

# Embedding Label Structures for Fine-Grained Feature Representation

Xiaofan Zhang  
UNC Charlotte  
Charlotte, NC 28223  
xzhang35@uncc.edu

Feng Zhou  
NEC Lab America  
Cupertino, CA 95014  
feng@nec-labs.com

Yuanqing Lin  
NEC Lab America  
Cupertino, CA 95014  
ylin@nec-labs.com

Shaoting Zhang  
UNC Charlotte  
Charlotte, NC 28223  
szhang16@uncc.edu

## Abstract

Recent algorithms in convolutional neural networks (CNN) considerably advance the fine-grained image classification, which aims to differentiate subtle differences among subordinate classes. However, previous studies have rarely focused on learning a fine-grained and structured feature representation that is able to locate similar images at different levels of relevance, e.g., discovering cars from the same make or the same model, both of which require high precision. In this paper, we propose two main contributions to tackle this problem. 1) A multi-task learning framework is designed to effectively learn fine-grained feature representations by jointly optimizing both classification and similarity constraints. 2) To model the multi-level relevance, label structures such as hierarchy or shared attributes are seamlessly embedded into the framework by generalizing the triplet loss. Extensive and thorough experiments have been conducted on three fine-grained datasets, i.e., the Stanford car, the Car-333, and the food datasets, which contain either hierarchical labels or shared attributes. Our proposed method has achieved very competitive performance, i.e., among state-of-the-art classification accuracy when not using parts. More importantly, it significantly outperforms previous fine-grained feature representations for image retrieval at different levels of relevance.

## 1. Introduction

Recent advances in image understanding (e.g., classification, detection, segmentation, retrieval) have been driven by the success of convolutional neural networks (CNN) [17, 22, 36, 34, 32]. Particularly, models of fine-grained image categorization have made tremendous progress in recognizing subtle differences among subordinate classes, such as different models of cars [19, 21, 26, 45], breeds of animals [18, 28, 11, 4, 20, 24, 25], and types of food dishes [6, 47]. Most of previous methods focus on improving the classification accuracy, by learning critical parts that



Figure 1: Examples from a fine-grained car dataset [19], where the similarity can be defined at different levels, i.e., body type, model, and even viewpoint, indicated by the distance to the query in the center. Images within the circle have exactly the same fine-grained labels, i.e., make and model, and the closest two also share the same viewpoint.

can align the objects and discriminate between neighboring classes [46, 7, 3, 51, 50, 15], or using distance metric learning to alleviate the issue of large intra-class variation [41, 38, 39, 30]. However, such studies have rarely been dedicated to learn a structured feature representation that can discover similar images at different levels of relevance. Fig. 1 shows examples of similar cars from a fine-grained dataset [19]. Having the same fine-grained labels indicate exactly the same make, model and year, while cars are still similar even they have different labels, e.g., the same make but different year, or the same body style (e.g., SUV, Coupe) from different make. Such hierarchy of similarity should also be explored in fine-grained feature representation, since it is applicable to various use cases such as the recommendation of relevant products in e-commerce.

To obtain the fine-grained feature representation, a potential solution is to incorporate similarity constraints (e.g.,

contrastive information [10] or triplets [27, 8]). For example, Wang et al. [39] proposes a deep ranking model to directly learn the similarity metric by sampling triplets from images. However, these strategies still have several limitations in fine-grained datasets: 1) Although the features learned from triplet constraints are effective at discovering similar instances, its classification accuracy may be inferior to the fine-tuned deep models that emphasize on the classification loss, as demonstrated in our experiments. In addition, the convergence speed using such constraints is usually slow. 2) More importantly, previous methods for fine-grained features do not embed label structures, which is critical to locate images with relevance at different levels.

In this paper, we propose two contributions to solve these issues: 1) A multi-task deep learning framework is designed to effectively learn the fine-grained feature representation without sacrificing the classification accuracy. Specifically, we *jointly optimize* the classification loss (*i.e.*, softmax) and the similarity loss (*i.e.*, triplet) in CNN, which can generate both categorization results and discriminative feature representations. 2) Furthermore, based on this framework, we propose to seamlessly *embed label structures* such as hierarchy (*e.g.*, make, model and year of cars) or attributes (*e.g.*, ingredients of food). We evaluate our methods on three fine-grained datasets, *i.e.*, the Stanford car, the Car-333, and a fine-grained food dataset, containing either hierarchical labels or shared attributes. The experimental results demonstrate that our feature representation can precisely differentiate fine-grained or subordinate classes, and also effectively discover similar images at different levels of relevance, both of which are challenging problems.

The rest of the paper is organized as follows. Section 2 provides a brief review of fine-grained image categorization and the recent approaches of learning fine-grained feature representation. Section 3 introduces our method which learns feature representation by multi-task learning and embedding label structures. Experiments are presented in Section 4, and we conclude the paper in Section 5.

## 2. Related Work

Fine-grained image understanding aims to differentiate subordinate classes. Its main challenges are the following: 1) Many fine-grained classes are highly correlated and are difficult to distinguish due to their subtle differences, *i.e.*, small inter-class variance. 2) On the other hand, the intra-class variance can be large, partially due to different poses and viewpoints. Many methods have been proposed to alleviate these two problems. In this section, we emphasize on the methods that are most relevant to our approaches, particularly the ones on fine-grained feature representation.

Many algorithms have been proposed to leverage parts of objects to improve the classification accuracy. Part based models [46, 7, 3, 51, 50, 15, 43] are proposed to capture the

subtle appearance differences in specific object parts and reduce the variance caused by different poses or viewpoints. Different from these part-based methods, distance metric learning can also address these challenges by learning an embedding such that data points from the same class are clustered together, while those from different classes are pushed apart from each other. In addition, it ensures the flexibility of grouping the same category, such that only a portion of the neighbors from the same class need to be pulled together. For example, Qian et al. [30] proposed a multi-stage metric learning framework that can be applied in large-scale high-dimensional data with high efficiency. In addition to directly classify the images using CNN, it is also possible to generate discriminative features that can be used for classification. In this context, DeCAF [13] is a commonly used feature representation with promising performance achieved by training a deep convolutional architecture on an auxiliary large labeled object database. These features are from the last few fully connected layers of CNN, which have sufficient generalization capacity to perform semantic discrimination tasks using classifiers, reliably outperforming traditional hand-engineered features.

One limitation of the above mentioned methods is that they are essentially driven by the fine-grained class labels for classification, while it is desired to incorporate similarity constraints as well. Therefore, other than using classification constraints alone (*e.g.*, softmax), several similarity constraints have been proposed for feature representation learning. For example, siamese network [10] defines similar and dissimilar image pairs, with the requirement that the distance between dissimilar pairs should be larger than a certain margin, while the one from similar pairs should be smaller. This type of similarity constraint can effectively learn feature representations for various tasks, especially for the verification [42, 33]. An intuitive improvement is to combine the classification and the similarity constraints together for better performance. This is particularly relevant to our framework. For example, [35, 48, 2] proposed to combine the softmax and contrastive loss in CNN via joint optimization. It improved traditional CNN because contrastive constraints might augment the information for training the network. Different from these approaches, our method leverages the triplet constraint [27, 8] instead of the contrastive ones, since triplet can preserve the intra-class variation [31], which is critical to the learning of fine-grained feature representation. Note that triplet constraint has been used in feature learning [39, 23, 38], face representation [31], and person re-identification [12]. Particularly, there are also efforts on combining this with the softmax. A representative example is that [29] proposed to learn a face classify first, and then use the triplet constraint to fine-tune and boost the performance. It achieved promising accuracy in face recognition. Although we also integrate triplet in-

formation with the traditional classification objective, our method jointly optimizes these two objectives simultaneously, which is different from [29]. As shown in the experiments, this joint optimization strategy generates better feature representations for fine-grained image understanding. In addition, our framework can also easily support the embedding of label structures in a unified framework, *e.g.*, hierarchy or shared attributes, which have been proven useful in various studies [5, 14, 1, 37, 44, 49, 9], but not well explored in learning fine-grained feature representation that can model similarity at different levels.

### 3. Methodology

#### 3.1. Jointly Optimize Classification and Similarity Constraints

Traditional classification constraints such as softmax with loss are usually employed in CNN for fine-grained image categorization, which can distinguish different subordinate classes with high accuracy. Suppose that we are given  $N$  training images  $\{r_i, l_i\}_{i=1}^N$  of  $C$  classes, where each image  $r_i$  is labeled as class  $l_i$ . Given the output of the last fully connected layer  $f_s(r_i, c)$  for each class  $c = 1, \dots, C$ , the loss of softmax can be defined as the sum of the negative log-likelihood over all training images  $\{r_i\}_i$ :

$$E_s(r, l) = \frac{1}{N} \sum_{i=1}^N -\log \frac{e^{f_s(r_i, l_i)}}{\underbrace{\sum_{c=1}^C e^{f_s(r_i, c)}}_{P(l_i|r_i)}}, \quad (1)$$

where  $P(l_i|r_i)$  encodes the posterior probability of the image  $r_i$  being classified as the  $l_i$ th class. In a nutshell, Eq. 1 aims to “squeeze” the data from the class into a corner of the feature space. Therefore, the intra-class variance is not preserved, while such variance is essential to discover both visually and semantically similar instances.

To address these limitations, we explicitly model the similarity constraint in CNN using a multi-task learning strategy. Specifically, the triplet loss is fused with the classification objective as the similarity constraint. A triplet consists of three images, denoted as  $(r_i, p_i, n_i)$ , where  $r_i$  is the reference image from a specific class,  $p_i$  an image from the same class, and  $n_i$  an image from a different class. Given an input image  $r_i$  (similarly for  $p_i$  and  $n_i$ ), this triplet-driven network can generate a feature vector  $f_t(r_i) \in \mathbb{R}^D$ , where the hyper-parameter  $D$  is the feature dimension after embedding. Ideally, for each reference  $r_i$ , we expect its distance from any  $n_i$  of different class is larger than  $p_i$  within the same class by a certain margin  $m > 0$ , *i.e.*,

$$\mathcal{D}(r_i, p_i) + m < \mathcal{D}(r_i, n_i), \quad (2)$$

where  $\mathcal{D}(\cdot, \cdot)$  is the squared Euclidean distance between two  $\ell_2$ -normalized vectors  $f_t(\cdot)$  of the triplet network. To

enforce this constraint in CNN training, a common relaxation [27] of Eq. 2 can be defined as the following hinge loss:

$$E_t(r, p, n, m) = \frac{1}{2N} \sum_{i=1}^N \max\{0, \mathcal{D}(r_i, p_i) - \mathcal{D}(r_i, n_i) + m\}. \quad (3)$$

In the feature space defined by  $f_t(\cdot)$ , it can group the  $r$  and  $p$  together while repelling the  $n$  by minimizing  $E_t(r, p, n, m)$ . The gradient can be computed as:

$$\begin{aligned} \nabla W_t &= 2(f_t(r_i) - f_t(p_i)) \frac{\partial f_t(r_i) - \partial f_t(p_i)}{\partial W_t} \\ &\quad - 2(f_t(r_i) - f_t(n_i)) \frac{\partial f_t(r_i) - \partial f_t(n_i)}{\partial W_t}, \end{aligned} \quad (4)$$

if  $\mathcal{D}(r_i, n_i) - \mathcal{D}(r_i, p_i) < m$ , otherwise 0. Different from the pairwise contrastive loss [10] that forces the data of the same class to stay close with a fixed margins, the triplet loss allows certain degrees of intra-class variance. Despite its merits in learning feature representation, minimizing Eq. 3 for recognition tasks still has several disadvantages. For example, given a dataset with  $N$  image, the number of all possible triplets is  $N^3$ , and each triplet contains much less information (*i.e.*, similar or dissimilar constraints with margins) compared with the classification constraint that provides a specific label among  $C$  classes. This can lead to slow convergence. Furthermore, without the explicit constraints for classification, the accuracy of differentiating classes can be inferior to the traditional CNN using softmax, especially in fine-grained problems where the differences of subordinate classes are very subtle.

Given the limitations of training with the triplet loss (Eq. 3) solely, we propose to jointly optimize two types of losses using a multi-task learning strategy. Fig. 2 shows the CNN architecture of our joint learning. The  $R, P, N$  networks share the same parameters during training. After the  $\ell_2$  normalization, the outputs of the three networks (*i.e.*,  $f_t(r), f_t(p), f_t(n)$ ) are transmitted to the triplet loss layer to compute the similarity loss  $E_t(r, p, n, m)$ . In the meantime, the output of the network  $R$ ,  $f_s(r)$ , is forwarded to the softmax loss layer to compute the classification error  $E_s(r, l)$ . Then, we integrate these two types of losses through a weighted combination:

$$E = \lambda_s E_s(r, l) + (1 - \lambda_s) E_t(r, p, n, m), \quad (5)$$

where  $\lambda_s$  is the weight to control the trade-off between two types of losses. We optimize Eq. 5 using the standard stochastic gradient descent with momentum. The final gradient is computed as a  $\lambda$ -weighted combination of  $\nabla W_s$  from the classification constraint and  $\nabla W_t$  from the similarity constraint, and propagated back to lower layers. This framework of unifying three networks through Eq. 5

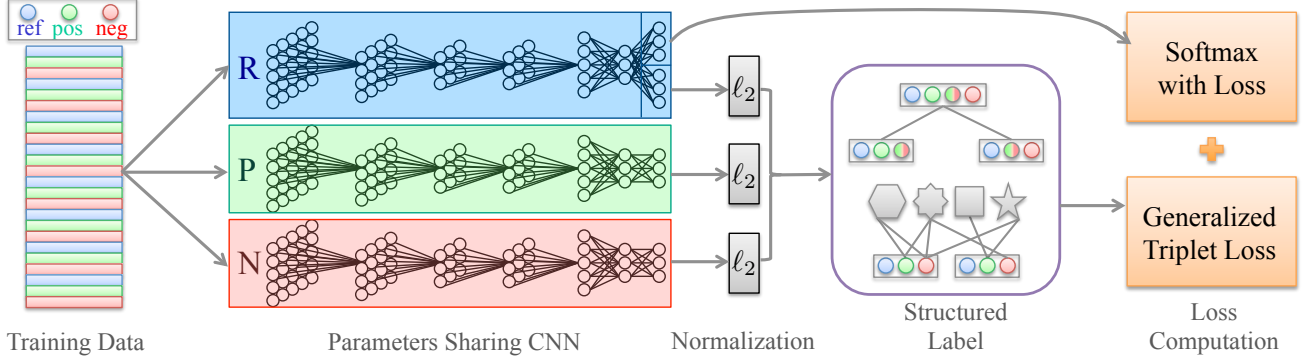


Figure 2: Our framework takes the triplets (*i.e.*, the reference, the positive and the negative images) and the label of the reference image as the input, which pass through the three networks with shared parameters. The label structures are embedded in the loss layer, including the hierarchy or shared attributes. Two types of losses are optimized jointly to obtain the fine-grained classifier and also the feature representation.

not only learns the discriminative features but also preserves the intra-class variance, without sacrificing the classification accuracy. In addition, it resolves the issue of the slow convergence when only using the triplet loss. Regarding the sampling strategy, one can either follow the methods in Facenet [31], or employ hard mining approaches to explore challenging examples in the training data. Both of them are effective in our framework, since jointly optimizing  $E_s(r, l)$  facilitates the searching of good solutions, allowing certain flexibility for the sampling.

During the testing stage, this framework takes one image as an input, and generates the classification result through the softmax layer, or the fine-grained feature representation after the  $l_2$  normalization. This discriminative feature representation can be employed for various tasks such as classification, verification and retrieval, which is more effective than solely optimizing the softmax with loss.

### 3.2. Embed Label Structures

As discussed before, an effective feature representation should be able to search relevant instances at different levels (*e.g.*, Fig. 1), even not within the same fine-grained class. Our framework serves as a baseline to naturally embed label structures, without sacrificing the classification accuracy on fine-grained datasets. In particular, we aim to handle two types of label structures, *i.e.*, hierarchical labels and shared attributes, both of which have wide applications in practice.

#### 3.2.1 Generalized Triplets for Hierarchical Labels

In the first case, the fine-grained labels can be naturally grouped in a tree-like hierarchy based on semantics or domain knowledge. The hierarchy can contain multiple levels. For simplicity purpose, we explain the algorithm with

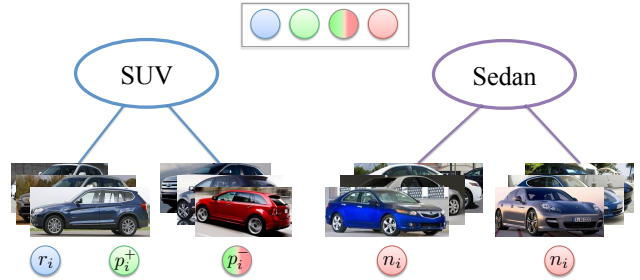


Figure 3: The hierarchy of labels in the fine-grained car dataset [19]. Blue ( $r_i$ ) means the reference image, green ( $p_i^+$ ) denotes the image with the same fine-grained label (*i.e.*, the same make, model and year), green-red ( $p_i^-$ ) represents different fine-grained labels but the same coarse label (*i.e.*, the body type), and red ( $n_i$ ) indicates different coarse labels.

a two-level structure, and then generalize to multiple levels. Fig. 3 illustrates an example of two-level labels from a car dataset [19], where the fine-grained car models in the leaf nodes are grouped according to their body types in the roots.

To model this hierarchy of coarse and fine class labels, we propose to generalize the concept of triplet. Specifically, *quadruplet* is introduced to model the two-level structure. Each quadruplet,  $(r_i, p_i^+, p_i^-, n_i)$ , consists of four images. Similar to triplet,  $p_i^+$  denotes the image of the same fine-grained class as the reference  $r_i$ . The main difference is that in quadruplet, all negative samples are classified into two sub-categories: the more similar one  $p_i^-$  that shares the same coarse class with  $r_i$ , and the more different one  $n_i$  sampled from different coarse classes. Given a quadruplet,

this hierarchical relation among the four images can be described in two inequalities,

$$\mathcal{D}(r_i, p_i^+) + m_1 < \mathcal{D}(r_i, p_i^-) + m_2 < \mathcal{D}(r_i, n_i), \quad (6)$$

where the two hyper-parameters,  $m_1$  and  $m_2$ , satisfying  $m_1 > m_2 > 0$ , control the distance margins across the two levels. It is worth to mention that if Eq. 6 is satisfied, then  $\mathcal{D}(r_i, p_i^+) + m_1 + m_2 < \mathcal{D}(r_i, n_i)$  automatically holds. Compared to triplet, quadruplet is able to model much richer label structures between different levels, *i.e.*, coarse labels and fine-grained labels. As a result, the learned feature representation can discover relevant instances that are appropriate in specific scenarios, *e.g.*, locating a car with specific model and year, or finding SUVs from different body types.

Regarding the sampling strategy, all training images are used as the references in every epoch. For each reference image  $r_i$ , we select  $p_i^+$ ,  $p_i^-$  and  $n_i$  from other corresponding classes, depending on both fine and coarse labels. To incorporate this quadruplet constraint in CNN training, we propose to decompose Eq. 6 into two triplets,  $(r_i, p_i^+, p_i^-)$  and  $(r_i, p_i^-, n_i)$ , phrased as *generalized triplets*. Similar to Eq. 3, our approach seeks for the optimal parameters that minimize the joint loss over the sampled quadruplets:

$$E_q(r, p^+, p^-, n, m_1, m_2) = \frac{1}{2N} \sum_{i=1}^N \max\{0, \mathcal{D}(r_i, p_i^+) - \mathcal{D}(r_i, p_i^-) + m_1 - m_2\} + \frac{1}{2N} \sum_{i=1}^N \max\{0, \mathcal{D}(r_i, p_i^-) - \mathcal{D}(r_i, n_i) + m_2\}. \quad (7)$$

Clearly, this generalized triplets can be naturally incorporated into our multi-task learning framework (Eq. 5).

So far we have mainly discussed in the scenario of a two-level label hierarchy, through the generalized triplet representation of quadruplet. In fact, our method is also applicable to the more general multi-level case using the same strategy, *i.e.*, representing a ‘‘tuple’’ with generalized triplets. Similar to the quadruplet sampling strategy, each tuple is formed by selecting the classes at different similarity levels, from which training images are sampled (one image at each level). Therefore, a tuple from an  $x$ -level hierarchy contains  $x + 2$  images (*e.g.*, the quadruplet from a two-level hierarchy has four images). This tuple is decomposed into  $x$  triplets, by taking the reference image and two more images from two adjacent levels. Intuitively, this means that multiple triplets are sampled to represent different levels of similarity, *i.e.*, images with the same finer-level labels are more similar than ones sharing the same coarser-level labels. Same as the two-level case, it can be optimized using the multi-task learning framework based on triplets. Even

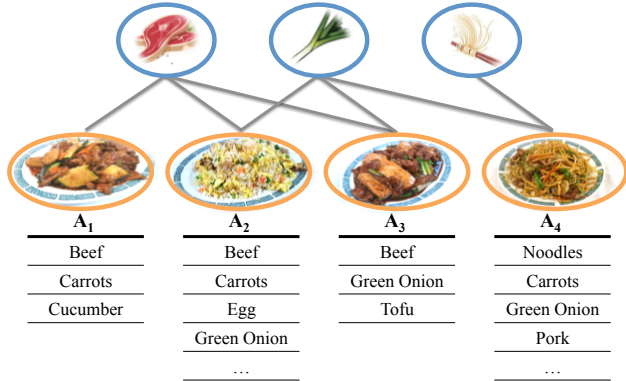


Figure 4: The shared attributes in our food dataset, where the attributes ( $A_1$ - $A_4$ ) mean the ingredients.

though this is not exhaustive sampling or exact decomposition for the tuple, the generalized triplets are representative enough to ensure a good performance, which is demonstrated in our experiments (Section 4.2). It is also worth mentioning that the traditional triplet is a special case of the generalized triplet, *i.e.*, only one-level hierarchy.

### 3.2.2 Generalized Triplets for Shared Attributes

In the second case, fine-grained objects can share common attributes with each other. For instance, Fig. 4 illustrates that fine-grained food dishes can share some ingredients, indicating relevance at different levels. Intuitively, classes that share more attributes should be more similar than the classes sharing less attributes. Unlike the tree-like hierarchy in the first case, we are not able to directly model the label dependency as Eq. 6, because some fine-grained classes can own multiple attribute labels. Instead, we model this graph dependency using a modified triplet idea. To have a better understanding of our method, we can consider the first three dishes shown in Fig. 4. Although both the second and third dishes belong to different classes compared to the first one, the second dish shares more attributes (beef, carrots) with the first dish. This difference in attribute overlapping inspires us to re-define the margin  $m$ , *i.e.*, the distance between  $\mathcal{D}(r_i, p_i)$  and  $\mathcal{D}(r_i, n_i)$ , as the Jaccard similarity [16] of attributes from different classes:

$$m = m_b \left( 1 - \frac{|A_p \cap A_n|}{|A_p \cup A_n|} \right), \quad (8)$$

where  $m_b$  is a constant factor specified as the base margin,  $A_p$  and  $A_n$  are the sets of attributes belonging to the positive and negative categories, respectively. Therefore, the more attributes these classes share, the smaller margin this triplet has. Using such adaptive margin for the triplet loss, the learned feature can discover images containing common attributes as the query images. Similarly, Eq. 8 can

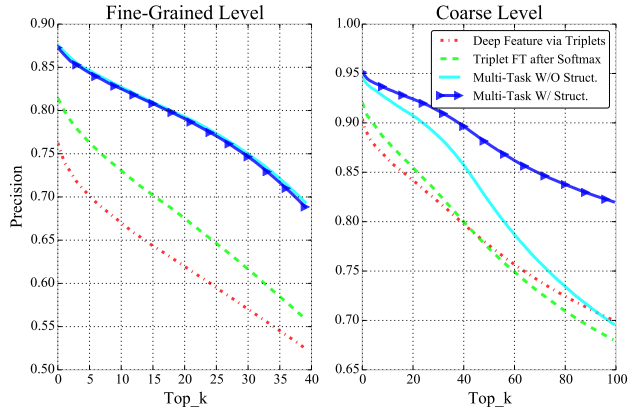


Figure 5: Comparison of retrieval precision on the Stanford car, with two levels of labels.

be naturally incorporated in our multi-task learning framework based on the triplet loss. In fact, the original triplet constraint is also a special case of the multi-attribute constraint, when each fine-grained label only connects to one attribute.

## 4. Experiments

In this section, we conduct thorough experiments to evaluate this proposed framework on three fine-grained datasets with label structures. Particularly, we aim to demonstrate that our learned feature representations can be used to retrieve images at different levels of relevance, with significantly higher precision than other CNN-based methods. In addition, we also report its promising classification accuracy on these fine-grained classes.

We focus on the comparison of four methods that can generate fine-grained feature representation: 1) deep feature learning by triplet loss [31, 39], 2) triplet-based fine-tuning after softmax [29], *i.e.*, not joint optimization, 3) our multi-task learning framework, and 4) our framework with label structures. In the classification task, besides these four methods, we also report the accuracy of using CNN with traditional softmax. All CNNs are based on GoogleNet [36], and are fine-tuned on these fine-grained datasets for the best performance and fair comparisons. We also carefully follow the specifications from these compared papers for their settings and parameters. Regarding our hyper-parameters, we empirically set the feature dimension as 200, the margin as 0.2, and the weight  $\lambda_s$  as 0.8, with discussions of the sensitivity in Section 4.4.

### 4.1. Stanford Car with Two-Level Hierarchy

The first experiment focuses on the efficacy of embedding hierarchical labels, using the Stanford car dataset [19]. It contains 16,185 images (with bounding boxes) of 196 car

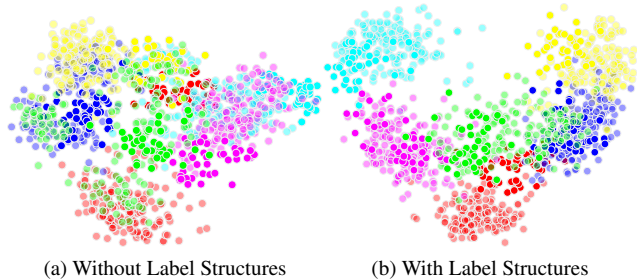


Figure 6: Visualization of features after dimension reduction. Different colors represents different coarse-level labels, and intensities (or transparency) from the same color indicate fine-grained labels.

categories, with 8,144 for training and the rest for testing. The categories, *i.e.*, fine-grained class labels, are defined as make, model and year, such as Audi S4 Sedan 2012. Following [19], we have assigned each fine-grained label to one of nine coarse body types, such as SUV, Coupe and Sedan (Fig. 3 in [19]), resulting in a two-level hierarchy.

Fig. 5 shows the retrieval precision using feature representations extracted by various CNNs, at both the fine-grained level and the coarse level. At the fine-grained level, results from our multi-task learning methods are better than the others, *i.e.*, at least 13.5% higher precision at top-40 retrievals (using top-40 since each fine-category has around 40 images). The reason is that the joint optimization strategy leverages the similarity constraints via triplets, which can augment the training information, assisting the network to reach better solutions. No matter using the traditional or generalized triplets (*i.e.*, without or with the label structures) in our framework, the difference of precision is within 0.5%, which can be caused by the sampling strategies. At the coarse level, our method without label structures also fails to achieve high precision at top-100 retrievals, while using generalized triplets significantly outperforms the others, *i.e.*, at least 12.4% higher precision, demonstrating the efficacy of our embedding scheme. To provide insights of our promising results on this coarse-level retrieval, we extract features from our multi-task learning framework using traditional and generalized triplets, and visualize them in Fig. 6 after dimension reduction. Six coarse-level classes are randomly chosen, and five fine-level classes are sampled from each coarse one. The features from generalized triplets are consistently much better separated than ones from traditional triplets, benefited from the embedding of label structures.

We also report the classification accuracy of these methods on fine-grained classes. A fine-tuned GoogleNet achieves 86.9%. Learning deep features via triplets alone [31, 39] attains 78.7%, which is worse than

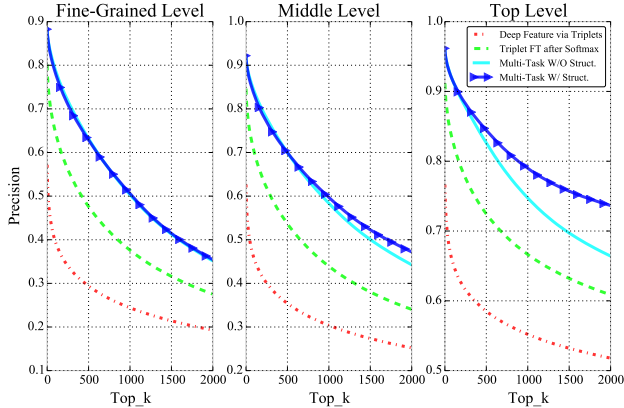


Figure 7: Comparison of retrieval precision on the Car-333 dataset. Top-level means the car make only. Mid-level represents both make and model. Fine-level denotes the fine-grained labels of make, model and year range.

GoogleNet. The reason is that softmax with loss can explicitly minimize the classification error, while triplets attempt to implicitly separate classes by constraining the similarity measures. Fine-tuning with triplets after the softmax [29] also aims to integrate the classification and similarity constraints, same as ours. This identification and verification framework achieves promising performance in face recognition. However, different from our framework, it embeds the triplet loss after learning a face classifier, *i.e.*, not a joint optimization strategy as ours. This may adversely affect the classification accuracy in fine-grained image categorization, since triplet loss only implicitly constrains the classification error, which may not be sufficient in further differentiating subordinate classes during fine-tuning. As a result, it achieves 83.0%, which is worse than the fine-tuned GoogleNet. Our multi-task learning framework achieves 88.4% when jointly optimizing both types of losses, which are higher than these compared methods, and among state-of-the-art results that do not use parts.

## 4.2. Car-333 with Three-Level Hierarchy

The second experiment also investigates the hierarchical labels, but using a much larger car dataset [44] to validate the scalability. These are end-user photos from the Craigslist, so they are more naturally photographed. It contains 157,023 training images and 7,840 testing images, from 333 car categories. The categories are defined by make, model and year range. Note that two cars of the same model but from different years are considered as different classes. The bounding boxes are generated by Regionlets [40], which produces promising results in car detection. Different from the Stanford car, this has a three-level hierarchy: 333 fine-grained labels are grouped into 140 models by ignoring the difference of years, and then

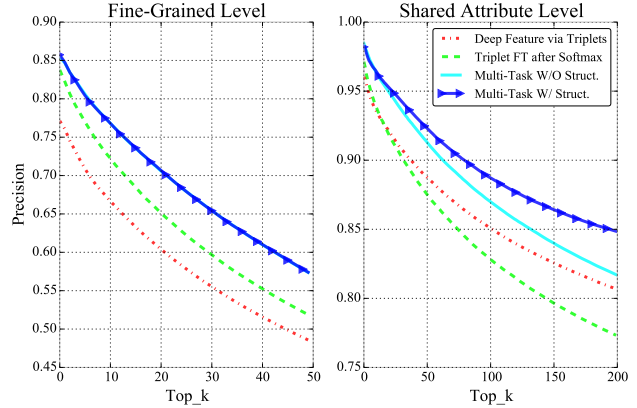


Figure 8: Comparison of retrieval precision on the food dataset. Share Attribute Level means that two images are relevant if they share at least one attribute.

five makes (*i.e.*, Chevrolet, Ford, Honda, Nissan, Toyota).

Fig. 7 shows the retrieval precision at these three levels. Since the training data is around 20 times larger than the previous one, we show the precision upon top-2000 retrievals (note that the number of images in a fine-level class can be less than 2000). The results are consistent with the ones on the Stanford car, demonstrating that the strategy of generalized triplets is applicable to multi-level hierarchies. Specifically, our method with label structures is at least 13.2% better than other methods in terms of the top-2000 retrieval precision at the middle level, and 12.8% better at the top level. This is also 7.2% better than ours without embedding structures at the top level, proving the efficacy of our generalized triplets. In addition, such promising results also demonstrate that the scalability of our methods such as generalized triplets is sound. Regarding the classification accuracy, GoogleNet achieves 87.9%, the deep feature via traditional triplets attains 61.2%, fine-tuning with triplets after softmax reaches 81.7%. It is worth mentioning that the deep feature via triplets has considerably worse performance on this dataset, compared to the results on the Stanford car. It indicates that this method does not have good scalability for fine-grained image categorization, although it is proven to be effective for other tasks such as verification and ranking [31, 39]. On the other hand, jointly optimizing the softmax with loss can alleviate this issue even on this larger-scale dataset, as it directly tackles the classification problem. Using this strategy, our method achieves 89.4%, which is among state-of-the-art.

## 4.3. Food Dataset with Shared Attributes

The third experiment aims to examine the embedding of shared attributes, using our newly collected food dataset that consists of ultra-fine-grained classes and rich class relationships. To generate this dataset, we sent multiple data

collectors to six restaurants, and they took photos of most dishes during two months. In total, we acquired 37,086 food photos from 975 menu items, *i.e.*, fine-grained class labels. In addition, we built a list of 51 ingredients, *i.e.*, shared attributes, to precisely describe these dishes. This dataset is divided into 32,135 training and 4,951 testing images, and testing images are collected on different days from the training, to mimic a realistic scenario by avoiding potential correlations of taking photos in the same day (*e.g.*, multiple photos from the same dish at the same time cannot be used for both training and testing).

Fig. 8 shows the retrieval precision on this food dataset with respect to top-50 retrievals, as each category has around 20 to 50 images. In addition to evaluate on the fine-grained labels, we also define a new level of relevance: two images are similar when they share at least one attribute. Our method by embedding shared attributes outperforms the others by 5.5% at the fine-grained level, and 4.2% at the attribute level in terms of the precision. Since the precisions of these methods are already above 80%, such improvement means a reducing of 21.7% for the errors. Compared to our method without embedding attributes, it is nearly the same performance at the fine-grained level, while 3.1% better at the attribute level (reducing errors by 16.9%), demonstrating the efficacy of the generalized triplets with adaptive margins. Note that the improvement may not be as significant as on the other two datasets using hierarchical labels. The reason is that the similarity measure for attributes is more subtle, *i.e.*, two cars having different coarse labels could be more distinguishable than two dishes sharing no attributes. In terms of the classification accuracy, we have achieved 89.0%, comparing to 87.1% by GoogleNet, 78.2% by learning the deep feature and 86.1% by fine-tuning with triplets after softmax. This is also a promising result, considering that this challenging dataset is ultra-fine-grained.

#### 4.4. Discussions

Fig. 9 shows the convergence rate of these methods. Since each triplet contains much less information compared to the one of using the label directly (*i.e.*, softmax with loss), their convergence rates can be dramatically different. Particularly, using softmax with loss has much faster convergence rate than using triplet loss. Our multi-task learning framework jointly minimizes both of them, so it harvests augmented information from both sides, resulting in a fast convergence rate as well. Overall, our methods converge after 800 epochs on the Stanford car, 150 epochs on the Car-333, and 600 epochs on the food dataset, which are reasonably fast in practice.

Our framework has one important parameter, the weight  $\lambda_s$  to balance two types of losses, and setting  $\lambda_s$  to be 0 or 1 degenerates our framework to deep feature learning by triplet loss [31, 39] or GoogleNet (softmax with loss), re-

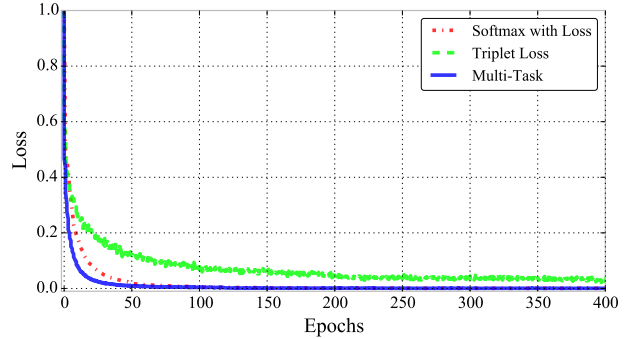


Figure 9: Comparison of the convergence rate on the Stanford car dataset. The first 400 epochs are shown for better visualization.

spectively, which will either fail to differentiate fine-grained classes or lose the ability to generate effective feature representations. Since softmax with loss may contain more information than a triplet in each iteration, it is reasonable to assign a higher weight to softmax, *i.e.*, larger than 0.5. Our experiments show that the performance is not sensitive to small variations to  $\lambda_s$ , *i.e.*, within 0.8% difference in a range of [0.55, 0.85]. Besides the weight, the feature dimension and the margin is also relevant to the classification accuracy. From our extensive experiments, we observe that our methods are also stable with respect to their variations up to a certain range, *e.g.*, within 2% difference for feature dimensions from 128 to 512. Therefore, it is relatively easy to tune the hyper-parameters in our framework. In fact, we use the same group of parameters on all datasets.

## 5. Conclusion

In this paper, we proposed a multi-task learning framework to effectively generate fine-grained feature representations by embedding label structures, such as hierarchical labels or shared attributes. In our method, the label structures are seamlessly embedded in CNN through the proposed generalized triplets, which can incorporate the similarity constraints at different levels of relevance. Such a framework retains the classification accuracy for subordinate classes with subtle differences, and at the same time considerably improves the image retrieval precision at different levels of label structures on three fine-grained datasets, including a newly-collected benchmark dataset for food. These merits warrant further investigating the embedding of label structures for learning fine-grained feature representation.



## References

- [1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *CVPR*, pages 819–826. IEEE, 2013. 3
- [2] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. In *SIGGRAPH*, 2015. 2
- [3] T. Berg and P. N. Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, pages 955–962. IEEE, 2013. 1, 2
- [4] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *CVPR*, pages 2019–2026. IEEE, 2014. 1
- [5] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, pages 663–676. Springer, 2010. 3
- [6] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101—mining discriminative components with random forests. In *ECCV*, pages 446–461. Springer, 2014. 1
- [7] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, pages 321–328. IEEE, 2013. 1, 2
- [8] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *JMLR*, 11:1109–1135, 2010. 2
- [9] Q. Chen, J. Huang, R. Feris, L. M. Brown, J. Dong, and S. Yan. Deep domain adaptation for describing people based on fine-grained clothing attributes. In *CVPR*, pages 5315–5324, 2015. 3
- [10] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, volume 1, pages 539–546. IEEE, 2005. 2, 3
- [11] J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *CVPR*, pages 580–587. IEEE, 2013. 1
- [12] S. Ding, L. Lin, G. Wang, and H. Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 2015. 2
- [13] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In T. Jebara and E. P. Xing, editors, *ICML*, pages 647–655, 2014. 2
- [14] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR*, pages 3474–3481. IEEE, 2012. 3
- [15] C. Goering, E. Rodner, A. Freytag, and J. Denzler. Nonparametric part transfer for fine-grained recognition. In *CVPR*, pages 2489–2496. IEEE, 2014. 1, 2
- [16] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912. 5
- [17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, pages 675–678. ACM, 2014. 1
- [18] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, 2011. 1
- [19] J. Krause, J. Deng, M. Stark, and L. Fei-Fei. Collecting a large-scale dataset of fine-grained cars. 2013. 1, 4, 6
- [20] J. Krause, H. Jin, J. Yang, and L. Fei-Fei. Fine-grained recognition without part annotations. In *CVPR*, pages 5546–5555, 2015. 1
- [21] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*, pages 554–561. IEEE, 2013. 1
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1
- [23] H. Lai, Y. Pan, Y. Liu, and S. Yan. Simultaneous feature learning and hash coding with deep neural networks. *CVPR*, 2015. 2
- [24] D. Lin, X. Shen, C. Lu, and J. Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *CVPR*, pages 1666–1674, 2015. 1
- [25] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. *ICCV*, 2015. 1
- [26] Y.-L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis. Jointly optimizing 3d model fitting and fine-grained classification. In *ECCV*, pages 466–480. Springer, 2014. 1
- [27] M. Norouzi, D. M. Blei, and R. R. Salakhutdinov. Hamming distance metric learning. In *NIPS*, pages 1061–1069, 2012. 2, 3
- [28] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505. IEEE, 2012. 1
- [29] O. M. Parkhi, A. Vedaldi, A. Zisserman, A. Vedaldi, K. Lenc, M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, K. Lenc, et al. Deep face recognition. *BMVC*, 2015. 2, 3, 6, 7
- [30] Q. Qian, R. Jin, S. Zhu, and Y. Lin. Fine-grained visual categorization via multi-stage metric learning. In *CVPR*, pages 3716–3724, 2015. 1, 2
- [31] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *CVPR*, 2015. 2, 4, 6, 7, 8
- [32] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 1
- [33] G. Sharma and B. Schiele. Scalable nonlinear embeddings for semantic category-based image retrieval. *ICCV*, 2015. 2
- [34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1
- [35] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, pages 1988–1996, 2014. 2
- [36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CVPR*, 2015. 1, 6

- [37] A. Vedaldi, S. Mahendran, S. Tsogkas, S. Maji, R. Girshick, J. Kannala, E. Rahtu, I. Kokkinos, M. B. Blaschko, D. Weiss, et al. Understanding objects in detail with fine-grained attributes. In *CVPR*, pages 3622–3629. IEEE, 2014. [3](#)
- [38] C. Wah, G. Van Horn, S. Branson, S. Maji, P. Perona, and S. Belongie. Similarity comparisons for interactive fine-grained categorization. In *CVPR*, pages 859–866. IEEE, 2014. [1](#), [2](#)
- [39] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, pages 1386–1393. IEEE, 2014. [1](#), [2](#), [6](#), [7](#), [8](#)
- [40] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, pages 17–24. IEEE, 2013. [7](#)
- [41] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10:207–244, 2009. [1](#)
- [42] L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*. IEEE, 2014. [2](#)
- [43] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. *CVPR*, 2015. [2](#)
- [44] S. Xie, T. Yang, X. Wang, and Y. Lin. Hyper-class augmented and regularized deep learning for fine-grained image classification. In *CVPR*, volume 580, 2015. [3](#), [7](#)
- [45] L. Yang, P. Luo, C. C. Loy, and X. Tang. A large-scale car dataset for fine-grained categorization and verification. In *CVPR*, pages 3973–3981. IEEE, 2015. [1](#)
- [46] S. Yang, L. Bo, J. Wang, and L. G. Shapiro. Unsupervised template learning for fine-grained object recognition. In *NIPS*, pages 3122–3130, 2012. [1](#), [2](#)
- [47] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar. Food recognition using statistics of pairwise local features. In *CVPR*, pages 2249–2256. IEEE, 2010. [1](#)
- [48] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. [2](#)
- [49] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, pages 192–199. IEEE, 2014. [3](#)
- [50] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *ECCV*, pages 834–849. Springer, 2014. [1](#), [2](#)
- [51] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, pages 729–736. IEEE, 2013. [1](#), [2](#)