engineering
Tsinghua University
Haidian District, Beijing 100084
dcp19@mails.tsinghua.edu.cn

Han Jian
2019211012
Department of Eletronic Engineering
Tsinghua University
Haidian District, Beijing 100084
hanj19@mails.tsinghua.edu.cn

Hu Miao
2019211000
Department of Eletronic Engineering
Tsinghua University
Haidian District, Beijing 100084
hum19@mails.tsinghua.edu.cn

Wang DaoHui
2019211025
Department of Eletronic Engineering
Tsinghua University
Haidian District, Beijing 100084
wang-dh19@mails.tsinghua.edu.cn

Xu JingTao
2019210964
Department of Eletronic Engineering
Tsinghua University
Haidian District, Beijing 100084
xjd19@mails.tsinghua.edu.cn

Abstract

Without human annotated supervision, Self-supervised learning aims to learn robust feature representations by mining supervision signals from the data itself. Existing efforts leverage other discriminating tasks where supervision signals are brought inherently by the images. But the existing task misalignment inhibits the performance improvement. In this work, we propose a novel Self-supervised learning pipeline, Mutual Guidance to Train Auto-Encoder (MGTA). MGTA trains two anto-encoder networks alternately where they learn from each other by accepting mutually generated pseudo labels. Here pseudo labels are predicted through feature clustering. Explicit category supervision signals are exploited in MGTA. As a result, the task misalignment could be mitigated. We intensively investigate the representations quality across different benchmarks including image classification and object detection tasks. Without bells and whistles, the proposed MGTA steadily outperforms the baseline by a large margin on different tasks, which provides a strong pipeline for Self-supervised learning.

1 Introduction

In recent years, deep learning method has been drawing more and more attention because of its powerful representations learning capacity. It achieves great success in many fields such as computer vision, natural language processing. However, most of its performance which exceeds human is based on supervised data-hungry learning paradigm which requires intensive manual labeling effort. Therefore, a large amount researches focus on unsupervised learning which learns representation from raw

Submitted to Advanced Machine Learning (Fall 2019).

data without annotations [1, 2]. Self-supervised learning is a popular form of unsupervised learning, defining pretext tasks to guide networks to learn feature representation for downstream tasks such as classification, object detection and so on. In self-supervised learning approaches, the supervised information comes from data itself such as predicting rotation [3] and relationship between image patches [4]. However, the results of these methods may collapse to trivial solutions by relying on chromatic aberration or texture instead of semantic information. In addition, inter-relations among images may provide informative knowledge for networks which is generally neglected. Based on the insight of mining the inter-relations among images, we propose a novel self-supervised learning pipeline, Mutual Guidance to Train Auto-Encoder (MGTA), to employ reconstruction loss to guide the learning of pixel-wise representation and clustering to mining the inter-relations among images with triplet loss. Besides, in order to avoid the network falling into local optimum, We build two independent subnetworks to learn feature representation and design mutual guidance between each other. Specifically, we use the clustering results from one subnetwork to train auto-encoder component in another subnetwork, avoiding misguidance by self bias. Our contributions are two-fold: First, we combine pixel-wise representation learning by reconstruction task and inter-relations mining by clustering. Second, we design a novel mutual guidance approach to avoid misguidance by self bias, which improves generalization of model. We test our method on image classification and object detection task as downstream tasks on Pascal VOC datasets and the proposed MGTA learn effective feature representation outperforming the baseline on different tasks.

2 Related Work

The self-supervised framework focus on supervising with only unlabeled data itself. It is a prominent paradigm that defines different pretext task without annotations for feature representations learning which will hopefully be informative for down stream "real" tasks. Self-supervised learning is drawing more and more attention in a wide range of applications.

2.1 Self-learning based on image transformation

An image can be transformed slightly such as translation, rotation and scaling, without losing semantic information and generate more different images. [5] propose an alternate strategy of directly modeling complex invariance of object features which has involved self-learning idea. Following researches focus more attention on learning invariance features from transformed images without any annotations. [6] propose Exemplar-CNN which samples at different locations on images of different scales transformed randomly by translation, rotation and scaling. All these transformed images compose a surrogate class which can be used as pseudo-labels. [3] follow the self-supervised paradigm and proposes a method to learn image representations by training network to recognize the geometric transformation applied to images. In order to recognize the rotation angle of an image, the network are forced to learn the features of both global semantic pattern and local texture pattern.

2.2 Self-learning based on image patches

Another self-learning pretext task is based on relationship between image patches. [4] defines the pretext task as predicting the relevant locations between two random image patches. The underlying hypothesis is that predicting relevant locations needs to understand scenes and objects relationship. Following this work, [7] design a "jigsaw puzzle" to predicts the order of a set of permutation image patches which complicates the pretext task. In order to accelerate training procedure, [8] employ graph convolution network and multi-agent deep reinforcement learning to achieves better sample efficiency and scalability. However, edge pattern and texture of image result in trivial solution of "jigsaw puzzle" method. Specifically, the network may "cheat" and bypass learning the desired semantic structure by finding incidental clues that reveal the location of a patch such chromatic aberration. Based on "jigsaw puzzle" method, [9] address some overt problems such as chromatic aberration and prevent problems with testing generalization on common self-supervised benchmarks tests by using different datasets. In order to overcomes limitations in designing and comparing different tasks, [10] decouples the structure of the self-supervised model from final task-specific fine-tuned model. In addition, [11] define a conserved scale characteristic by summation of image patches.

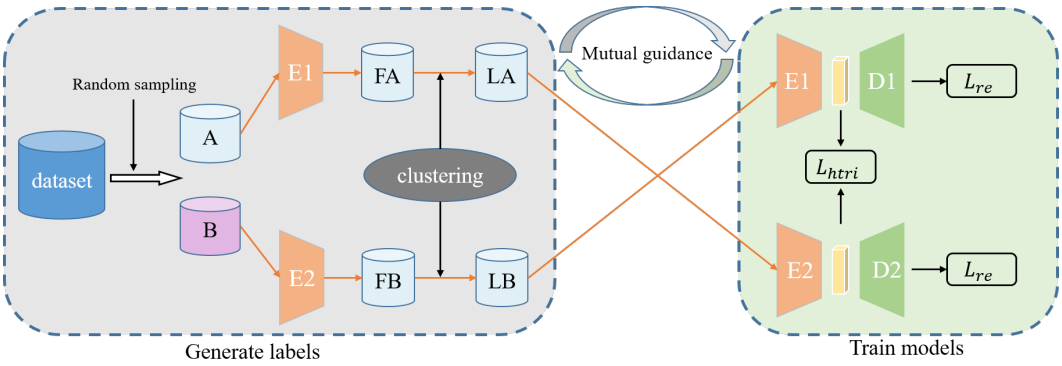

Figure 1: The framework of our MGTA; There are two Auto-Encoder structure in MGTA, denoted as $AE_1 = \{E_1, D_1\}$ and $AE_2 = \{E_2, D_2\}$. MGTA consists of three parts: generating labels, training models and mutual guidance. In the part of generating labels, the dataset is randomly divided into A and B. F_A and F_B represent the features of A and B extracted by E_1 and E_2, respectively. L_A and L_B represent A and B with the corresponding category labels generated by clustering. In the part of training models, L_A is used to train AE_2, while L_B is used to train AE_1. The hard triplet loss (L_{htri}) is adopted for features extracted by encoder E_1 and E_2. Moreover, the basic reconstruction loss (L_{re}) is still preserved. The clustering provides labels for training models, while the encoder E_1 and E_2 are used to extract features for clustering, which is mutual guidance.

2.3 Self-learning based on image generation

Besides, there are some other effective pretext tasks. [12] propose a network trained to generate the contents of an arbitrary image region conditioned on its surroundings, sharing similar idea with [13]. [14] build two disjoint sub-networks to predict one subset of the data channels from another to reconstruct data. Generative adversarial network is an effective method to learn the map from latent variables to complex data distribution. [15] propose DCGANs to learn a hierarchy of representations from object parts to scenes in both the generator and discriminator. Similarly, [16] introduce an encoder to learn map from input to latent variables contributing to feature learning.

Different from the work mentioned above, our method builds a reconstruction pretext task with encoder-decoder networks. Dissimilar with [17], our proposed method focus on feature representation in a low dimension latent space with guidance of relationship between similar images. Inspired by [18], we introduce clustering approach to assign pseudo-labels and train the network with triplet loss under the hypothesis that similar images are close to each other after being mapped to latent feature space.

3 Methods

In this section, we will introduce the framework of our proposed approach MGTA for self-supervised representation learning. As depicted in Figure 1, the basic network structure is Auto-Encoder. What's more, MGTA consists of three parts: (1) generating labels process (denoted as GLP), (2) training models process (denoted as TMP) and (3) mutual guidance.

3.1 Auto-Encoder network structure

An Auto-Enocder (AE) network is composed of two subnetworks: the encoder network $\mathcal{E}$ and the decoder network $\mathcal{D}$. We train the AE based on the proposed images reconstruction tasks. Given a sample X, the $\mathcal{E}$ provides a reduced representation F_X of it, with maximum information preservation, while the $\mathcal{D}$ need to retrieve X from F_X. That is, we want the output (denoted as $\hat{X}$) of $\mathcal{D}$ is similar to X as much as possible, which can be described as blow:

$$\hat{X} \approx X,\ X = \mathcal{D}(\mathcal{E}(X)) \tag{1}$$

Table 1: Architectures for encoder and decoder

layer name	encoder	decoder
conv1	7×7, 64, stride 2	Trans Conv $\begin{bmatrix} 1 \times 1, 2048 \\ 3 \times 3, 512 \\ 1 \times 1, 512 \end{bmatrix} \times 3$
conv2_x	3×3 max pool, stride 2 Conv $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	Trans Conv $\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 256 \\ 1 \times 1, 256 \end{bmatrix} \times 6$
conv3_x	Conv $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	Trans Conv $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 128 \\ 1 \times 1, 128 \end{bmatrix} \times 4$
conv4_x	Conv $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	Trans Conv $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 64 \\ 1 \times 1, 64 \end{bmatrix} \times 3$ interpolate scale factor=2
conv5_x	Conv $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	Trans Conv 8×8, 3, padding 3, stride 2

We adopt the resnet50 [19] without the last classifier layer as our encoder structure. The decoder uses a symmetrical structure, which replace the convolutional layer by the transposed convolutional layer. The structure of our AE is described in Table 1

3.2 Generating labels

Traditional unsupervised representation learning highly depends on a pretext assumption, and our MGTA is defined on top of the reconstruction power of learned features. However, these methods learn the spatial dependencies within the images, hence the inter-relations among the samples are neglected. Based on the observations that the features of samples belonging to the same classes span close-by while the features of samples belonging to different classes are far apart in a pleasurable embedding space, we utilize the clustering to generate labels for data points. Suppose we already have a superior feature extractor, then the distribution of features in the embedding space will also be outstanding, so that the result of clustering for features will be relatively accurate.

Clustering We adopt the K-means clustering algorithm to generate labels for training data points. However, according to the settings of self-supervised representation learning, the real value of K is unknown. The determination of the K value has always been the key to the K-means algorithm. Since K-means is an unsupervised learning method, there is no so-called "best" K value. However, from the characteristics of the data itself, within the category corresponding to the best K value, the intra-class distance should be minimized and the inter-class distance should be maximized. There are multiple indicators can be used to evaluate such characteristics, such as average contour coefficient, intra-class distance / inter-class distance, etc. We use the Calinski-Harabasz (CH) criterion to indicate whether the K is "best" or not. The CH can be calculated as blow:

$$CH(k) = \frac{tr(B_k)}{tr(W_k)} \frac{m-k}{k-1} \tag{2}$$

where m represents the number of data points, k represents the number of categories. B_k is the covariance matrix between categories, while W_k is the covariance matrix of data within categories. From the equation 2, we can know that the CH is higher, the K is better. Based on this idea, we can calculate the CH score at each K, and then select the K value at which the CH score is the largest.

3.3 Training models

After clustering, we can get the relative labels of training data points. Our approach is inspired by the combination of triplet loss and reconstruction loss. The former aims to utilize the inter-relations among the data points, while the latter aims to learn the spatial dependencies within the images.

3.3.1 The hard triplet loss

The triplet loss is widely used in image retrieval tasks such as face recognition, person re-identification and so on. It aims to restrict the features in the embedding space. For the anchor image x^a in the candidate triplet tuple $\{x_i^a, x_i^p, x_i^n\}, i \in [1, N]$, x^p is a positive sample image of the same identity, and x^n is a negative sample image of a different identity. suppose the distance between the samples can be expressed as follow,

$$d\left(x_i, x_j\right) = \left\| f\left(x_i\right) - f\left(x_j\right) \right\|_2 \tag{3}$$

where $f(\cdot)$ convert a image to the embedding space. Then the triplet loss can be obtained as follows,

$$L_{tri} = \sum_{i=1}^{N} \left[d\left(x_i^a, x_i^p\right) - d\left(x_i^a, x_i^n\right) + \alpha \right]_+ \tag{4}$$

where $[z]_+ = max(z, 0)$. α is a hyper-parameter that forces the positive and negative sample pairs in the embedding space. Actually, The triplet loss requires that for any triplet tuple $\{x_i^a, x_i^p, x_i^n\}$, the positive and negative sample pairs meet the follow inequality,

$$d\left(x_i^a, x_i^p\right) + \alpha < d\left(x_i^a, x_i^n\right) \tag{5}$$

As generated by clustering, some labels may be inaccurate. Based on the hypothesis that farther away features are less likely belong to the same class, we adopt the hard triplet loss [20] rather than triplet loss. According to [20], the hard triplet loss can improve the training speed and accuracy in many retrieval tasks by improving the triplet sampling method. Each batch consists of P categories, and each category consists of K images. For each anchor in the batch, select the hardest positive and negative samples to form the hardest triplet. The hardest positive sample represents the positive sample with the largest distance from the anchor, and the hardest negative sample represents the negative sample with the smallest distance from the anchor. Then the hard triplet loss can be expressed as follow,

$$L_{htri} = \sum_{i=1}^{P} \sum_{a=1}^{K} \left[\alpha + \overbrace{\max_{p=1\ldots K} d\left(x_i^a, x_i^p\right)}^{hardest\ positive} - \overbrace{\min_{j=1\ldots P, n=1\ldots K, j\neq i} d\left(x_i^a, x_j^n\right)}^{hardest\ negative} \right]_+ . \tag{6}$$

3.3.2 Reconstruction loss

As mentioned before, we use reconstruction loss to learn the spatial dependencies within the images. Given a image X, $\hat{X}$ is the corresponding reconstruction images through the Auto-Encoder (AE) structure, $\hat{X} \approx \mathcal{D}(\mathcal{E}(X))$. We hope the $\hat{X}$ is similar to X as much as possible. So our reconstruction loss can be expressed as follow,

$$L_{re} = \sum_{i,j} \|\hat{X_{ij}} - X_{ij}\|_2^2. \tag{7}$$

where X_{ij} represents the ij-th pixel in image X.

3.3.3 Final loss

The hard triplet loss and the reconstruction loss represent different demand for the model. The final loss integrate the advantages of these two loss, which can be described as follow,

$$L_{all} = L_{re} + \lambda_1 L_{htri} \tag{8}$$

where $\lambda_1 \in (0, 1)$ is a trade-off parameters, and it will increase with the progresses of iteration.

3.4 Mutual guidance

According to our method, if we want to get more accurate clustering result, we need more excellent feature extractor. If we want to train more excellent feature extractor, we need a more accurate clustering result. There two are mutually prerequisites. In order to let generating labels and training model benefit from each other, we design the mutual guidance for them. Simultaneously, in order to prevent the model from guiding itself into a closed loop, we set up a dual network method.

Firstly, we train an AE (denoted as IAE) model through the reconstruction task, regarded as $baseline$. Then we redefine two AE networks $AE_1 = \{E_1, D_1\}$ and $AE_2 = \{E_2, D_2\}$, and both of them copy the parameters of IAE. In the generating labels process (**GLP**), we randomly divide the training set into A and B, and A and B are the same number of training points. Then we use E_1 to extract the features of A, denoted as F_A, while F_B represents the features of B extracted by E_2. Both F_A and F_B are used to cluster to get the L_A and L_B, which represent the images with category labels. In the training models process (**TMP**), we use the L_A to train AE_2, and use L_B to train AE_1. The hard triplet loss L_{htri} is employed in the features extracted by encoder E, and the reconstruction loss L_{re} is employed in the output of decoder D.

The GLP and TMP are performed alternately. One GLP and one TMP are considered as one iteration. In one iteration, we train models N epochs in corresponding training points. And In the processes of iteration, the λ_1 will increase due to the result of clustering will be more accurate.

4 Experiments

In this section, we conduct a series of experiments to compare the performance of our model with fully-supervised method at classification and object detection tasks. Firstly, we introduce experiment setup, followed by training process and performance evaluation.

4.1 Experiment Setup

4.1.1 Datasets

Because the limitation of computing power and time, we train the auto-encoder part with miniImageNet, which is one of the mainly used datasets in the field of few shot learning. It is composed of 60,000 images of 100 categories.

To verify the quality of learned feature representations, we evaluate the transfer performance of feature representations on both image classification task using linear SVMs classifier and object detection task using Faster R-CNN [21] on Pascal VOC dataset. There are 20 classes categorized into person, vehicles, animals, and so on in Pascal VOC dataset. The dataset is divided into two subsets: training and validation with 1464 and 1449 images respectively.

4.1.2 Evaluation Metrics

We evaluate the proposed MGTA on image classification and object detection tasks. Self-supervised learning is a way to learn feature representations rather than an 'initialization method' [22] and thus we perform limited fine-tuning of the features. The common transfer process for image classification and object detection is as follows: Fist, we perform self-supervised pre-training using the proposed self-supervised learning method MGTA on miniImageNet dataset. Second, we extract features from res4 (the last layer of the 4th residual stage) of ResNet-50. Thirdly, we then evaluate the quality of these learned feature representations by transfer learning on dataset Pascal VOC and specific task.

Image classification. we extract image features from res4 of ResNet50 and train linear classifiers on these fixed feature representations. More specifically, we train linear SVMs on the frozen feature representations on trainval split of VOC2007 dataset and evaluate on test split of VOC2007 dataset.

Object detection. we use the Detectron framework [23] to train the Faster R-CNN [21] object detection model on VOC2007 and VOC2007+2012 datasets. More specifically, we freeze the full conv body of Faster R-CNN [21] and only train the RPN and ROI heads. We train object detection model using 2 GPUs with initial learning rate of 2e-3. And we train Faster R-CNN [21] for 50k/38k iterations on VOC2007 and for 100k/65k iterations on VOC2007+2012. All other hyper-parameters

are defaults set in Detectron [23], in which training with scale value 1000 and inference with scale value 600.

4.2 Training

In this section, we give a detailed description to how we train the auto-encoder part. Firstly, all of the 60,000 images are used to train the auto-encoder part for 100 epochs. At this stage, Mean Square Error(MSE) is used as loss function to enhance the model's reconstruction ability. Then the trained parameters are used to initialize two models donated by N_1 and N_2 for convenience. We split the miniImageNet dataset into two parts D_1 and D_2 equally which are used to make the distance between the features of the "same" category small and "different" category large. We use the N_1 to get the features of D_2 and clustering algorithm, such as K-means, is used to group these features into predefined number of categories. The images whose features belong to the same category are thought to be in the "same" category. It is clear the labels is with noise, but this problem can be solved by training iteratively as descriped below. To test the influence of the number of clustering, we conduct experiment with $k = 100$ and $k = 300$. After getting the labels of D_2, we train N_2 using D_2 for 10 epochs with MSE loss function and Triplet loss function jointly. Similarly, we process D_1 with N_2 and train N_1 by following the same steps. Then the whole procedure is repeated for 10 times and the weight of Triplet loss is increasing from 0.1 to 1 linearly. Finally, the features of all 60,000 images are got by N_1 and are processed by K-means algorithm. N_2 which is trained with all 60,000 images for 10 epochs is our final result.

4.3 Performance Evaluation

Image classification. The IAE baseline achieves 23.8% mAP on the VOC2007 classification benchmark, which is 15.8% higher than the randomly initialized ResNet50. It suggests that the auto-encoder architecture and the reconstruction loss is effective for self-supervised learning, which could learn relative good feature representations. The proposed MGTA achieves 29.7% and 31.3% mAP respectively when the *k* value for clustering differs from 100 to 300. Apparently, MGTA exceeds the baseline by a large margin (+5.9% when *k*=100, +7.5% when *k*=300). It uncovers that the iterative model training and clustering paradigm could help to obtain a better feature space and improve the classification performance. When compared to the ResNet50 trained with full annotations, the self-supervised method MGTA faces a significant accuracy gap (-24.0%), which implies that much future work could be done for further enhancement.

Table 2: ResNet-50 Linear SVMs mAP on the VOC07 classification benchmark. ResNet50 Random is the randomly initialized ResNet50. IAE is the baseline which is only trained on the reconstruction task. MGTA (*k*=100) and MGTA (*k*=300) is our method with different *k* values for clustering.

Method	**PASCAL VOC2007 Classification**
ResNet50 Random	8.0
ResNet50 IAE (Baseline)	23.8
ResNet50 MGTA (*k*=100)	29.7
ResNet50 MGTA (*k*=300)	**31.3**
ResNet50 Supervised	52.8

Object detection. We note that compared to image classification task, the object detection task lack of ResNet50 Random method because of disconvergence in training. Due to freezing whole conv body and small size of miniImageNet, the detection mAPs of all methods are relatively lower to mothods pretraining on ImageNet-1k. IAE achieves a relatively low detection mAP (11.27%@VOC2007 and 17.01%@VOC2007+2012), indicating that the reconstruction task can benefit the learning of feature representation compared to random initialization. The detection mAPs of MGTA(*k*=100) and MGTA(*k*=300) are almost the same and both outperforms the baseline IAE by a large margin (+4.61%@VOC2007 and +4.18%@VOC2007+2012 when *k*=100, +3.80%@VOC2007 and +2.83%@VOC2007+2012 when *k*=300). Based on reconstruction task, the proposed MGTA adds Mutual guidance to learn a better feature representation more effectively. And it shows that the feature space obtained by the iterative model training and clustering paradigm is better and also improves the object detection performance. Similarly, there exists a significant accuracy gap (-17.79%@VOC2007 and -16.65%@VOC2007+2012 when *k*=100) between the proposed

Figure 2: Some examples of detection results on Pascal VOC dataset. The top row is the better detection results, while the bottom row is the worse detection results. The proposed MGTA can encode spatial information and semantic information to a certain extent so that some objects (the top row) are detected correctly. However, There is still some gap between self-supervised and fully-supervised and some objects (the bottom row) are missed, duplicated or misclassified.

MGTA and the fully-supervised method, which means there exists much room for future improvement. However, MGTA is still worth mentioning since it could exploits large quantities of unlabeled images directly to improve the performance log-linearly with the data size [22].

Table 3: **Detection mAP for frozen conv body** on VOC2007 and VOC2007+2012 using Faster R-CNN [21] with ResNet-50-C4. We freeze the conv body for all models. '-' means disconvergence in training.

Method	**PASCAL VOC2007**	**PASCAL VOC2007+2012**
ResNet50 Random	-	-
ResNet50 IAE (Baseline)	11.27	17.01
ResNet50 MGTA (k=100)	**15.88**	**21.19**
ResNet50 MGTA (k=300)	15.07	19.84
ResNet50 Supervised	33.67	37.84

5 Conclusion

Self-supervised learning aims to learn a robust feature representation, with the expectation that this representation can carry good structural or semantic meanings and can be beneficial to a variety of practical downstream tasks. This can be achieved by framing a supervised learning task in a special form to predict only a subset of information using the rest. In this work, we proposed a novel self-supervised learning pipeline, Mutual Guidance to Train Auto-Encoder (MGTA). MGTA can learn a robust feature representation encoding both structural meanings and semantic effectively. MGTA trains two separate auto-encoder networks to reconstruct the images alternately to encode the structural meanings, at the same time, the feature representations of images are adaptively clustered to generate the pseudo labels and then a hard triplet loss is used to learn the semantic. To investigate the quality of feature representation, both image classification and object detection tasks are evaluated on Pascal VOC dataset and we observe that the proposed MGTA steadily outperforms the baseline by a large margin on different tasks.

Future Work

Although the proposed MGTA can learn a robust feature representation relatively, there still exists much room for future improvement and many experiments are not carried out due to the limited time and computing power.

Firstly, producing a dataset with clean labels is expensive but unlabeled data is being generated all the time, to make use of this much larger amount of unlabeled data, most of self-supervised learning methods conduct the related experiments on a much larger dataset, such as ImageNet-1k. While we carry out our experiments on a small image dataset miniImageNet which only contains 60,000 images, it is far from being enough for self-supervised learning. So we will do some experiments on a lager dataset to verify the effectiveness of MGTA.

Secondly, image reconstruction is not the goal of self-supervised learning but a pretext task to give supervision signal, however, the network structure of auto-encoder is crucial to the learning of feature representation. The network structure investigation of auto-encoder will be carried out in the future, so that a better auto-encoder structure can exert more effective supervision signal.

Thirdly, we have studied the empact of different cluster numbers to the quality of learned feature representation.It is reasonable that different cluster numbers heavily influence the quality of feature representation because the distribution of feature representations is getting more and more separable as the training process going. We consider using an adaptive cluster method to generate pseudo labels instead of K-means with fixed cluster numbers.

In addition, as the proposed MGTA takes a long time to train model and there is no precedent method to refer to, the adjustment of hyper-parameters is very expensive. This means the hyper-parameters of MGTA are probably not optimal and there exist much room to search a better hyper-parameters.

In a word, our proposed MGTA certainly has many shortcomings and problems, but is provides an efficient process to learn the feature representations of images.

Acknowledgements

Thanks the course Advanced Machine Learning for providing us with an opportunity to learn advanced machine learning knowledge, and thanks all of the teachers and teaching assistants for your efforts to this course.

References

[1] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[2] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[3] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[4] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.

[5] Ka Yu Hui. Direct modeling of complex invariances for visual object features. In *International conference on machine learning*, pages 352–360, 2013.

[6] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

[7] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[8] Iou-Jen Liu, Raymond A Yeh, and Alexander G Schwing. Pic: Permutation invariant critic for multi-agent deep reinforcement learning. *arXiv preprint arXiv:1911.00025*, 2019.

[9] T Nathan Mundhenk, Daniel Ho, and Barry Y Chen. Improvements to context based self-supervised learning. In *CVPR*, pages 9339–9348, 2018.

[10] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9359–9367, 2018.

[11] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5898–5906, 2017.

[12] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[13] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[14] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017.

[15] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[16] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[17] Mohammad Sabokrou, Mohammad Khalooei, and Ehsan Adeli. Self-supervised representation learning via neighborhood-relational encoding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8010–8019, 2019.

[18] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[22] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. *arXiv preprint arXiv:1905.01235*, 2019.

[23] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron, 2018.