Weakly Supervised Representation Learning with Coarse Labels

Yuanhong Xu[1] Qi Qian[2*] Hao Li[1] Rong Jin[2] Juhua Hu[3]
[1] Alibaba Group, Hangzhou, China
[2] Alibaba Group, Bellevue, WA, 98004, USA
[3] School of Engineering and Technology
University of Washington, Tacoma, WA, 98402, USA
{yuanhong.xuyh, qi.qian, lihao.lh, jinrong.jr}@alibaba-inc.com, juhuah@uw.edu

Abstract

With the development of computational power and techniques for data collection, deep learning demonstrates a superior performance over most existing algorithms on visual benchmark data sets. Many efforts have been devoted to studying the mechanism of deep learning. One important observation is that deep learning can learn the discriminative patterns from raw materials directly in a task-dependent manner. Therefore, the representations obtained by deep learning outperform hand-crafted features significantly. However, for some real-world applications, it is too expensive to collect the task-specific labels, such as visual search in online shopping. Compared to the limited availability of these task-specific labels, their coarse-class labels are much more affordable, but representations learned from them can be suboptimal for the target task. To mitigate this challenge, we propose an algorithm to learn the fine-grained patterns for the target task, when only its coarse-class labels are available. More importantly, we provide a theoretical guarantee for this. Extensive experiments on real-world data sets demonstrate that the proposed method can significantly improve the performance of learned representations on the target task, when only coarse-class information is available for training.

1. Introduction

Deep learning attracts more and more attentions due to its tremendous success in computer vision [11, 14, 19] and NLP applications [7, 21]. With modern neural networks, deep learning can even achieve a better performance than human beings on certain fundamental tasks [14, 26]. The improvement from deep learning makes many applications, e.g., autonomous driving [5], visual search [23], question-answering system [30], etc., become feasible.

*Corresponding author

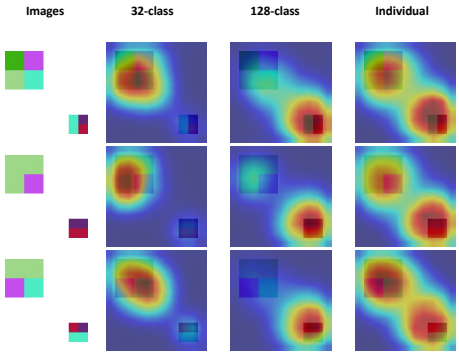

Figure 1. Illustration of different patterns learned from different tasks on the same synthetic data consisting of 512 images. According to different combinations of patches, three tasks are included: 32 coarse-class classification (i.e., 32-class with big patches), 128-class classification (i.e., 128-class with small patches) and instance-level classification (i.e., Individual with big and small patches). The detailed setting of the experiment can be found in supplementary.

Compared with many existing models, which are designed for hand-crafted features, deep learning works in an end-to-end learning manner. It can explore the most discriminative patterns (i.e., features) from raw materials directly for a specific task. Without an explicit phase of generating features, deep learning demonstrates a significant improvement over existing methods [14, 19]. Using the features generated by deep learning, conventional methods can also perform better than the counterpart with hand-crafted features [2, 8, 11, 12]. This observation implies that neural networks can learn the task-related patterns sufficiently.

In deep learning, representations are often learned with respect to a specific task. Therefore, different patterns can be extracted even on the same data set for different application scenarios as shown in the example of Fig. 1. This

phenomenon demonstrates that neural networks will only pay attention to those patterns that are helpful for the training task and ignore the unrelated patterns. Therefore, deep learning has to access a massive amount of labeled examples to achieve the ideal performance while the label information has to be closely related to the target task.

With the development of deep learning, a large training data size has been emphasized and many large-scale labeled data sets [6, 20] become available. However, the correlation between the learned representations from provided labels and the target task is less investigated. In some real-world applications, it is often too expensive to gather task-specific labels, while their coarse-class labels are much more accessible. Taking visual search [23] as an example, given a query image of "husky", a result of "husky" is often expected than a "dog". Apparently, the label information like "husky" is much more expensive than that like "dog". The problem becomes more challenging in the online shopping scenario, where many items (e.g., clothes) have very subtle differences. The gap between the available labels and the target task makes the learned representations suboptimal.

To improve the performance of learned representations for a target task, a straightforward way is to label a sufficient number of examples specifically for that task, which can align the supervised information and the target task well. However, this strategy is not affordable. Unlike coarse-class labels, some task-specific labels (e.g., species of dogs) can only be identified by very experienced experts, which is expensive and inefficient. For the visual search task in the online shopping scenario, even experts cannot label massive examples accurately.

Recently, unsupervised methods become popular for representation learning [3, 9, 13, 24, 29]. These methods first learn a deep model without any supervision on the source domain. After that, the learned model will be fine-tuned with the labeled data from the target domain. Although the pre-trained model is learned in an unsupervised manner, task-specific labels are required in the phase of fine-tuning, which are often very limited or have no access in some real-world applications. Considering that their coarse-class labels are much more affordable, in this work, we study the problem when data is from the target domain but only coarse-class labels are available.

Concretely, we aim to mitigate the issue by leveraging the information from coarse classes to learn appropriate representations for a target task. We verify that fine-grained patterns, which are essential for a target task, are often neglected when the deep model is trained only with coarse-class labels. Meanwhile, the popular pretext task in unsupervised representation learning, i.e., instance classification, may introduce too many noisy patterns that are irrelevant to the target task. Fortunately, we can theoretically prove that incorporating the task of coarse-class classification, representations learned from instance classification will be more appropriate for the target task. Based on this, we propose a new algorithm to learn appropriate representations for a target task when the task-specific labels are not available but their coarse-class labels are accessible. Besides, inspired by our analysis, a novel instance proxy loss is proposed to further improve the performance. Extensive experiments on benchmark data sets demonstrate that the proposed algorithm can significantly improve the performance on real-world applications when only coarse-class labels are available.

2. Related Work

Different from many existing methods, deep learning can directly learn patterns from raw materials, which avoids the information loss in the phase of feature extraction. By investigating the patterns learned by deep neural networks, researchers find that it can adaptively figure out discriminative parts in images for classification due to the end-to-end learning manner [8, 19], which interprets the effectiveness of convolutional neural networks (CNNs).

Besides supervised learning, unsupervised representation learning attracts much attention recently since it does not require any supervised information and can exploit the information from the large-scale unlabeled data sets [3, 9, 13, 29]. A popular pretext task is instance classification [9] that identifies each example as an individual class, while the computational cost can be a challenge on a large-scale data set. After its success, many algorithms are developed to improve the efficiency by contrastive learning [3, 13, 29].

Despite the desired performance on the target domain, the process still relies on fine-tuning with labels of the target task. It is because that instance classification aims to identify each individual example and may introduce too many irrelevant patterns for the target task. Therefore, a fine-tuning phase is necessary to filter noisy patterns. Besides, the gap between the source and target domain may degrade the performance of learned representations [17, 22]. In this work, we focus on the application scenario when the task-specific labels are hard to access, while their coarse-class labels (e.g., main categories for animals) are much cheaper and more accessible. We will leverage the weakly supervised information from coarse classes to improve the performance of learned representations on the target task, when target-specific labels are not available for fine-tuning.

It should be noted that generalizing learned models for different target tasks has also been researched in transfer learning and domain adaptation [31]. However, the problem addressed in this work is significantly different from them. Both of transfer learning and domain adaptation try to improve the performance on the target domain with the knowledge from a different source domain. In this work, we focus on learning with data from the target domain only.

3. Proposed Method

Given a set of n images $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, a model can be learned by solving the optimization problem

$$\min_\theta \sum_{i=1}^n \ell(\mathbf{x}_i, y_i; \theta)$$

where $\ell(\cdot)$ is the loss function and θ denotes the parameters of a neural network. Cross-entropy loss with the softmax operator is a popular loss in deep learning.

Many modern neural networks have multiple convolutional layers and a single fully-connected (FC) layer, e.g., ResNet [15], MobileNet [25], EfficientNet [28], etc. We will investigate this popular architecture in this work, while the analysis can be extended to more generic structures.

For a K-class classification problem, the cross-entropy loss can be written as

$$\ell(\mathbf{x}_i, y_i) = -\log \frac{\exp(f(\mathbf{x}_i)^\top \mathbf{w}_{y_i})}{\sum_j^K \exp(f(\mathbf{x}_i)^\top \mathbf{w}_j)}$$

where $f(\cdot)$ extracts features with convolutional layers from an image and $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\} \in \mathbb{R}^{d\times K}$ denotes the parameters of the last FC layer in a neural network. d is the input dimension of FC layer when ignoring the bias term.

Apparently, the behavior of function f heavily depends on the training labels in $\{y_i\}$. When the task implied by $\{y_i\}$ is consistent with the target one, the patterns discovered by f can perform well. However, when the training task is different from the target one (e.g., 32-class labels for the 128-class target task in Fig. 1), the learned patterns can be suboptimal. In this work, we aim to learn an appropriate function f that can extract sufficient and appropriate fine-grained patterns, even when only coarse-class labels are available.

3.1. Instance Classification

We start our analysis from the popular instance classification problem. The optimization problem for instance classification can be cast as

$$\min_\theta \sum_i \ell(\mathbf{x}_i, y_i^I; \theta) \tag{1}$$

where $y_i^I \in \{1, \ldots, n\}$ and $y_i^I = i$. The problem in Eqn. 1 considers that each example is from a different class, which leads to an n-class classification problem. It can be more challenging than the classification problem with target labels and various patterns will be extracted to identify each individual example. However, the desired patterns for the target task can be overwhelmed by too many patterns obtained from instance classification. Therefore, the obtained representations can be far away from optimum, which is demonstrated in the following theoretical analysis.

Let $W^I \in \mathbb{R}^{d\times n}$ denote the parameters of the FC layer for instance classification. Both of f^I and W^I will be optimized as the parameters of the neural network. We define the prediction probability as

$$\Pr\{y_i^I | f^I(\mathbf{x}_i), W^I\} = \frac{\exp(f^I(\mathbf{x}_i)^\top \mathbf{w}_{y_i^I}^I)}{\sum_j^n \exp(f^I(\mathbf{x}_i)^\top \mathbf{w}_j^I)}$$

It should be noted that we can have $\mathbf{w}_{y_i^I}^I = f^I(\tilde{\mathbf{x}}_i)$ in the contrastive learning [13] where $\tilde{\mathbf{x}}_i$ is a different view of $\mathbf{x}_i$. Assuming the task-specific labels are $y_i^F \in \{1, \ldots, F\}$ and $F < n$, the performance of learned representations on the target task without fine-tuning can be evaluated by measuring the probability

$$\Pr\{y_i^F | f^I(\mathbf{x}_i), W^I\} = \frac{\exp(f^I(\mathbf{x}_i)^\top \bar{\mathbf{w}}_{y_i^F}^I)}{\sum_s^F \exp(f^I(\mathbf{x}_i)^\top \bar{\mathbf{w}}_s^I)}$$

where $\bar{\mathbf{w}}_s^I = \frac{1}{z}\sum_{y_j^F = s} \mathbf{w}_j^I$. We assume each target class contains z examples to simplify the analysis and $zF = n$. In this formulation, we adopt the mean vector of parameters from the same target class as the proxy for the target classification problem. The probability can measure the intra-class variance and inter-class distance in the learned representations. By investigating the performance, we can have the guarantee for the representations learned from instance classification as in the following Lemma. All detailed proofs of this work can be found in the supplementary.

Lemma 1. *If solving the problem in Eqn. 1 such that* $\forall i, \Pr\{y_i^I | f^I(\mathbf{x}_i), W^I\} \geq \alpha$, *we have*

$$\forall i, \quad \Pr\{y_i^F | f^I(\mathbf{x}_i), W^I\} \geq z\alpha \exp(f^I(\mathbf{x}_i)^\top (\bar{\mathbf{w}}_{y_i^F}^I - \mathbf{w}_{y_i^I}^I))$$

Remark Lemma 1 shows that the performance of representations on the target task depends on both the accuracy of instance classification and the factor $f^I(\mathbf{x}_i)^\top \bar{\mathbf{w}}_{y_i^F}^I - f^I(\mathbf{x}_i)^\top \mathbf{w}_{y_i^I}^I$. When we have $\mathbf{w}_{y_i^I}^I = f^I(\mathbf{x}_i)$ as in contrastive learning, the factor becomes $\frac{1}{z}\sum_{y_j^F = y_i^F} f^I(\mathbf{x}_i)^\top f^I(\mathbf{x}_j)$. Explicitly, the latter factor is corresponding to the intra-class variance.

Since instance classification is to identify every individual example, it can handle the inter-class difference well but the similarity between examples from the same target class can be arbitrary due to redundant patterns, which may result in a suboptimal performance. Therefore, we consider to leverage the coarse-class information to aggregate examples appropriately and filter irrelevant patterns to reduce intra-class variance.

3.2. Intra-Class Optimization

In many real-world applications, coarse-class labels (e.g., "dog", "cat", and "bird") are easy to access. The

learning problem with coarse-class labels can be defined as

$$\min_{\theta} \sum_{i} \ell(\mathbf{x}_i, y_i^C; \theta) \tag{2}$$

where $y_i^C \in \{1, \ldots, C\}$ indicates the coarse-class label of $\mathbf{x}_i$. In this work, we assume that examples from the same target class will share the same coarse labels. Explicitly, representations learned by solving this task can be inapplicable on a target task involving classes like "bulldog", "husky", and "poodle" under the coarse class "dog". It is because that the learned features have small intra-class variance but cannot handle the inter-class difference on the target classes. Consequently, they can separate the examples on the coarse classes well, while they cannot provide a meaningful separation for the target classes.

Based on these complementary observations from Eqns. 1 and 2, we consider to incorporate the problem in Eqn. 2 to guide the learning of fine-grained patterns in Eqn. 1. Intuitively, with the coarse-class label information, the model can explore the target task related fine-grained patterns more effectively. Thereafter, the classification problem can be written as

$$\min_{\theta} \sum_{i} \ell(\mathbf{x}_i, y_i^C) + \lambda \sum_{i} \ell(\mathbf{x}_i, y_i^I) \tag{3}$$

where λ is a trade-off between the performance of the coarse-class classification and instance classification, which is corresponding to reducing intra-class variance and increasing inter-class difference, respectively. The hybrid loss functions share the same backbone for feature extraction that is denoted as $f^H(\mathbf{x}_i)$. The classification head is different and we let the corresponding FC layer as W^C and W^I, respectively.

By optimizing the problem in Eqn. 3, we prove that the performance of the learned representations can be guaranteed on the target classes as follows.

Theorem 1. *If learned representations have the bounded norm as* $\forall i, j, \|f^H(\mathbf{x}_i)\|_2, \|\mathbf{w}_j^I\|_2, \|\mathbf{w}_j^C\|_2 \leq c$ *and solving the problem in Eqn. 3 such that*

$$\forall i, \Pr\{y_i^I | f^H(\mathbf{x}_i), W^I\} \geq \alpha; \Pr\{y_i^C | f^H(\mathbf{x}_i), W^C\} \geq \beta$$

where α, β *are constants that are balanced by* λ*, we have*

$$\forall i, \quad \Pr\{y_i^F | f^H(\mathbf{x}_i), W^I\} \geq \alpha z h(c, \alpha, \beta)$$

where $h(c, \alpha, \beta) \leq 1$ *is a constant that depends on* c, α, β.

Remark Concretely, with the help of Eqn. 2, we can bound the difference between examples from the same target class in $h(c, \alpha, \beta)$, while Eqn. 1 helps obtain sufficient fine-grained patterns to identify different classes for the target problem.

It should be noted that the sub-problem of instance classification in Eqn. 3 is an n-class classification problem. When n is large, it has to compute the scores and the corresponding gradient from $W^I \in \mathbb{R}^{d \times n}$ for each example, which can slow down the optimization significantly. This challenge has been extensively studied in the literature of unsupervised representation learning and mitigated by contrastive learning [3, 13]. Differently, we can decompose the instance classification problem according to the coarse classes in our work, which is discussed in the following subsection.

3.3. Large-Scale Challenge

According to the analysis in Theorem 1, we can decompose the original problem as

$$\min_{\theta} \sum_{i} \ell(\mathbf{x}_i, y_i^C) + \lambda \sum_{k=1}^{C} \sum_{i: y_i^C = k} \ell_k(\mathbf{x}_i, y_i^I) \tag{4}$$

where $\ell_k(\mathbf{x}_i, y_i^I)$ is the cross entropy loss defined for instance classification within the k-th coarse class

$$\begin{aligned} \ell_k(\mathbf{x}_i, y_i^I) &= -\log(\Pr\{y_i^I | f^H(\mathbf{x}_i), y_i^C, W^I\}) \\ &= -\log(\frac{\exp(f^H(\mathbf{x}_i)^\top \mathbf{w}_{y_i^I}^I)}{\sum_{j: y_j^C = k} \exp(f^H(\mathbf{x}_i)^\top \mathbf{w}_j^I)}) \end{aligned}$$

Compared with the standard instance classification, the new loss is to distinguish between the example $\mathbf{x}_i$ and other examples with the same coarse-class label (i.e., $y_j^C = k$) in lieu of total n examples. Therefore, the computational cost of the FC layer for each example can be reduced from $\mathcal{O}(dn)$ to $\mathcal{O}(dn_k)$, where n_k denotes the number of examples in the k-th coarse class.

We prove that the performance using the above speedup strategy can still be guaranteed on the target problem as stated in the following theorem.

Theorem 2. *With the same assumptions as in Theorem 1, if solving the problem in Eqn. 4 such that*

$$\forall i, \Pr\{y_i^I | f^H(\mathbf{x}_i), y_i^C, W^I\} \geq \alpha; \Pr\{y_i^C | f^H(\mathbf{x}_i), W^C\} \geq \beta$$

we have

$$\forall i, \quad \Pr\{y_i^F | f^H(\mathbf{x}_i), W^I\} \geq \alpha' z h(c, \alpha', \beta)$$

where $\alpha' = \frac{1}{1/\alpha + (1-\beta)c''/\beta}$ *and* c'' *is a constant.* $h(c, \alpha', \beta)$ *is a constant that depends on* c, α', β.

Remark Compared with the guarantee in Theorem 1, the cost of relaxation is given in α'. It contains a factor of $(1-\beta)/\beta$, which measures the performance on the coarse-class classification problem. When an example can be separated well from other coarse classes as $\beta \to 1$, the patterns obtained by solving Eqn. 4 can almost recover the performance from solving the more expensive problem in Eqn. 3.

3.4. Instance Proxy Loss

Till now, we theoretically analyze the behaviors of instance classification and coarse-class classification. Inspired by our analysis, we propose a novel loss to enhance the informative patterns for the target task.

A standard proxy-based triplet constraint [23] for representation learning can be written as

$$\forall \mathbf{x}_i, \mathbf{c}_{j:j\neq y_i}, \quad \|\mathbf{x}_i - \mathbf{c}_j\|_2^2 - \|\mathbf{x}_i - \mathbf{c}_{y_i}\|_2^2 \geq \delta$$

where $\mathbf{c}_j$ denotes the proxy for the j-th class and δ is a margin. We omit the feature extraction function $f^H(\cdot)$ for brevity. In Theorem 1, we demonstrate that the mean vector of individual classes from the same target class can be an appropriate proxy for the target task. However, the labels of target task are not available when training representations. Therefore, assuming that there are P target classes, we will learn the relation with a membership variable $\mu \in \{0,1\}^{n\times P}$ ($\forall i, \sum_p \mu_{i,p} = 1$) simultaneously. Specifically, we let W^P denote the parameters for the P-class classification problem and

$$\mathbf{w}_p^P = \frac{\sum_i \mu_{i,p}\mathbf{w}_i^I}{\sum_i \mu_{i,p}} \tag{5}$$

With the proxy from averaging instance parameters, we have the triplet constraints as

$$\forall \mathbf{x}_i, \sum_p \frac{(1-\mu_{i,p})}{P-1}\|\mathbf{x}_i - \mathbf{w}_p^P\|_2^2 - \sum_j \mu_{i,j}\|\mathbf{x}_i - \mathbf{w}_j^P\|_2^2 \geq \delta$$

To maximize the margin δ, the optimization problem can be written as

$$\min_{\mathbf{x},\mu} \sum_i \Big(\sum_j \mu_{i,j}\|\mathbf{x}_i - \mathbf{w}_j^P\|_2^2 - \sum_p \frac{(1-\mu_{i,p})}{P-1}\|\mathbf{x}_i - \mathbf{w}_p^P\|_2^2\Big) \tag{6}$$

The problem can be solved in an alternating manner. At each epoch, when fixing μ, the representation can be optimized over P classes as

$$\min_{\mathbf{x}} \sum_i \|\mathbf{x}_i - \mathbf{w}_{y_i^P}^P\|_2^2 - \sum_{p:\mu_{i,p}=0} \frac{\|\mathbf{x}_i - \mathbf{w}_p^P\|_2^2}{P-1}$$

where $\mu_{i,y_i^P} = 1$. Following the suggestion in [23], we propose an instance proxy loss to optimize the sub-problem effectively as

$$\ell_p(\mathbf{x}_i, y_i^P) = -\log\Big(\frac{\exp(f^H(\mathbf{x}_i)^\top \mathbf{w}_{y_i^P}^P)}{\sum_p \exp(f^H(\mathbf{x}_i)^\top \mathbf{w}_p^P)}\Big) \tag{7}$$

When fixing $\mathbf{x}$, the sub-problem becomes

$$\min_{\mu} \sum_i P \sum_j \mu_{i,j}\|\mathbf{x}_i - \mathbf{w}_j^P\|_2^2 - \sum_p \|\mathbf{x}_i - \mathbf{w}_p^P\|_2^2$$

Algorithm 1 Representation Learning with Coarse Labels

Input: training set $\{\mathbf{x}_i, y_i^C\}_{i=1}^n$, total epochs T, M, P, λ_I, λ_P
for epoch: $t = 1$ **to** M **do**
  Optimize the problem in Eqn. 4
end for
Obtain P clusters with W^I
Initialize W^P as in Eqn. 5
for epoch: $t = M+1$ **to** T **do**
  Optimize the problem in Eqn. 9
  Update W^P with fixed W^I by solving Eqn. 8
end for

Note that W^P also contains μ that makes the optimization challenge. When P is large, the latter term can be considered as a constant (e.g., $P = n$ for the extreme case) and the problem can be simplified as

$$\min_{\mu} \sum_i \sum_j \mu_{i,j}\|\mathbf{x}_i - \mathbf{w}_j^P\|_2^2$$

Since W^P is spanned by W^I, we can optimize the upper-bound instead

$$\min_{\mu} \sum_i \sum_j \mu_{i,j}\|\mathbf{w}_i^I - \mathbf{w}_j^P\|_2^2 + \|\mathbf{x}_i - \mathbf{w}_i^I\|_2^2$$

Without the constant term, the problem can be rewritten as

$$\min_{\mu, W^P} \sum_i \sum_j \mu_{i,j}\|\mathbf{w}_i^I - \mathbf{w}_j^P\|_2^2$$
$$s.t. \qquad \mathbf{w}_p^P = \frac{\sum_i \mu_{i,p}\mathbf{w}_i^I}{\sum_i \mu_{i,p}} \tag{8}$$

Therefore, it becomes a standard k-means clustering problem and can be solved efficiently. To make the approximation tight, i.e., $\|\mathbf{x}_i - \mathbf{w}_i^I\|_2^2$ is small, we have to optimize the problem in Eqn. 6 after W^I is sufficiently trained.

With the proposed instance proxy loss, the objective for representation learning becomes

$$\min_{\theta} \sum_i \ell(\mathbf{x}_i, y_i^C) + \lambda_I \sum_{k=1}^{C} \sum_{i:y_i^C=k} \ell_k(\mathbf{x}_i, y_i^I)$$
$$+\lambda_P \sum_{p=1}^{P} \sum_i \ell_p(\mathbf{x}_i, y_i^P) \tag{9}$$

Alg. 1 summarizes the proposed algorithm. Note that the clustering can be implemented within each coarse class as suggested in Section 3.3.

4. Experiments

To evaluate the proposed method, we adopt ResNet-18 [15] as the neural network for comparison since it is the

most popular deep architecture and has been widely applied for real tasks. We include five methods in the main comparison as follows.

Ins: optimize representations with instance classification only as in Eqn. 1.

Cos: optimize representations with coarse-class classification only as in Eqn. 2.

CoIns: learn representations with coarse-class classification and instance classification simultaneously as in Eqn. 3.

CoIns$_{\text{imp}}$: improve the efficiency by optimizing the instance classification within each coarse class as in Eqn. 4.

Opt: optimize representations with target labels that are not available in our problem setting. Therefore, this method provides the performance upper-bound as a reference.

ResNet-18 is trained with stochastic gradient descent (SGD). All methods in the comparison have the same backbone network and training pipeline but with different objectives and classification heads. Augmentation is important for training CNNs and we adopt both random horizontal mirroring and random crop as suggested in [15]. Other configurations on each data set follow the common practice and are elaborated in the corresponding subsections.

Three benchmark image data sets, i.e., CIFAR-100 [18], SOP [27], and ImageNet [6], are included for comparison. We note that all of these data sets contain both coarse-class labels and target-class labels for a comprehensive evaluation, where target-class labels are only used by "Opt" to provide the upper-bound of the performance.

We evaluate the performance of different representations with multiple metrics. First, we measure the accuracy on coarse classes as a side product. With more fine-grained patterns, the generalization on coarse classes can be further improved. More importantly, we evaluate the performance on the target classes by conducting the retrieval task (i.e., visual search) that motivates this work. We adopt Recall@k metric as in [23, 27] for comparison. The similarity for retrieval is computed by the cosine similarity using the outputs before the FC layer, i.e., $f(\mathbf{x})$. [23] shows that deep features learned by classification can capture the similarity between examples well.

4.1. CIFAR-100

In this subsection, we evaluate the methods on CIFAR-100 [18] that contains 20 coarse classes. Each coarse class contains 5 target classes that contribute 100 target classes. We adopt the standard splitting, where each target class has 500 color images for training and 100 for test.

SGD with a mini-batch size of 256 is applied to learn the model. Following the common practice, we set momentum to 0.9 and weight decay as $5e^{-4}$. Each model is trained with 200 epochs. The initial learning rate is 0.1 and is decayed by a factor of 5 at $\{60, 120, 160\}$ epochs. The 32×32 images is randomly cropped from the zero-padded 40×40 images for the crop augmentation. The only parameter in "CoIns" is λ that balances different loss functions and we search it in $\{1, 5\} \times \{10^{-i}\}_{i=0}^{4}$ for all experiments.

	Top1	Top5	R@1	R@2	R@4	R@8
Ins	-	-	22.4	32.9	46.8	62.6
Cos	85.6	97.5	81.1	87.0	90.7	93.2
CoIns	86.3	98.2	82.4	88.0	91.4	94.1
CoIns$_{\text{imp}}$	86.1	97.9	82.3	87.5	91.4	94.2

Table 1. Comparison of accuracy and recall (%) for 20 coarse classes on CIFAR-100. ("-" means NA)

As a side product, Table 1 summarizes both the classification and retrieval performance on the 20 coarse classes (i.e., not the target task). First, it is surprising to observe that fine-grained patterns learned by "CoIns" can improve the performance on the coarse-class classification problem. It illustrates that the task-dependent patterns learned by CNNs focus on the training task and can be suboptimal for unseen examples of the same problem. Exploring more fine-grained patterns in training as suggested by "CoIns" can generalize the learned patterns better on unseen data. Second, "CoIns$_{\text{imp}}$" has the similar performance as "CoIns". It is consistent with the analysis in Theorem 2. The examples with coarse-class labels can be separated well on this data set with an accuracy of more than 85%, which implies a large β in Theorem 2. Therefore, the performance of "CoIns$_{\text{imp}}$" can approach that of "CoIns" with significantly less computational cost. Note that there are 20 coarse classes with a uniform distribution in this data set, and thus the cost of computing the fully-connected layer for instance classification in "CoIns$_{\text{imp}}$" is only 5% of that in "CoIns".

A similar observation can be obtained for the retrieval task on these 20 coarse classes. We observe that "CoIns" and "CoIns$_{\text{imp}}$" can outperform the baseline "Cos" with a significant margin on R@1. "Ins" is included in this comparison while it provides the worst performance. It is because that the task of instance classification cannot leverage the supervised information from the coarse classes.

	R@1	R@2	R@4	R@8
Ins	13.6	19.2	27.1	37.3
Cos	37.1	51.6	67.0	79.9
CoIns	57.0	68.0	77.5	85.5
CoIns$_{\text{imp}}$	56.6	68.0	77.5	85.1
CoIns*	60.8	71.2	79.2	85.5
CoIns**	60.5	71.1	79.8	86.5
CoInsP**	62.0	71.7	80.2	86.6
Opt	71.8	78.8	84.1	88.3

Table 2. Comparison of recall (%) for 100 classes on CIFAR-100. CoIns* adopts cosine softmax while CoIns** has both cosine softmax and MLP head as in [1].

More importantly, the comparison on the target retrieval task of 100 classes is demonstrated in Table 2. Evidently, both "Cos" and "Ins" cannot handle the retrieval task well. As illustrated in our analysis, "Cos" lacks the fine-grained patterns, i.e., small inter-class difference and "Ins" lacks the guidance to filter massive noisy patterns, i.e., large intra-class variance. By complementing each other in "CoIns", the performance can be dramatically improved. The R@1 of "CoIns" is better than "Cos" by about 20% and surpasses "Ins" by more than 40%. It confirms the observation in Theorem 1 that the proposed method can explore the fine-grained patterns sufficiently and effectively for the target task when only coarse-class labels are available. Without doubt, "Opt" provides the best performance when target-class labels are available for training. Compared to "Opt", we can observe that R@4 of "CoIns" is better than R@1 of "Opt" and is comparable to R@2 of "Opt". It means that when only coarse-class labels are available, by optimizing the objective in Eqn. 3, the learned model can handle the target retrieval task well by retrieving two additional examples. Finally, the negligible difference between the performance of "CoIns" and "$\text{CoIns}_{\text{imp}}$" implies that "$\text{CoIns}_{\text{imp}}$" is efficiently applicable for real-world applications.

Many recent works including SimCLR [2], MoCo-v2 [4], and PIC [1] indicate that some additional components are essential for the success of unsupervised learning on ImageNet. Therefore, we introduce these components to "CoIns" to evaluate their effects in our problem. Specifically, three components including cosine softmax, MLP, and strong augmentation, are compared. We observe that strong augmentation always hurts the performance. It may be due to the fact that strong augmentation introduces too much noise for CIFAR, so we ignore its results in Table 2. "CoIns" with cosine softmax and with both cosine softmax and MLP are referred as **CoIns*** and **CoIns****, respectively.

From Table 2, it is evident that "CoIns*" can further improve the performance of "CoIns" with a significant margin of 3%, which is consistent with the observation in [1]. Since applying unit norm for both representations of examples and parameters in the FC layer, it can have a better guarantee as illustrated in Theorem 1. However, "CoIns**" with an additional MLP head cannot surpass "CoIns*" when retrieved examples are limited. The reason may be from the low resolution of images in CIFAR. Finally, we incorporate these two components into "CoInsP" that adds the proposed instance proxy (IP) loss for training as in Eqn. 7. In the experiment, we add the IP loss to "CoIns" after half of the training process, i.e., 100 epochs. To make the approximation tight as analyzed in Section 3.4, we have a large P as $P = 10,000$. By mimicking the target classes and enhancing instance classification, "CoInsP**" achieves the best performance that is closest to "Opt". Moreover, R@2 of "CoInsP**" already has the similar performance to R@1 of "Opt", which demonstrates the effectiveness of the proposed method.

4.2. Stanford Online Products

Then, we evaluate different algorithms in a challenging online shopping scenario. Stanford Online Products (SOP) [27] collects $120,053$ product images from eBay.com. There are a total of $22,634$ classes from 12 coarse classes. Therefore, each target class contains very limited number of examples. Since there is no public splitting on this data set for classification, we randomly sample $80,000$ images for training and the rest for test. We then filter all classes that contain only a single example in the test set. This leads to $13,160$ target classes for evaluation.

For training, we adopt the suggested configuration as in [15]. Specifically, the model is learned from scratch with 90 epochs. The initial learning rate is 0.1 and decayed by a factor of 10 at $\{30, 60\}$ epochs. The similar results as CIFAR for coarse-class classification and retrieval can be found in the supplementary.

	R@1	R@10	R@100
Ins	25.5	38.3	54.9
Cos	21.8	34.4	52.7
CoIns	35.8	51.8	69.3
$\text{CoIns}_{\text{imp}}$	35.3	50.5	67.4
CoIns*	38.1	54.2	70.6
CoIns**	42.7	58.2	73.7
CoInsP**	43.5	59.0	74.3
Opt	46.5	61.6	75.2

Table 3. Comparison of recall ($\%$) for $13,160$ classes on SOP. CoIns* adopts cosine softmax while CoIns** has both cosine softmax and MLP head as in [1].

The retrieval performance on the target classes is shown in Table 3. Considering the well-known difficulty of this task, we report the Recall@$\{1,10,100\}$ as suggested in [23, 27]. First, we can observe that "CoIns" outperforms "Cos" by 14% on R@1. It demonstrates that our method can be applied for online shopping scenario when there is limited supervision. Besides, even with the supervised information on target classes, R@1 of "Opt" is less than 50%, which shows that retrieval in online shopping is an important but challenging application. With more retrieved examples, recall of "CoIns" can outperform 50% as shown in R@10. Note that customers for online shopping tend to review only the top ranked items, which is known as position bias [16]. Therefore, improving R@10 is important for better customer experience.

With the additional components, "CoIns*" surpasses "CoIns", while "CoIns**" shows an even better performance that is closer to "Opt". It demonstrates that cosine softmax can benefit both of low-resolution images and high-

resolution ones, while MLP is especially effective for high-resolution images as in SOP. Finally, "CoInsP**" with an additional loss after 45 epochs demonstrates the best performance among variants of "CoIns". It shows that the informative patterns can be further captured using the proposed instance proxy loss.

We illustrate the retrieved images on SOP in Fig. 2. Evidently, there are many similar products from different target classes in online shopping, which makes the application very challenging. Given a query image, it is hard for "Cos" (i.e., baseline) to retrieve appropriate similar items. By learning fine-grained patterns sufficiently as in "CoIns", the examples from different target classes are eliminated from the top ranked items.

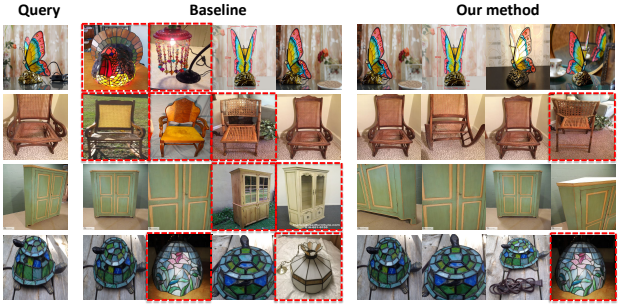

Figure 2. Examples of retrieved images from **Cos** (i.e., baseline) and **CoIns** (i.e., our method) on SOP. The examples from a different target class are denoted with red bounding-boxes.

4.3. ImageNet

Finally, we compare different methods on ImageNet [6]. ImageNet is a popular benchmark data set for visual categorization. It contains $1,000$ classes and each class has about $1,200$ images. These classes are organized according to WordNet [10] and there can be 11 coarse classes in ImageNet as analyzed in [22]. Each coarse class can have multiple target classes. For example, the coarse class "dog" has 118 different species of dogs and "bird" contains 59 different species of birds.

Instance classification has been extensively studied on ImageNet and many sophisticated algorithms have been developed. To make the comparison fair, we adopt one state-of-the-art method, MoCo-v2 [4], as the substitute of instance classification in Eqn. 3. We implement our method by adding coarse-class classification to the official code of MoCo. The training follows the configuration of MoCo-v2 with 200 epochs. We also adopt ResNet-50 rather than ResNet-18 in the comparison to align with the result of MoCo that is only implemented with ResNet-50.

Table 4 compares different methods on ImageNet. The performance of MoCo-v2 is directly borrowed from the official pre-trained model while that of "Opt" is from the pre-trained model provided by PyTorch[1]. First, we can observe that MoCo-v2 is worse than "Opt" by more than 20% on R@1. It demonstrates that without labels, instance classification cannot learn the patterns well related to the target classes. However, the performance "Cos" is even worse since we only introduce 11 coarse classes that cannot handle the inter-class difference for the target task. By incorporating these coarse classes as in our method, R@1 using learned representations can be increased from 42.8% to 51.4%. It confirms our analysis in Theorem 1 that coarse classes help to eliminate noisy patterns and can improve the performance on the target task.

	Top1	Top5	R@1	R@2	R@4	R@8
MoCo-v2	67.5	88.0	42.8	52.9	62.4	71.1
Cos	60.4	83.0	21.1	28.9	37.9	48.2
CoIns	70.4	89.9	51.4	61.5	70.7	78.4
Opt	76.2	92.9	66.4	75.3	82.1	87.3

Table 4. Comparison of accuracy and recall (%) for $1,000$ classes on ImageNet.

Besides, we also include the comparison of classification on $1,000$ target classes in Table 4. The performance is evaluated by a linear classifier with fixed representations and the classifier is learned by the standard pipeline provided in MoCo-v2. Note that target labels will be applied for training linear classification. First, the accuracy of MoCo achieves 67.5% after fine-tuning with target-class labels. It shows that unsupervised instance classification relies on target label information to filter noisy patterns in representations. With more related patterns in "CoIns", the Top1 accuracy can achieve 70.4%, which is about 3% better than MoCo-v2. This further demonstrates our proposed method. It also implies that even with full supervised information from the target task for fine-tuning, the representations learned from instance classification is worse than our proposal.

5. Conclusion

In this work, we propose an algorithm to explore fine-grained patterns sufficiently with an access of only coarse-class labels for training. The empirical study on benchmark data sets confirms the effectiveness of our proposed method and its theoretical guarantee. Besides, we propose a new instance proxy loss to further improve the performance according to our theoretical analysis.

Considering that the number of unlabeled data is significantly larger than that of labeled data, incorporating unlabeled data to improve the performance can be our future work. Moreover, there can be various weakly supervised information besides labels (e.g., triplet constraints, multiple views), exploring and incorporating more coarse information to catch up the performance upper-bound is also an interesting future direction.

[1] https://pytorch.org/vision/stable/models.html

References

[1] Yue Cao, Zhenda Xie, Bin Liu, Yutong Lin, Zheng Zhang, and Han Hu. Parametric instance classification for unsupervised visual feature learning. *CoRR*, abs/2006.14618, 2020. 6, 7

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020. 1, 7

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020. 2, 4

[4] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020. 7, 8

[5] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, pages 6526–6534, 2017. 1

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 2, 6, 8

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. 1

[8] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014. 1, 2

[9] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin A. Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(9):1734–1747, 2016. 2

[10] Christine Fellbaum. 1998, wordnet: An electronic lexical database, 1998. 8

[11] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 1

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *CoRR*, abs/1911.05722, 2019. 1

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735. IEEE, 2020. 2, 3, 4

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 1

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3, 5, 6, 7

[16] Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. *SIGIR Forum*, 51(1):4–11, 2017. 7

[17] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? In *CVPR*, pages 2661–2671. Computer Vision Foundation / IEEE, 2019. 2

[18] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 6

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1106–1114, 2012. 1, 2

[20] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 2

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, pages 3111–3119, 2013. 1

[22] Qi Qian, Juhua Hu, and Hao Li. Hierarchically robust representation learning. In *CVPR*, pages 7334–7342. IEEE, 2020. 2, 8

[23] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *ICCV*, 2019. 1, 2, 5, 6, 7

[24] Qi Qian, Yuanhong Xu, Juhua Hu, Hao Li, and Rong Jin. Unsupervised visual representation learning by online constrained k-means. *CoRR*, abs/2105.11527, 2021. 2

[25] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520. IEEE Computer Society, 2018. 3

[26] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 1

[27] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, pages 4004–4012, 2016. 6, 7

[28] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019. 3

[29] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742. IEEE Computer Society, 2018. 2

[30] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alexander J. Smola. Stacked attention networks for image question answering. In *CVPR*, pages 21–29, 2016. 1

[31] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proc. IEEE*, 109(1):43–76, 2021. 2