Grafit: Learning fine-grained image representations with coarse labels

Hugo Touvron[1,2] Alexandre Sablayrolles [1] Matthijs Douze [1] Matthieu Cord [2] Hervé Jégou [1]

[1] Facebook AI Research [2]Sorbonne University

Abstract

This paper tackles the problem of learning a finer representation than the one provided by training labels. This enables fine-grained category retrieval of images in a collection annotated with coarse labels only.

Our network is learned with a nearest-neighbor classifier objective, and an instance loss inspired by self-supervised learning. By jointly leveraging the coarse labels and the underlying fine-grained latent space, it significantly improves the accuracy of category-level retrieval methods.

Our strategy outperforms all competing methods for retrieving or classifying images at a finer granularity than that available at train time. It also improves the accuracy for transfer learning tasks to fine-grained datasets.

1. Introduction

Image classification now achieves a performance that meets many application needs [27, 37, 54]. However, in practice, datasets and labels available at training time do not necessarily correspond to those needed in subsequent applications [17]. The granularity of the training-time concepts may not suffice for fine-grained downstream tasks. This has encouraged the development of specialized classifiers offering a more precise representation. Fine-grained classification datasets [29] have been developed for specific domains, for instance to distinguish different plants [13] or bird species [59].

Gathering a sufficiently large collection with fine-grained labels is difficult by itself, as it requires to find enough images of rare classes, and annotating them precisely requires domain specialists with in-domain expertise. This is evidenced by the Open Images construction annotation protocol [38] that states that: "*Manually labeling a large number of images with the presence or absence of 19,794 different classes is not feasible*". For this reason they resorted to computer-assisted annotation, at the risk of introducing biases due to the assisting algorithm.

To circumvent this issue, we propose in this paper a strategy to get strong classification and image retrieval performance on fine concepts using only coarse labels at training. Our work leverages two intuitions. First, in order to improve the granularity beyond the one provided by image labels, we need to exploit another signal than just the labels. For this purpose, we build upon recent works [3, 62] that exploit *two* losses to address both image classification and instance recognition, leveraging the "free" annotations provided by multiple data augmentations of a same instance, in the spirit of self-supervised learning [6, 9, 10, 25].

The second intuition is that it is better to explicitly infer coarse labels even when classifying at a finer granularity. For this purpose, we propose a simple method that exploits both a coarse classifier and image embeddings to improve fine-grained category-level retrieval. This strategy outperforms existing works that exploit coarse labels at training time but do not explicitly rely on them when retrieving finer-grained concepts [61].

By these ways our method liberates the data collection process from the quirks of a rigid fine-grained taxonomy as previously discussed. To validate our strategy, we investigate two challenging use-cases:

On-the-fly classification. For this task, the fine-grained labels are available at test time only, and we use a non-parametric kNN classifier [61] for on-the-fly classification, *i.e.* without training on the fine-grained labels.

Category-level Retrieval. Given a collection of images annotated with coarse labels, like a product catalog, we aim at ranking these images according to their fine-grained semantic similarity to a new query image outside the collection, as illustrated by Figure 1. We believe that this new task better is more realistic than the on-the-fly classification setting.

In summary, in this context of coarse-to-fine representation learning, our paper makes the following contributions:

- We propose Grafit, a method to learn image representations at a finer granularity than the one offered by the annotation at training time. Inspired by the recent self-supervised BYOL [25] instance learning approach, we

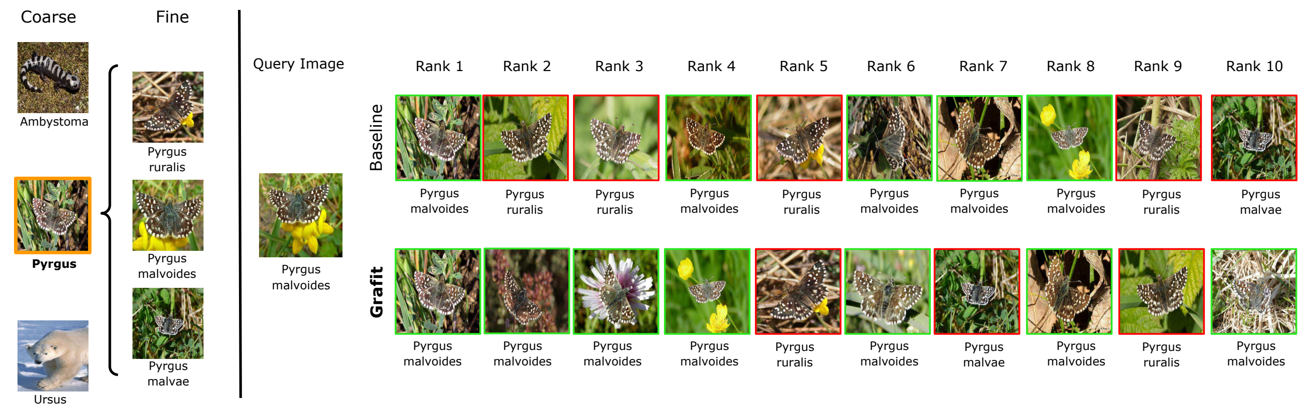


Figure 1: Category-level retrieval orders images based on their semantic similarity to a query. Our Grafit method, although it has used only coarse labels (like 'pyrgus') at training time, produces a ranking consistent with fine-grained labels. Unsupervised learning is a particular case of this task, in which the set of coarse labels is reduced to a singleton. Image credit: [1].

carefully design a joint learning scheme integrating instance and coarse-label based classification losses. For the latter one, we exploit a knn strategy but with a dedicated process to manage the memory both at train-time and for inference at test-time.

- We propose two original use-cases to deeply evaluate coarse-trained fine-grained testing evaluation, for which Grafit exhibits outstanding performance. For instance, we improve by **+16.3%** the top-1 accuracy for on-the-fly classification on ImageNet. This improvement is still +9.5% w.r.t. our own stronger baseline, everything being equal otherwise. Grafit also improves transfer learning: our experiments show that our representation discriminates better at a finer granularity.

This paper is organized as follows. After reviewing related works in Section 2, we present our method in Section 3. Section 4 compares our approach against baselines on various datasets, and presents an extensive ablation. Section 5 concludes the paper. In the supplemental material, Appendix A summarizes two experiments that show how an instance-level loss improves the granularity beyond the one learned by a vanilla cross-entropy loss. Appendix B complements our experimental section 4 with more detailed results. Appendix C provides visual results associated with different levels of training/testing granularities.

2. Related work

Label granularity in image classification. In computer vision, the concept of granularity underlies several tasks, such as fine-grained [13, 29] or hierarchical image classification [18, 60, 65]. Some authors consider a formal definition of granularity, see for instance Cui et al. [15]. In our paper, we only consider levels of granularity relative to each other, where each coarse class is partitioned into a set of finer-grained classes.

In some works on hierarchical image classification [19, 26, 45, 49], a coarse annotation is available for all training images, but only a subset of the training images are labelled at a fine granularity. In this paper we consider the case where no fine labels at all are available at training time.

Train-Test granularity discrepancy. A few works consider the case where the test-time labels are finer than those available at training time and where each fine label belongs to one coarse label. Approaches to this task are based on clustering [61] or transfer learning [33]. Huh et al. [33] address the question: "is the feature embedding induced by the coarse classification task capable of separating finer labels (which it never saw at training)?" To evaluate this, they consider the 1000 ImageNet classes as fine, and group them into 127 coarse classes with the WordNet [20] hierarchy. Wu et al. [61] evaluate on the 20 coarse classes of CIFAR-100 [36] and on the same subdivision of ImageNet into 127 classes. They evaluate their method, Scalable Neighborhood Component Analysis (SNCA), with a kNN classifier applied on features extracted from a network trained with coarse labels. Note that this work departs from the popular framework of object/category discovery [11, 21, 32, 57, 58], which is completely unsupervised.

In our work we mainly compare to the few works that consider coarse labels at train time, therefore SNCA [61] is one of our baseline. We adopt their coarse labels definition and evaluation procedure for on-the-fly classification.

Unified embeddings for classes and instances. Similar to Wu et al. [61], several Distance Metric Learning (DML) approaches like the Magnet loss [44] or ProxyNCA [40, 51] jointly take into account intra- and inter-class variability.

This improves transfer learning performance and favors in some cases the emergence of finer hierarchical concepts. Berman et al. proposed Multigrain [3], which simply adds to the classification objective a triplet loss that pulls together different data-augmentations of a same image. Recent works on semi-supervised learning [4, 5, 48, 62, 66, 69] rely on both supervised and self-supervised losses to get information from unlabelled data. For instance the approach of Xie et al. [62] is similar to Multigrain, except that the Kullback-Leibler divergence replaces the triplet loss. Matching embeddings of the same images under different data-augmentations is the main signal in current works on self-supervised learning, which we discuss now.

Unsupervised and Self-Supervised Learning. In unsupervised and self-supervised approaches [9, 10, 22, 25, 34, 56] the model is trained on unlabeled data. Each image instance is considered as a distinct class and the methods aim at making the embeddings of different data-augmentations of a same instance more similar than those of other images. To deal with finer semantic levels than those provided by the labels, we use an approach similar to BYOL [25]. BYOL only requires pairs of positive elements (no negatives), more specifically different augmentations of the same image. A desirable consequence is that this limits contradictory signals on the classification objective.

Transfer Learning. Transfer learning datasets [7, 35, 41] are often fine grained and rely on a feature extractor pre-trained on another set of classes. However, the fine labels are not a subset of the pre-training labels, so we consider transfer learning as a generalization of our coarse-to-fine task. It is preferable to pre-train on a domain similar to the target [16], *e.g.,* pre-training on iNaturalist [29] is preferable to pre-training on ImageNet if the final objective is to discriminate between species of birds. The impact of pre-training granularity is discussed in prior works [15, 67]. In Section 4.6 we investigate how Grafit pre-training performs on fine-grained transfer learning datasets.

3. Grafit: Fitting a finer granularity

Figure 2 depicts our approach at training time. In this section, we discuss the different components and training losses. Then, we detail how we produce the category-level ranking, and how we perform on-the-fly classification.

3.1. Training procedure: Grafit and Grafit FC

We first introduce an instance loss inspired by BYOL [25] that favors fine-grained recognition. The Grafit model includes a trunk network f_θ, to which we add two multi-layers perceptrons (MLP): a "projector" P_θ and a predictor q_θ. In the Grafit FC variant, P_θ is linear for a more direct fair comparison with Wu et al. [61]'s projector. The

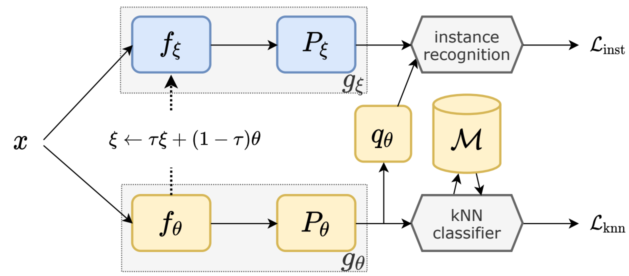


Figure 2: Illustration of our method at train time. The convnet trunk that receives gradient is f_θ and is used to update the target network f_ξ as a moving average. The database of neighbors is updated by averaging embedding in each mini-batch with corresponding embeddings in the database.

learnable parameters are represented by the vector θ. As in BYOL we define a "target network" f_ξ as an exponential moving average of the main network f_θ: the parameters ξ are not learned, but computed as $\xi \leftarrow \tau\xi + (1-\tau)\theta$, with a target decay rate $\tau \in [0, 1]$.

Instance loss. Each image x is transformed by T data augmentations $(t_1, \dots, t_T)$. Denoting $\cos$ the cosine similarity and $g_\theta(x) = P_\theta(f_\theta(x))$, the instance loss is:

$$\mathcal{L}_{\text{inst}}(x) = -\sum_{1\leq i\neq j\leq T} \frac{\cos\left(q_\theta \circ g_\theta(t_i(x)), g_\xi(t_j(x))\right)}{T(T-1)}), \tag{1}$$

The instance loss allows the network to discriminate at the instance level, which is a finer granularity than the class level. We give more insights about this loss in Appendix A.

kNN loss. A parametric classifier with softmax yields a representation that does not generalize naturally to new classes [61] and is not adapted for kNN classification. Therefore, inspired by the neighborhood component analysis [23, 39, 47], Wu et al. [61] propose a loss function optimized directly for kNN evaluation, that we adopt and denote by $\mathcal{L}_{\text{knn}}$. Let x_i be a training image with coarse label y_i and σ a temperature hyper-parameter. For each image x_i we select $x_j (j \neq i)$ as its neighbor with probability $p_{i,j}$, computed as

$$p_{i,j} \propto \exp\left(\cos(g_\theta(x_i), g_\theta(x_j))/\sigma\right), \tag{2}$$

where the $p_{i,j}$ are normalized so that $\sum_{j\neq i} p_{i,j} = 1$. The loss is then defined as:

$$\mathcal{L}_{\text{knn}}(x_i, y_i) = -\log \sum_{j, y_j = y_i, j\neq i} p_{i,j}. \tag{3}$$

We ℓ_2-normalize after the P_θ projection. The $\mathcal{L}_{\text{knn}}$ scores all classes with Equation 3.

Memory of embeddings. One of the limitations of the kNN approach is that it requires to use all the features of the training set. To avoid recomputing all the embeddings of the training set, we use a memory $\mathcal{M} = \{m_1, \ldots, m_i, \ldots\}$. It is updated as follows: when the image x_i in the training set is in the current mini-batch, we update its embedding m_i as follows: $m_i \leftarrow \frac{1}{2}(m_i + g_\theta(x_i))$. In order to limit the memory space needed, we apply the $\mathcal{L}_{\text{knn}}$ loss on the space of the projected features, which allows us to store smaller embedding and hence requires less memory. For instance for ImageNet we have to store 1.2M training images. Without the projection with ResNet-50 architecture for f_θ, the memory size is $2048 \times 1.2M$ but with a projection on a space of size 256 the memory size is $256 \times 1.2M$ what is $\times 8$ smaller.

Combined loss. Our method is summarized in Figure 2. The total loss at training time for an image x with label y is:

$$\mathcal{L}_{\text{tot}}(x) = \mathcal{L}_{\text{knn}}(g_\theta(x), y) + \mathcal{L}_{\text{inst}}(x). \tag{4}$$

Appendix B empirically shows that weighting differently the losses does not bring much difference.

Adapting the architecture at test-time. The training parameters include the model weights (f_θ, P_θ) and the parameters related to $\mathcal{L}_{\text{inst}}$ (f_ξ, P_ξ and q_θ) as described previously. At test time we remove the $\mathcal{L}_{\text{inst}}$ branch, keeping only f_θ and P_θ. In order to have consistent representations of all the training images with the final weights, we re-compute $m_i = g_\theta(x_i)$ for each training image x_i and store it in $\mathcal{M}$.

3.2. Category-level retrieval

For a given test image x' the task is to order by semantic relevance all images from the training collection. In our coarse-to-fine case, a search result is deemed correct if it has the same fine label as the query.

Cosine-based ranking. The standard strategy to order the images is to compute $g_\theta(x')$, and to order all images x_i in the collection by they cosine similarity score $\cos(g_\theta(x_i), g_\theta(x'))$ to the query (the $g_\theta(x_i)$ are precomputed in $\mathcal{M}$). The experiments in Section 4 show that the way Grafit embeddings are trained already improves the ranking with that method.

Ranking conditioned by coarse prediction. Let x' be a test image and x a training image with coarse class y. Let $p_{\text{c}}(x, y)$ be the probability that the image x has coarse label y according to our classifier. Our conditional score ψ_{cond} is a compromise between the embedding similarity and the coarse classification, in spirit of the loss in Equation 4:

$$\psi_{\text{cond}}(x', x) = \cos\left(g_\theta(x'), g_\theta(x)\right) + \log\left(\frac{p_{\text{c}}(x', y)}{1 - p_{\text{c}}(x', y)}\right). \tag{5}$$

Note that, in that case, we rely on the fact that the collection in which we search is the training set, so that the coarse labels associated with the collection are known. In Section 4 we show experimentally that ψ_{cond} improves the category-level retrieval performance in the coarse-to-fine context.

Conditional ranking: Oracle. If we assume that the coarse label of the query test image is known (given by an oracle), then we can set $p_{\text{c}}(x', y) = 1_{y=y'}$ with y' the coarse class of the test image x'. This boils down to systematically putting images with the same coarse class as the test image first in the ranking. Experimentally, this shows the impact of test label prediction on the score, and provides an upper bound on the performance of the conditional ranking strategy. It is also relevant in practice in a scenario where the user provides this coarse labeling, for instance by selecting it from an interface.

3.3. On-the-fly classification

In on-the-fly classification, a kNN classifier "knows" about the fine classes of the training images only at test time [61]. Such a non-parametric classification does not require any training or fine-tuning. As a side note, this flexible classifier can handle settings with evolving datasets, including dynamic additions of new classes, although such setups are outside the scope of this paper.

For a test image x we compute the embedding $g_\theta(x)$ and compare it to the training image embeddings stored in $\mathcal{M}$. We select the k embeddings maximizing the cosine similarity to the query, $(x_1, ..., x_k)$, with labels $(y_1, ..., y_k)$. For a direct comparison with Wu et al. [61] and consistently with Equation 3, we apply an exponentially decreasing neighbour weighting that computes the probability that x belongs to class y as

$$p_{\text{kNN}}(x, y) \propto \sum_{j=1, y_j=y}^{k} \exp\left(\cos(g_\theta(x), g_\theta(x_j))/\sigma\right). \tag{6}$$

We normalize the probabilities so that $\sum_y p_{\text{kNN}}(x, y) = 1$.

4. Experiments

We consider evaluation scenarios where it is beneficial to learn at a finer granularity than that provided by the training labels. The first two tasks are coarse-to-fine tasks (category-level retrieval and on-the-fly classification), where we measure the capacity of the network to discriminate fine labels without having seen them at training time. The third protocol is vanilla transfer learning, where we transfer from Imagenet to a fine-grained dataset.

4.1. Datasets and evaluation metrics

We carry out our evaluations on public benchmarks, which statistics are detailed in Table 1.

Table 1: Datasets used for our different tasks. The four top datasets offer two or more levels of granularity, we use them for all coarse-to-fine tasks. The bottom three are fine-grained datasets employed to evaluate transfer learning.

Dataset	Train size	Test size	#classes
CIFAR-100 [36]	50,000	10,000	20/100
ImageNet [46]	1,281,167	50,000	127/1000
iNaturalist 2018 [30]	437,513	24,426	6/.../8,142
iNaturalist 2019 [31]	265,240	3,003	6/.../1,010
Flowers-102 [41]	2,040	6,149	102
Stanford Cars [35]	8,144	8,041	196
Food101 [7]	75,750	25,250	101

CIFAR-100 [36] has 100 classes grouped into 20 coarse concepts of 5 fine classes each. For instance the coarse class *large carnivore* includes fine classes *bear*, *leopard*, *lion*, *tiger* and *wolf*. In all experiments, we use the coarse concepts to train our models and evaluate the trained model using the fine-grained labels.

ImageNet [46] classes follow the WordNet [20] hierarchy. We use the 127 coarse labels defined in Huh et al. [33] in order to allow for a direct comparison with their method.

iNaturalist-2018 offers 7 granularity levels from the most general to the most specific, that follow the biological taxonomy: Kingdom (6 classes), Phylum (25 classes), Class (57 classes), Order (272 classes), Family (1,118 classes), Genus (4,401 classes) and Species (8,142 classes). We consider pairs of (coarse,fine) granularity levels in our experiments. **iNaturalist-2019** is similar to iNaturalist-2018 with fewer classes and images, and yields similar conclusions.

Flowers-102, Stanford Cars and Food101 are fine-grained benchmarks with no provided coarse labelling. Therefore we can use them for the transfer learning task.

Evaluation metrics. For category-level retrieval we report the mean average precision (mAP), as commonly done for retrieval tasks [2, 42]. For on-the-fly classification we report the top-1 accuracy.

4.2. Baselines

We use existing baselines and introduce stronger ones:

Wu's baselines [61] use activations of a network learned with cross-entropy loss, but evaluated with a kNN classifier. Huh et al. [33] evaluate how a network trained on the 127 ImageNet coarse classes transfers on the 1000 fine labels[1].

[1]They fine-tune a linear classifier with fine labels. We do not consider this task in the body of the paper, but refer to Appendix B.2: our approach provides a significant improvement in this case as well.

Table 2: **Coarse-to-fine: comparison with the state of the art** for category-level retrieval (mAP, %) and kNN classification (top-1, %), with the ResNet50 architecture. We compare Grafit with the state of the art [61] and our stronger baselines. We highlight methods that use more parameters (32.9M vs $\sim$23.5M), see Table 5 for details.

Method	CIFAR-100		ImageNet-1k	
	kNN	mAP	kNN	mAP
Baseline, Wu et al. [61]	54.2	-	48.1	-
SNCA, Wu et al. [61]	62.3	-	52.8	-
Baseline (ours)	71.8	42.5	54.7	22.7
ClusterFit+	72.5	23.0	59.5	12.7
SNCA+	72.2	35.9	55.4	31.8
Grafit FC	75.6	55.0	**69.1**	**44.4**
Grafit	**77.7**	**55.7**	**69.1**	42.9

Our main baseline: we learn a network with cross-entropy loss, and perform retrieval or kNN-classification with the ℓ_2-normalized embedding produced by the model trunk. We point out that, thanks to our strong optimization strategy borrowed from recent works [28, 50], this baseline by itself outperforms all published results in several settings, for instance our ResNet-50 baseline without extra training data outperforms on ImageNet a ResNet-50 pre-trained on YFCC100M [66] (see Table 12 in Appendix B for a comparison).

SNCA. Wu et al. [61] proposed SNCA, a model optimized with a kNN loss. In our implementation, we add a linear operator P_θ to the network trunk f_θ when training the supervised loss $\mathcal{L}_{\mathrm{knn}}$.

SNCA+. We improve SNCA with our stronger optimization procedure. The retrieval or kNN evaluation uses features from a MLP instead of a simple linear projector, which means that its number of parameters is on par with Grafit (and larger than Grafit FC).

ClusterFit+. Same as for SNCA, we improve ClusterFit [67] with our training procedure, and cross-validate the number of clusters (15000 for Imagenet and 1500 for CIFAR-100). As a result we improve its performance and have a fair comparison, everything being equal otherwise.

4.3. Experimental details

Architectures. Most experiments are carried out using the ResNet-50 architecture [27] except for Section 4.6 where we also use RegNet [43] and ResNeXt [64].

Training settings. Our training procedure borrows from the bag of tricks [28]: we use SGD with Nesterov momentum and cosine learning rates decay. We follow Goyal et

Table 3: kNN evaluation on iNaturalist-2018 with different semantic levels. The symbol ∅ refers to the unsupervised case (a unique class). We compare with the best competing method according to Table 2.

	Train → ↓Test / #classes →	∅ 1	Kingdom 6	Phylum 25	class 57	Order 272	Family 1,118	Genus 4,401	Species 8,142
ClusterFit+	Kingdom	70.9	94.7	95.0	95.3	95.6	96.2	96.3	96.1
	Phylum	48.8	87.4	90.3	90.7	91.1	92.6	92.6	92.2
	Class	40.4	80.2	83.8	85.7	86.7	88.8	88.8	88.2
	Order	17.1	54.5	59.0	61.4	70.8	73.9	74.3	72.3
	Family	5.6	38.3	42.1	44.4	54.3	63.0	64.2	61.9
	Genus	0.9	26.7	29.5	31.5	40.1	49.4	53.9	51.7
	Species	0.3	21.8	23.7	25.2	32.7	40.3	44.7	43.4
Grafit	Kingdom	95.5	98.1	98.2	98.2	98.2	98.2	98.4	98.3
	Phylum	90.0	94.1	96.6	96.7	96.8	96.7	96.9	96.7
	Class	82.2	87.5	90.9	94.5	94.9	94.9	95.0	95.0
	Order	54.0	61.7	66.9	72.7	87.1	87.5	87.6	87.3
	Family	33.7	42.1	48.7	55.1	70.9	81.8	82.4	82.1
	Genus	20.5	27.0	33.5	39.5	54.2	64.6	75.6	75.5
	Species	15.9	20.4	25.5	30.8	42.7	51.2	61.9	67.7

Table 4: Category-retrieval evaluation (mAP, %) on iNaturalist-2018 with different semantics levels. We compare with the best baseline according to Table 2.

	Train → ↓Test / #classes →	Kingdom 6	Phylum 25	class 57	Order 272	Family 1,118	Genus 4,401	Species 8,142
SNCA+	Kingdom	97.6	83.3	75.9	59.2	56.0	54.9	55.0
	Phylum	59.8	91.7	79.4	49.1	35.0	32.3	32.2
	Class	41.3	73.1	89.9	49.2	28.1	23.6	23.0
	Order	9.09	24.9	35.7	77.9	35.3	18.0	15.0
	Family	2.24	6.43	11.2	35.7	68.4	29.1	21.7
	Genus	0.39	2.47	5.03	18.1	36.6	60.5	46.0
	Species	0.19	1.86	3.80	12.8	26.4	46.0	54.9
Grafit	Kingdom	98.6	88.3	79.7	60.8	58.0	55.9	55.5
	Phylum	67.8	97.2	82.1	50.9	38.9	34.2	33.0
	Class	50.1	74.9	95.4	51.2	32.3	25.9	24.1
	Order	17.7	30.7	42.7	88.3	42.3	21.1	16.2
	Family	8.70	13.2	18.0	43.9	83.1	34.8	24.2
	Genus	6.78	9.72	13.5	29.0	46.9	77.2	53.9
	Species	6.45	9.02	12.1	23.6	35.6	55.4	70.0

al.'s [24] recommendation for the learning rate magnitude: $\mathrm{lr} = \frac{0.1}{256} \times \mathrm{batchsize}$. The data augmentation consists of random resized crop, RandAugment [14] and Erasing [70]. We train for 600 epochs with batches of 1024 images at resolution 224×224 pixels (except for CIFAR-100: 32×32). We set the temperature σ to 0.05 in all our experiments following Wu et al. [61]. Appendix B.1 gives more details.

For the on-the-fly classification task, the unique hyper-parameter k is cross-validated in $k \in \{10, 15, 20, 25, 30\}$.

4.4. Coarse-to-fine experiments

CIFAR and ImageNet experiments. Table 2 compares Grafit results for coarse to fine tasks with the baselines from Section 4.2. On CIFAR-100, Grafit outperforms other methods by **+5.5%** top-1 accuracy. On ImageNet the gain over other methods is **+13.7%**.

Grafit also outperforms other methods on category-level retrieval, by **13.2%** on CIFAR and **11.1%** on ImageNet. Table 2 shows that Grafit not only provides a better on-the-fly classification (as evaluated by the kNN metric), but that the ranked list is more relevant to the query (results for mAP).

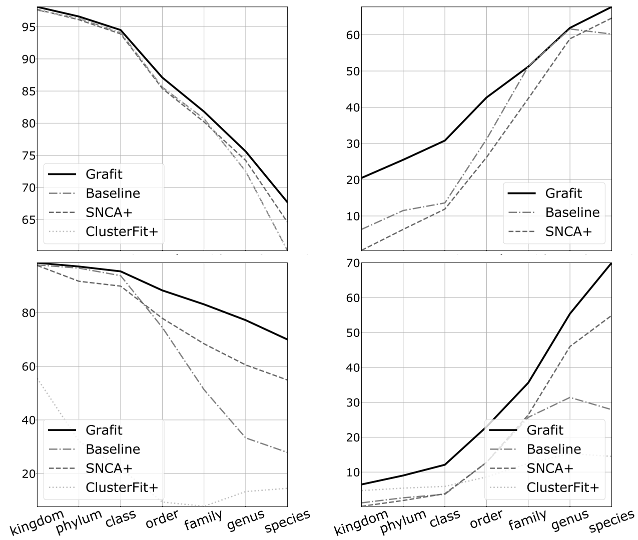


Figure 3: Evaluation on iNaturalist-2018 [30] with and *left:* train=test granularity *right:* test at finest granularity (species). We compare our method Grafit, SNCA+, ClusterFit+ and Baseline. *Top:* on-the-fly kNN classification (top-1 accuracy); *bottom:* category-level retrieval (mAP).

Coarse-to-Fine with different taxonomic ranks. We showcase Grafit on various levels of coarse granularity by training one model on each annotation level of iNaturalist-2018 and evaluating on all levels with kNN classification (Table 3) and retrieval (Table 4).

Figure 3 presents results with retrieval and kNN classification for two of the most interesting cases: when the train and test granularities are the same (left), and on the finest test level (Species) with varying granularities at training time (right). On the left, the accuracy of all methods decreases as the granularity increases: this is expected as the task moves from coarse classification to fine, as it is more difficult to discriminate amongst a larger number of classes.

We observe that the performance drop of Grafit for category-level retrieval is reduced in comparison with other methods. On the right figures, the accuracy of all methods increases as the level of annotation increases (keeping evaluation at Species). Grafit also stands out in this context, outperforming other methods.

We report comprehensive results with Grafit and the baselines from Section 4.2 on iNaturalist-2019 & 2018 in the supplemental material (Appendix B.3).

Visualizations. Figure 1 shows visual results for the category-level retrieval task with Grafit. All the results for the baseline and Grafit have the correct coarse label, but our method is better at a finer granularity. In Appendix C we show that the improvement is even more evident when the granularity level at training time is coarser.

Figure 4 presents t-SNE visualizations [55] of the latent

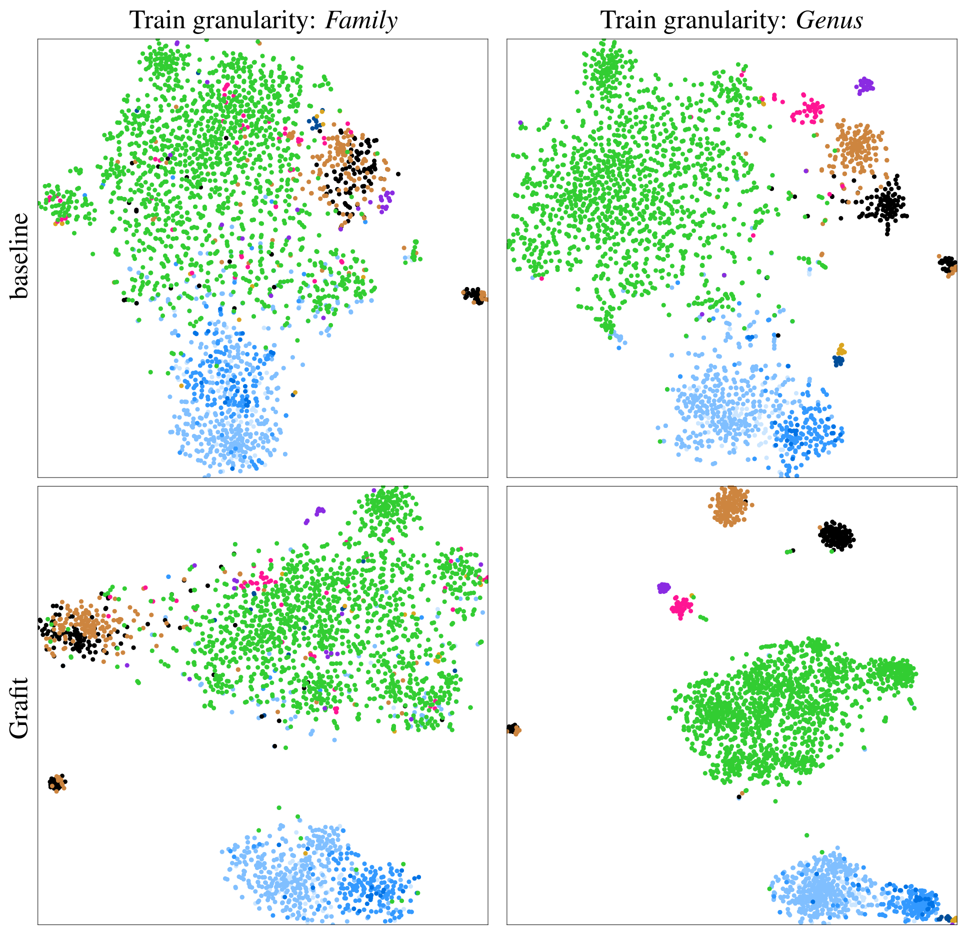
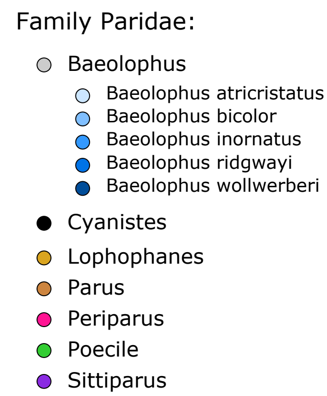


Figure 4: t-SNE representations of features from images of the family *paridae*, focusing on the genus *baeolophus* (in blue). When trained with granularity Family, all depicted points have the same coarse label, while granularity Genus means that the network has seen 7 distinct labels. Visually, Grafit offers a better separation of the images than the baseline w.r.t. the two finest level 'Genus' and 'Species'.



spaces associated with the baseline and Grafit for images associated with a sub-hierarchy of iNaturalist-2018.

4.5. Ablation studies

Losses, architectural choice and conditioning. Table 5 presents a study on CIFAR-100 and ImageNet-1k, where we ablate several components of our method. A large improvement stems from the instance loss when it supplements the supervised kNN loss. It is key for discriminating at a finer grain. The category-level retrieval significantly benefits from our approach, rising from 22.7% to 44.4% in the best case. Coarse conditioning also has a consistent measurable impact on performance, yielding around 3 mAP points across the various settings.

Sanity check: training with coarse vs fine labels. Table 6 compares the performance gap of several methods when training with coarse labels vs fine labels. The performance improvement of Grafit over competing methods on Imagenet is quite sizable: with fine-tuning, Grafit with coarse labels is almost on par with the baseline on fine labels. For on-the-fly classification, Grafit with coarse labels reaches 69.1% performance on Imagenet, significantly decreasing the gap with fine-grained labels settings. The kNN classification performance is 79.3%. This concurs with our prior observations in Section 4.4 on iNaturalist-2018.

Overall, in this setting Grafit provides some slight yet systematic improvement over the baseline. With a ResNet-50 architecture at image resolution 224×224 pixels, Grafit reaches **79.6%** top-1 accuracy with a kNN classifier on ImageNet, which is competitive with classical cross-entropy results published for this architecture. See Appendix B for a comparison (Table 12) and more results on Imagenet.

4.6. Transfer Learning to fine-grained datasets

We now evaluate Grafit for transfer learning on fine-grained datasets (See) Table 2, with ImageNet pre-training.

Settings. We initialize the network trunk with ImageNet pre-trained weights and fine-tune model. For our method, the network trunk f_θ remains fixed and the projector P_θ is discarded. For all methods we fine-tune during 240 epochs with a cosine learning rate schedule starting at 0.01 and batches of 512 images (details in Appendix B.4).

Table 5: Ablation study on CIFAR-100 and ImageNet with ResNet50 architecture. We report results both for on-the-fly classification (kNN classifier, top-1 accuracy, %) and category-level retrieval (mAP, %). The rows corresponding to the main baselines and methods discussed through our paper are highlighted: our baseline and improved SNCA+ in grey and red, and our two variants Grafit-FC and Grafit in blue. The last row is the result that Grafit would obtain with a perfect coarse classification.

Loss $\mathcal{L}_{\text{CE}}$	Loss $\mathcal{L}_{\text{knn}}$	Loss $\mathcal{L}_{\text{inst}}$	knn head proj. P_θ	coarse cond.	CIFAR100 kNN	CIFAR100 mAP	Imagenet kNN	Imagenet mAP	Imagenet #Params
✓	–	–	–	–	71.8	42.5	54.7	22.7	23.5M
✓	–	–	–	✓	71.8	43.1	54.7	24.4	23.5M
–	–	✓	–	–	54.3	14.3	41.7	3.47	23.5M
✓	–	✓	–	–	76.9	51.0	65.0	26.0	23.5M
–	✓	–	FC	–	70.0	39.7	57.8	30.7	23.8M
–	✓	✓	FC	–	75.6	53.6	**69.1**	41.7	23.8M
–	✓	✓	FC	✓	75.6	55.0	**69.1**	**44.4**	23.8M
–	✓	–	MLP	–	72.2	35.9	55.4	31.8	32.9M
–	✓	–	MLP	✓	72.2	41.4	55.4	32.9	32.9M
–	✓	✓	MLP	–	**77.7**	52.9	**69.1**	39.4	32.9M
–	✓	✓	MLP	✓	**77.7**	**55.7**	**69.1**	42.9	32.9M
–	✓	✓	MLP	oracle	77.7	*59.3*	69.1	*47.2*	32.9M

Table 6: We compare coarse-to-fine and fine-to-fine context with mAP (%), kNN (top-1, %) and fine-tuning (FT) with fine labels (top-1, %) on ImageNet.

Method (with ResNet50)	Train Coarse mAP	Train Coarse kNN	Train Coarse FT	Train Fine mAP	Train Fine kNN	Train Fine FT
Baseline	22.7	54.7	78.1	51.5	78.0	**79.3**
SNCA+	31.8	55.4	77.9	72.0	79.1	77.4
Grafit FC	**44.4**	**69.1**	**78.3**	**72.4**	79.2	78.5
Grafit	42.9	**69.1**	77.9	71.2	**79.6**	78.0

Classifier. We experiment with two types of classifiers: a standard linear classifier (FC) and a multi-layer perceptron (MLP) composed of two linear layers separated by a batch-normalization and a ReLU activation. We introduce this MLP because, during training, both Grafit and SNCA+ employ an MLP projector, so their feature space is not learned to be linearly separable. In contrast, the baseline is trained with a cross-entropy loss associated with a linear classifier.

Tasks. We evaluate on five classical transfer learning datasets: Oxford Flowers-102 [41], Stanford Cars [35], Food101 [7], iNaturalist 2018 [30] & 2019 [31]. Table 1 summarizes some statistics associated with each dataset.

Results. Table 7 compares a ResNet-50 pretrained on ImageNet with Grafit, SNCA+, ClusterFit [67] and our baseline on five transfer learning benchmarks. Our method outperforms all methods. The table also shows the relatively strong performance of SNCA+.

Table 8 compares Grafit with the RegNetY-8.0GF [43] architecture against the state of the art, on the same benchmarks. Note that this architecture is significantly faster than the EfficientNet-B7 and ResNet-152 employed in other papers, and that we use a lower resolution in most settings.

Table 7: Comparison of transfer learning performance for different pre-training methods. All methods use a ResNet-50 pre-trained on Imagenet. The training procedues are the same (except the result reported for ClusterFit [67]). We report the top-1 accuracy (%) with a single center crop evaluation at resolution 224×224. See Table 15 of Appendix B.4 for additional results with other architectures.

Dataset	Baseline	ClusterFit [67]	ClusterFit+	SNCA+	Grafit	Grafit FC
Flowers-102	96.2	–	96.2	**98.2**	**98.2**	97.6
Stanford Cars	90.0	–	89.4	92.5	92.5	**92.7**
Food101	88.9	–	88.9	88.8	**89.5**	88.7
iNaturalist 2018	68.4	49.7	67.5	69.2	**69.8**	68.5
iNaturalist 2019	73.7	–	73.8	74.5	**75.9**	74.6

Table 8: State of the art for transfer learning with pretrained ImageNet-1k models. We report top-1 accuracy (%) with a single center crop. For Grafit we use a 39M-parameter RegNetY-8.0GF [43] with resolution 384×384 pixels that is $4\times$ faster than EfficientNetB7 at inference. "Res" is the inference resolution in pixels.

Dataset	Best reported results (%) State of the art	# Params	Res	Top-1	Grafit Top-1
Flowers-102	EfficientNet-B7 [50]	64M	600	98.8	**99.1**
Stanford Cars	EfficientNet-B7 [50]	64M	600	**94.7**	**94.7**
Food101	EfficientNet-B7 [50]	64M	600	93.0	**93.7**
iNaturalist 2018	ResNet-152 [12]	60M	224	69.1	**81.2**
iNaturalist 2019	–	–	–	–	**84.1**

In Table 8 we consider models pre-trained on ImageNet with and fine-tuned on the fine-grained target dataset. In each case we report results with Grafit (with a MLP for the projector P_θ) and Grafit FC. See more detailed results in Appendix B Table 16.

In summary, Grafit establishes the new state of the art. We point out that we have used a consistent training scheme across all datasets, and a single architecture that is more efficient than in competing methods.

5. Conclusion

This paper has introduced a procedure to learn a neural network that offers a finer granularity than the one provided by the annotation. It improves the performance for fine-grained category retrieval within a coarsely annotated collection. For on-the-fly kNN classification, Grafit significantly reduces the gap with a network trained with fine labels. It also translates into better transfer learning to fine-grained datasets, outperforming the current state of the art with a more efficient network.

References

[1] Authors:copyright for Figure 1 images from inaturalist-2018, top to down, left to right, employed for illustration of research work. Diana-Terry Hibbitts: CC BY-NC 4.0, Ronald Werson:CC BY-NC-ND 4.0, Greg Lasley: CC BY-NC 4.0, Robin Agarwal: CC BY-NC 4.0, Stefano Tito: CC BY-NC 4.0, Marion Zöller: CC BY-NC 4.0, Stefano Tito: CC BY-NC 4.0, Ronald Werson: CC BY-NC-ND 4.0, Ronald Werson: CC BY-NC-ND 4.0, Donna Pomeroy: CC BY-NC 4.0, Chris van Swaay: CC BY-NC 4.0, Donna Pomeroy: CC BY-NC 4.0, Giuseppe Cagnetta: CC BY-NC 4.0, Chris van Swaay: CC BY-NC 4.0, martinswarren: CC BY-NC 4.0, Donna Pomeroy: CC BY-NC 4.0, Robin Agarwal: CC BY-NC 4.0, Fernando de Juana: CC BY-NC 4.0, Giuseppe Cagnetta: CC BY-NC 4.0, Ronald Werson: CC BY-NC-ND 4.0, Marion Zöller: CC BY-NC 4.0, martinswarren: CC BY-NC 4.0, Ronald Werson: CC BY-NC-ND 4.0, Donna Pomeroy: CC BY-NC 4.0, Donna Pomeroy: CC BY-NC 4.0, Marion Zöller: CC BY-NC 4.0, martinswarren: CC BY-NC 4.0, note = Accessed: 2020-06-10.

[2] Artem Babenko, Anton Slesarev, Alexander Chigorin, and Victor S. Lempitsky. Neural codes for image retrieval. *arXiv preprint arXiv:1404.1777*, 2014.

[3] Maxim Berman, Hervé Jégou, Andrea Vedaldi, Iasonas Kokkinos, and Matthijs Douze. Multigrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019.

[4] David Berthelot, N. Carlini, E. D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019.

[5] David Berthelot, Nicholas Carlini, I. Goodfellow, Nicolas Papernot, A. Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019.

[6] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. *arXiv preprint arXiv:1704.05310*, 2017.

[7] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

[8] Stéphane Boucheron, Olivier Bousquet, and Gabor Lugosi. Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics*, 2005.

[9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[11] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. *Conference on Computer Vision and Pattern Recognition*, 2015.

[12] P. Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. *arXiv preprint arXiv:2008.03673*, 2020.

[13] Tan Kiat Chuan, Liu Yulong, Ambrose Barbara, Tulig Melissa, and Belongie Serge. The herbarium challenge 2019 dataset. *arXiv preprint arXiv:1906.05372*, 2019.

[14] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. *arXiv preprint arXiv:1909.13719*, 2019.

[15] Yin Cui, Zeqi Gu, Dhruv Kumar Mahajan, Laurens van der Maaten, Serge J. Belongie, and Ser-Nam Lim. Measuring dataset granularity. *arXiv preprint arXiv:1912.10154*, 2019.

[16] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge J. Belongie. Large scale fine-grained categorization and domain-specific transfer learning. *Conference on Computer Vision and Pattern Recognition*, 2018.

[17] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. Underspecification presents challenges for credibility in modern machine learning, 2020.

[18] Jia Deng, J. Krause, A. Berg, and Li Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. *Conference on Computer Vision and Pattern Recognition*, 2012.

[19] Amir Erfan Eshratifar, David Eigen, Michael J. Gormish, and Massoud Pedram. Coarse2fine: A two-stage training method for fine-grained visual classification. *arXiv preprint arXiv:1909.02680*, 2019.

[20] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[21] Carolina Galleguillos, Brian McFee, Serge J. Belongie, and Gert R. G. Lanckriet. From region similarity to category discovery. *Conference on Computer Vision and Pattern Recognition*, 2011.

[22] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Learning representations by predicting bags of visual words. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[23] J. Goldberger, S. Roweis, Geoffrey E. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*, 2004.

[24] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[25] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[26] Yanming Guo, Yu Liu, Erwin M. Bakker, Yuanhao Guo, and Michael S. Lew. Cnn-rnn: a large-scale hierarchical image classification framework. *Multimedia Tools and Applications*, 2017.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, June 2016.

[28] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. *arXiv preprint arXiv:1812.01187*, 2018.

[29] Grant Van Horn, Oisin Mac Aodha, Yang Song, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist challenge 2017 dataset. *arXiv preprint arXiv:1707.06642*, 2017.

[30] Grant Van Horn, Oisin Mac Aodha, Yang Song, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist challenge 2018 dataset. *arXiv preprint arXiv:1707.06642*, 2018.

[31] Grant Van Horn, Oisin Mac Aodha, Yang Song, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist challenge 2019 dataset. *arXiv preprint arXiv:1707.06642*, 2019.

[32] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Deep image category discovery using a transferred similarity function. *arXiv preprint arXiv:1612.01253*, 2016.

[33] Mi-Young Huh, Pulkit Agrawal, and Alexei A. Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

[34] Xu Ji, Andrea Vedaldi, and João F. Henriques. Invariant information clustering for unsupervised image classification and segmentation. *International Conference on Computer Vision*, 2019.

[35] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, 2013.

[36] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, CIFAR, 2009.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[38] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper R. R. Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset v4. *International Journal of Computer Vision*, 2020.

[39] Martin Renqiang Min, L. V. D. Maaten, Zineng Yuan, A. Bonner, and Z. Zhang. Deep supervised t-distributed embedding. In *International Conference on Machine Learning*, 2010.

[40] Yair Movshovitz-Attias, A. Toshev, T. Leung, S. Ioffe, and S. Singh. No fuss distance metric learning using proxies. *International Conference on Computer Vision*, 2017.

[41] M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

[42] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Conference on Computer Vision and Pattern Recognition*, 2007.

[43] Ilija Radosavovic, Raj Prateek Kosaraju, Ross B. Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. *Conference on Computer Vision and Pattern Recognition*, 2020.

[44] Oren Rippel, Manohar Paluri, Piotr Dollár, and Lubomir D. Bourdev. Metric learning with adaptive density discrimination. *International Conference on Learning Representations*, 2016.

[45] Marko Ristin, Juergen Gall, Matthieu Guillaumin, and Luc Van Gool. From categories to subcategories: Large-scale image classification with partial class label refinement. *Conference on Computer Vision and Pattern Recognition*, 2015.

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International journal of Computer Vision*, 2015.

[47] R. Salakhutdinov and Geoffrey E. Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *AISTATS*, 2007.

[48] Kihyuk Sohn, David Berthelot, C. Li, Zizhao Zhang, N. Carlini, E. D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[49] Fariborz Taherkhani, Hadi Kazemi, Ali Dabouei, Jeremy Dawson, and Nasser M. Nasrabadi. A weakly supervised fine label classifier enhanced by coarse supervision. In *International Conference on Computer Vision*, 2019.

[50] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[51] Eu Wern Teh, Terrance Devries, and Graham W. Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. *European Conference on Computer Vision*, 2020.

[52] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Lijia Li. Yfcc100m: the new data in multimedia research. *Commun. ACM*, 2016.

[53] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Herve Jegou. Fixing the train-test resolution discrepancy. *Advances in Neural Information Processing Systems*, 2019.

[54] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy: Fixefficientnet. *arXiv preprint arXiv:2003.08237*, 2020.

[55] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008.

[56] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, 2020.

[57] Huy V. Vo, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Pérez, and Jean Ponce. Unsupervised image matching and object discovery as optimization. *Conference on Computer Vision and Pattern Recognition*, pages 8279–8288, 2019.

[58] Huy V. Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. *arXiv preprint arXiv:2007.02662*, 2020.

[59] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011.

[60] Cinna Wu, M. Tygert, and Y. LeCun. A hierarchical loss and its problems when classifying non-hierarchically. *PLoS ONE*, 2019.

[61] Zhirong Wu, Alexei A Efros, and Stella Yu. Improving generalization via scalable neighborhood component analysis. In *European Conference on Computer Vision*, 2018.

[62] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

[63] Qizhe Xie, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Self-training with noisy student improves imagenet classification. *arXiv preprint arXiv:1911.04252*, 2019.

[64] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *Conference on Computer Vision and Pattern Recognition*, 2017.

[65] Saining Xie, Tianbao Yang, X. Wang, and Y. Lin. Hyper-class augmented and regularized deep learning for fine-grained image classification. *Conference on Computer Vision and Pattern Recognition*, 2015.

[66] Ismet Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Kumar Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

[67] Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Kumar Mahajan. Clusterfit: Improving generalization of visual representations. *Conference on Computer Vision and Pattern Recognition*, 2020.

[68] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *arXiv preprint arXiv:1905.04899*, 2019.

[69] Xiaohua Zhai, A. Oliver, A. Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. *International Conference on Computer Vision*, 2019.

[70] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020.