This CVPR 2020 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Online Deep Clustering for Unsupervised Representation Learning

Xiaohang Zhan*[1], Jiahao Xie*[2], Ziwei Liu[1], Yew Soon Ong[2,3], Chen Change Loy[2]

[1]CUHK - SenseTime Joint Lab, The Chinese University of Hong Kong
[2]Nanyang Technological University [3]AI3, A*STAR, Singapore

[1]{zx017, zwliu}@ie.cuhk.edu.hk
[2]{jiahao003, asysong, ccloy}@ntu.edu.sg

Abstract

Joint clustering and feature learning methods have shown remarkable performance in unsupervised representation learning. However, the training schedule alternating between feature clustering and network parameters update leads to unstable learning of visual representations. To overcome this challenge, we propose Online Deep Clustering (ODC) that performs clustering and network update simultaneously rather than alternatingly. Our key insight is that the cluster centroids should evolve steadily in keeping the classifier stably updated. Specifically, we design and maintain two dynamic memory modules, i.e.*, samples memory to store samples' labels and features, and centroids memory for centroids evolution. We break down the abrupt global clustering into steady memory update and batch-wise label re-assignment. The process is integrated into network update iterations. In this way, labels and the network evolve shoulder-to-shoulder rather than alternatingly. Extensive experiments demonstrate that ODC stabilizes the training process and boosts the performance effectively.*

1. Introduction

Unsupervised representation learning [1, 2, 3, 4, 5, 6, 7, 8, 9] aims at learning transferable image or video representations without manual annotations. Among them, clustering-based representation learning methods [10, 11, 12, 13, 14] emerge as a promising direction in this area. Different from recovering-based approaches [2, 3, 4, 8], clustering-based methods require little domain knowledge [13] while achieving encouraging performances. Compared to contrastive representation learning [15, 16, 17] that captures merely intra-image invariance, clustering-based methods are able to explore inter-image similarity. Unlike conventional clustering that is typically performed on fixed features [18, 19], these works jointly optimize clustering and feature learning.

While evaluations of early works [11, 12] are mostly

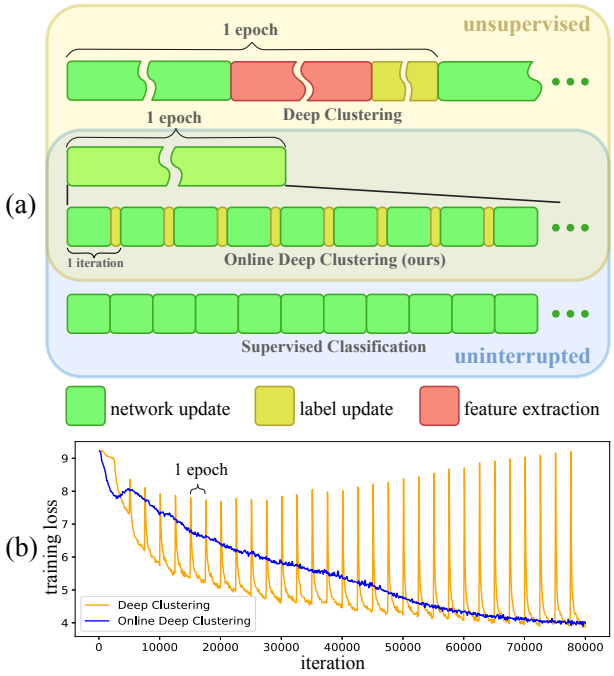

Figure 1. (a) Online Deep Clustering (ODC) seeks to reduce the discrepancy in training mechanism between Deep Clustering (DC) and supervised classification via integrating clustering process into network update iterations. ODC training is both unsupervised and uninterrupted. (b) Compared to DC, ODC updates labels continuously rather than in a pulsating manner, enabling the representations to evolve steadily. The loss curves (only initial 32 epochs for clarity) show the stability of ODC. After training, the loss is decreased to around 2.0 for ODC while 2.9 for DC.

*Equal Contribution.

performed on small datasets, Deep Clustering [13] (DC) proposed by Caron *et al*. is the first attempt to scale up clustering-based representation learning. DC alternates between deep feature clustering and CNN parameters update. In particular, at the start of each epoch, it performs offline clustering algorithms on the entire dataset to obtain pseudo-labels as the supervision for the next epoch. Offline clustering inevitably permutes the assigned labels in different epochs, *i.e*., even if some of the clusters do not change, their indices after clustering will be permuted randomly. As a result, parameters in the classifier cannot be inherited from the last epoch and they have to be randomly initialized before each epoch. The mechanism introduces training instability and exposes representations to a high risk of representation corruption. As shown in Figure 1 (a), network update in DC is interrupted by feature extraction and clustering in each epoch. This is in contrast to the conventional supervised classification that is performed in an uninterrupted manner using fixed labels, where an iteration consists of forward and backward propagations of the network.

In this work, we seek to devise a joint clustering and feature learning paradigm with high stability. To reduce the discrepancy of training mechanism between DC and supervised learning, we decompose the clustering process into mini-batch-wise label update, and integrate this update process into iterations of network update. Based on this intuition, we propose Online Deep Clustering (ODC) for joint clustering and feature learning. Specifically, an ODC iteration consists of forward and backward propagations, label re-assignment, and centroids update. For label update, ODC reuses the features in the forward propagation, thus avoiding additional feature extraction. To facilitate online label re-assignment and centroids update, we design and maintain two dynamic memory modules, *i.e*., samples memory to store samples' labels and features, and centroids memory for centroids evolution. In this way, ODC is trained in an uninterrupted manner similar to supervised classification, while no manual annotation is required. During the training process, labels and network parameters evolve shoulder-to-shoulder, rather than alternatingly. Since labels are updated in each iteration continuously and instantly, the classifier in the CNN also evolves more steadily, resulting in a much more steady loss curve as shown in Figure 1 (b).

While ODC alone achieves compelling unsupervised representation learning performance on various benchmarks, it can be naturally used to fine-tune models that have been trained using other unsupervised learning approaches. Extensive experiments show that the steadiness of ODC helps it to perform superiorly over DC as an unsupervised fine-tuning tool. We conclude our contributions as follows: **1)** we propose ODC that learns image representations in an unsupervised manner with high stability. **2)** ODC also serves as a unified unsupervised fine-tuning scheme that further improves previous self-supervised representation learning approaches. **3)** Promising performances are observed on different benchmarks, indicating the great potential of joint clustering and feature learning.

2. Related Work

Unsupervised Representation Learning. Many unsupervised visual representation learning algorithms are based on generative models, which usually use a latent representation bottleneck to reconstruct input images. Existing generation-based models include Auto-Encoders [20, 21], Restricted Boltzman Machines [22, 23, 24], Variational Auto-Encoders [25] and Generative Adversarial Networks [26], some of which have shown powerful ability in generating images or videos [27, 28, 29, 30, 31, 32]. By learning to generate examples, these models can learn meaningful latent representations that can be used for downstream tasks [5, 33, 34].

Another popular form of unsupervised representation learning is self-supervised learning, where a pretext task is designed to derive proxy labels from raw data. Representations are learned by encouraging a CNN to predict the proxy labels from the data. Various pretext tasks have been explored, *e.g*., predicting relative patch locations within an image [1], solving jigsaw puzzles [4], colorizing grayscale images [3, 35], inpainting of missing pixels [2], cross-channel prediction [36], counting visual primitives [37], and predicting image rotations [8]. For videos, self-derived supervision signals come from temporal continuity [38, 39, 40, 41, 42, 43, 44] or motion consistency [45, 46, 47, 48, 9].

Joint Clustering and Feature Learning. Clustering-based unsupervised representation learning is of particular interest recently. Various methods are proposed to jointly optimize feature learning and image clustering. Notably, these methods have shown great potential in learning unsupervised features on small datasets [11, 12, 49, 50]. To scale up to large datasets like ImageNet [51], Caron *et al*. [13] propose DeepCluster to cluster features and update CNN with subsequent assigned pseudo-labels for each epoch. In a subsequent study, Caron *et al*. [14] propose DeeperCluster to leverage self-supervision and clustering, and validate the representation learning ability of their approaches on non-curated data. Although deep clustering methods are capable of learning good representations from large-scale unlabeled data, the alternating update of feature clustering and CNN parameters update leads to instability in training.

Improvements to Self-supervised Learning. Some works aim at improving previous self-supervised learning approaches from different perspectives. For instance, Larsson *et al*. [6] give a first in-depth analysis on colorization as a pretext task and provide some insights on improving its effectiveness. Mundhenk *et al*. [52] explore a set of methods

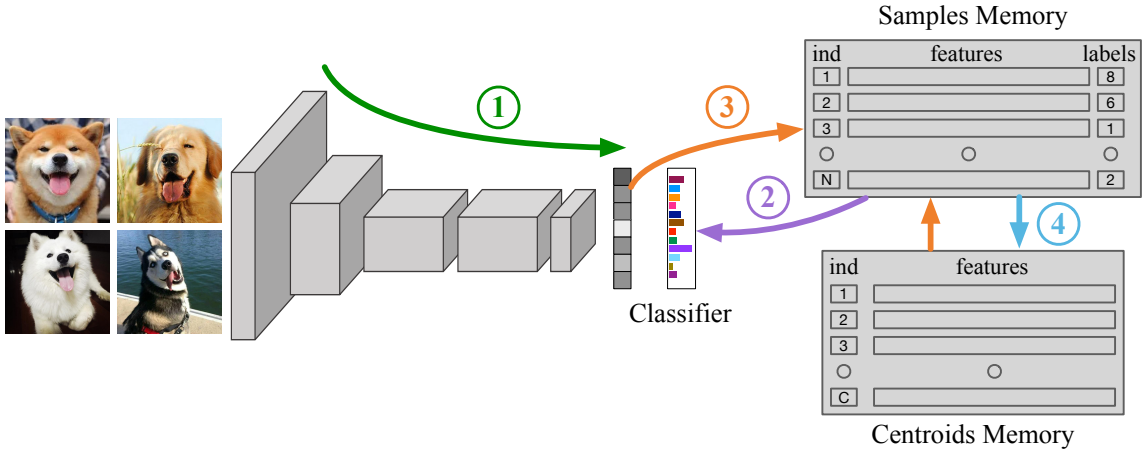

Figure 2. Each ODC iteration mainly contains four steps: 1. forward to obtain a compact feature vector; 2. read labels from the samples memory and perform back-propagation to update the CNN; 3. update samples memory by updating features and assigning new labels; 4. update centroids memory by recomputing the involved centroids.

to avoid some trivial shortcuts like chromatic aberration on context-based self-supervised learning. Noroozi *et al*. [53] improve the performance of self-supervised models using a clustering-based knowledge transfer method that allows a deeper network during pre-training. Wang *et al*. [54] and Doersch *et al*. [55] exploit multiple cues contained in different pretext tasks to improve self-supervised models. Recently, some works [56, 57] have studied extensively the architectures and scaling ability on existing self-supervised approaches. Complementary to these works, ODC serves as a flexible and unified unsupervised fine-tuning scheme to boost general self-supervised learning methods although it can be used alone to perform unsupervised representation learning from scratch.

3. Methodology

In the following sub-sections, we first discuss the differences between the proposed ODC to the conventional DC [13] in Sec. 3.1. We then recommend some useful strategies to maintain stable cluster size while using ODC in Sec. 3.2. We finally explain how one can use ODC for unsupervised fine-tuning (Sec. 3.3) and the implementation details of ODC (Sec. 3.4).

3.1. Online Deep Clustering

We first discuss the basic idea of DC [13] and then detail the proposed ODC. To learn representations, DC alternates between off-line feature clustering and network back-propagation with pseudo-labels. The off-line clustering process requires deep feature extraction on the entire training set, followed by a global clustering algorithm, *e.g*., K-Means clustering. The global clustering permutes the pseudo labels vastly, requiring the network to adapt to new labels rapidly in the subsequent epoch.

Framework Overview. Different from DC, ODC does not require an extra feature extraction process. Besides, labels evolve alongside the network parameters update smoothly. This is made possible by the newly introduced samples and centroids memories. As shown in Fig. 2, the samples memory stores features and pseudo-labels of the entire dataset; while the centroids memory stores the features of class centroids, *i.e*., the mean feature of all samples in a class. A "class" here represents a temporary cluster that evolves continuously during training. Labels and network parameters are updated simultaneously during uninterrupted iterations of ODC. Specific techniques including *loss re-weighting* and *dealing with small clusters* are introduced to avoid ODC from getting stuck into trivial solutions.

An ODC Iteration. Assuming that we are given with a randomly initialized network $f_\theta(*)$ along with a linear classifier $g_w(*)$, the goal is to train the backbone parameters θ to produce highly discriminative representations. To prepare for ODC, the samples and centroids memories are initialized via a global clustering process, *e.g*., K-Means. Next, one can perform uninterrupted ODC iteratively.

An ODC iteration contains four steps. First, given a batch of input images $\{x\}$, the network maps the images into compact feature vectors $F = f_\theta(x)$. Second, we read pseudo-labels for this batch from the samples memory. With the pseudo-labels, we update the network with stochastic gradient descent to solve the following problem:

$$\min_{\theta,w} \frac{1}{B} \sum_{n=1}^{B} l\left(g_w\left(f_\theta\left(x_n\right)\right), y_n\right), \tag{1}$$

where y_n is the current pseudo label from the samples memory, B denotes the size of each mini-batch. Third, $f_\theta(x)$ after L2 normalization is reused to update the samples memory:

$$F_m(x) \leftarrow m \frac{f_\theta(x)}{\|f_\theta(x)\|_2} + (1-m) F_m(x), \quad (2)$$

where $F_m(x)$ is the feature of x in the samples memory, $m \in (0, 1]$ is a momentum coefficient. Simultaneously, each involved sample is assigned with a new label by finding the nearest centroid following:

$$\min_{y \in \{1,..,C\}} \|F_m(x) - C_y\|_2^2, \quad (3)$$

where C_y denotes the centroid feature of class y. Finally, the involved centroids, including those in which new members join, and those from which old members leave, are recorded. They are updated every k-th iterations by averaging the features of all samples belonging to their corresponding centroid.

3.2. Handling Clustering Distribution in ODC

Loss Re-weighting. To avoid the training from collapsing into a few huge clusters, DC adopts uniform sampling before each epoch. However, for ODC, the number of samples over the clusters changes in each iteration. Using uniform sampling requires one to re-sample the entire dataset in each iteration, a process that is deemed redundant and costly. We propose an alternative approach, *i.e.*, re-weighting the loss according to the number of samples in each class. To verify their equivalence, we implement a DC model with loss re-weighting and empirically find that the performance remains unchanged when the weight follows $w_c \propto \frac{1}{\sqrt{N_c}}$, where N_c denotes the number of samples in class c. Hence, we adopt the same loss re-weighting formulation for ODC. With loss re-weighting, samples in smaller clusters contribute more towards backpropagation, thus pushing the decision boundary farther to accept more potential samples.

Dealing with Small Clusters. Loss re-weighting helps to prevent the formation of huge clusters. Nevertheless, we still face the risk of having some small clusters collapsing into empty clusters. To overcome this problem, we propose to process and eliminate extremely small clusters in advance before they collapse. Denoting normal clusters as $\mathcal{C}_n$ whose sizes are larger than a threshold, and small clusters as $\mathcal{C}_s$ whose sizes are not, for $c \in \mathcal{C}_s$, we first assign samples in c to the nearest centroids in C_n to make c empty. Next, we split the largest cluster $c_{max} \in \mathcal{C}_n$ into two sub-clusters by K-Means and randomly choose one of the sub-clusters as the new c. We repeat the process until all clusters belong to $\mathcal{C}_n$. Though this process alters some clusters abruptly, it only affects a small portion of samples which are involved in this process.

Dimensionality Reduction. Some of the backbone networks map an image to a high-dimensional vector, *e.g.*, AlexNet produces 4,096-dimensional features and ResNet-50 yields 2,048-dimensional features, leading to high space and time complexities in subsequent clustering. DC performed PCA on features across the entire dataset to reduce dimension. However, for ODC, the features of different samples have varying timestamps, leading to incompatible statistics among samples. Hence, PCA is not applicable anymore. It is also costly to perform PCA in each iteration. We therefore add a non-linear head layer of {fc-bn-relu-dropout-fc-relu} to reduce high dimensional features into 256 dimensions. It is jointly tuned during ODC iterations. The head layer is removed for downstream tasks.

3.3. ODC for Unsupervised Fine-tuning

Compared with self-supervised learning approaches that tend to capture intra-image semantics, clustering-based methods focus more on inter-image information. Hence, DC and ODC are naturally complementary to previous self-supervised learning approaches. As DC and ODC are not restricted to a specifically designed objective, like rotation angle or color prediction, they readily serve as an unsupervised fine-tuning scheme to boost the performance of existing self-supervised approaches. In this paper, we study the effectiveness of DC and ODC as a fine-tuning process with initialization from different self-supervised learning methods.

3.4. Implementation Details

Data Pre-processing. We use ImageNet that contains 1.28M images without labels for training. Images are first randomly cropped to have a resolution of 224x224 with augmentation including random flipping and rotation ($\pm 2°$). DC adopts a Sobel filter on the images to avoid exploiting color as the shortcut. Such a pre-processing step requires the downstream tasks to include the Sobel layer as well, which potentially limit its application. We find that strong color jittering shows the same effect as a Sobel filter in avoiding shortcuts, while it allows normal RGB images as inputs. Specifically, we adopt PyTorch style color jitter transform with brightness factor $(0.6, 1.4)$, contrast factor $(0.6, 1.4)$, saturation factor $(0, 2)$, and hue factor $(-0.5, 0.5)$. Besides, we randomly convert images to grayscale with a probability of 0.2. The random color jittering and grayscale applied on training samples randomize the similarity measured in color. This discourages the network from exploiting trivial information from color.

Training of ODC. We use ResNet-50 as our backbone. Considering that most early works use AlexNet, we also perform experiments on AlexNet for comparison. Follow-

ing [13], we use AlexNet architecture without Local Response Normalization and add batch normalization layers. The ODC models for AlexNet and ResNet-50 are trained from scratch. The batch size is 512 allocated to 8 GPUs. The learning rate is constantly 0.01 for AlexNet and 0.03 for ResNet-50 for 400 epochs, and decayed by 0.1 for further 80 epochs. Following DC, the number of clusters is set as 10,000, which is 10 times larger than the annotated number of classes of ImageNet. The momentum coefficient m is set as 0.5. The threshold to identify small clusters is set as 20. Varying this threshold does not affect the results significantly, provided that it does not exceed the average number of samples in a cluster. The centroids memory is updated in every 10 iterations. The centroids update frequency constitutes a trade-off between learning efficacy and efficiency. In our experiments, we observe that as long as the frequency is restricted to a reasonable range, the performance of ODC is not sensitive to it.

4. Experiments

4.1. Evaluation on Unsupervised Representation

After pre-training the ODC model, we evaluate the quality of unsupervised features on standard downstream tasks including ImageNet classification, Places205 [62] classification, VOC2007 [63] SVM classification, and VOC2007 Low-shot classification. We provide the details of each benchmark and show our competing results as follows.

Re-implementation of Deep Clustering. Since the original paper of DC does not include ResNet-50, we implement a DC model with ResNet-50. The DC model adopts the same data augmentations as ODC, except that DC applies a Sobel filter on images. For fair comparisons, the training hyper-parameters of DC are the same as ODC except that we empirically find $lr = 0.1$ is more suitable for DC.

ImageNet Classification. Following the setup in Zhang *et al*. [36], we keep the backbone including all convolution and batch normalization layers frozen, and train a 1000-way linear classifier on features from different depths of convolutional layers. The features are mapped to around 9000 dimensions via average pooling. We train all models for 100 epochs in total, using SGD with a momentum of 0.9 and batch size of 256. The learning rate is initialized as 0.01, decayed by a factor of 10 after every 30 epochs. Other hyper-parameters are set following Goyal *et al*. [57]. We report top-1 center-crop accuracy on the official validation split of ImageNet.

For AlexNet, as shown in Table 1, ODC has a consistent improvement over DC in all conv layers, with the largest improvement (6.7%) observed in conv1 layer. The performance in conv1 layer surpasses the ImageNet pre-trained model. With regard to the best-performing layer, ODC achieves 41.4% on conv4 layer, outperforming the latest LA [61], ranking only second to Rot-Decoupling [60]. Though ODC does not outperform Rot-Decoupling in its best performing layer, it provides a complementary perspective to rotation based methods.

ODC also scales well with deeper architectures. For ResNet-50, as shown in Table 2, ODC achieves 57.6% center-crop accuracy in the conv5 layer, which is 5.4% higher than the best performing layer of the re-implemented DC. Compared with the concurrent state-of-the-art method LA [61], our method produces competing results. Though the result of conv5 is slightly lower than LA, ODC outperforms LA from conv1 to conv4 layers by large margins. We observe a consistent performance increase from shallower layers to deeper layers, indicating that ODC makes full use of all residual layers.

Places205 Classification. Following Zhang *et al*. [36], to test the generalization ability on other domains, we also transfer the learned models to Places205 dataset that contains 2.45M images of 205 scene categories. Similar to the experiments on ImageNet, we train a 205-way linear classifier on top of each frozen convolutional layer on the train split of Places205, and report top-1 center-crop accuracy on the standard validation split. The evaluation setting and hyper-parameters are the same as those in the ImageNet classification task.

The results in Table 1 show that ODC with AlexNet as the backbone outperforms DC in all layers as well. ODC surpasses all previous works on conv1, conv3 and conv4 layers. Similar to the observation in the ImageNet classification task, ODC scales well on deeper architectures when it is transferred to Places205 with ResNet-50. As shown in Table 2, in all layers, ODC surpasses all previous works, with the largest margin (3.1%) to the runner-up observed in conv2 layer. With regard to the best performing layer, ODC reaches 49.3% center-crop accuracy in the conv5 layer, surpassing the re-implemented DC by 3.2% in the respective best layer. We observe the superiority of ODC in conv1 and conv2 layers over the supervised model using either Places labels or ImageNet labels. The transfer performance of our method in the Places205 classification task indicates that representations learned by ODC can generalize well to different domains from ImageNet.

VOC2007 SVM Classification. To further evaluate the generalization of learned features, we perform experiments on the VOC2007 transfer learning task that resembles real applications with smaller datasets. Following [57], we train linear SVMs on features extracted from the frozen backbone on the "trainval" split of VOC2007 and evaluate on the test split. We follow the same test setting and hyper-parameters used in [57], and report the best performing layers of different methods for ResNet-50. The results in Table 3 show that ODC surpasses previous approaches by a significant margin on the VOC2007 SVM classification

Table 1. AlexNet linear classification on ImageNet and Places. We report top-1 center-crop accuracy. Numbers for other methods are obtained either from [36] or from their original papers. The highest performance in each layer is in bold, and the second highest performance in each layer is underlined.

Method	ImageNet					Places				
(AlexNet)	conv1	conv2	conv3	conv4	conv5	conv1	conv2	conv3	conv4	conv5
Places labels [36]	-	-	-	-	-	22.1	35.1	40.2	43.3	44.6
ImageNet labels [36]	19.3	36.3	44.2	48.3	50.5	22.7	34.8	38.4	39.4	38.7
Random [36]	11.6	17.1	16.9	16.3	14.1	15.7	20.3	19.8	19.1	17.5
Context [1]	16.2	23.3	30.2	31.7	29.6	19.7	26.7	31.9	32.7	30.9
ContextEncoder [2]	14.1	20.7	21.0	19.8	15.5	18.2	23.2	23.4	21.9	18.4
Jigsaw [4]	19.2	30.1	34.7	33.9	28.3	23.0	32.1	35.5	34.8	31.3
Colorization [3]	13.1	24.8	31.0	32.6	31.8	22.0	28.7	31.8	31.3	29.7
SplitBrain [36]	17.7	29.3	35.4	35.2	32.8	21.3	30.7	34.0	34.1	32.5
Counting [37]	18.0	30.6	34.3	32.5	25.7	23.3	**33.9**	36.3	34.7	29.6
NPID [58]	16.8	26.5	31.8	34.1	35.6	18.8	24.3	31.9	34.5	33.6
Rotation [8]	18.8	31.7	38.7	38.2	36.5	21.5	31.0	35.1	34.6	33.7
DeepCluster [13]	12.9	29.2	38.2	39.8	36.1	18.6	30.8	37.0	37.5	33.1
AET [59]	19.2	32.8	40.6	39.7	37.7	22.1	32.9	37.1	36.2	34.7
Rot-Decouple [60]	19.3	**33.3**	**40.8**	**41.8**	**44.3**	22.9	32.4	36.6	37.3	**38.6**
LA [61]	14.9	30.1	35.7	39.4	40.2	17.1	32.2	36.5	38.3	37.8
ODC (Ours)	**19.6**	32.8	40.4	41.4	37.3	**24.0**	33.2	**38.3**	**38.4**	35.5

Table 2. ResNet-50 linear classification on ImageNet and Places. We report top-1 center-crop accuracy. Numbers for methods with * and † are produced by third-party studies as cited, and by us, respectively. Numbers for other methods are taken from their original papers. The highest performance in each layer is in bold, and the second highest performance in each layer is underlined.

Method	ImageNet					Places				
(ResNet-50)	conv1	conv2	conv3	conv4	conv5	conv1	conv2	conv3	conv4	conv5
Places labels [57]*	-	-	-	-	-	16.7	32.3	43.2	54.7	62.3
ImageNet labels [57]*	11.6	33.3	48.7	67.9	75.5	14.8	32.6	42.1	50.8	52.5
Random [57]*	9.6	13.7	12.0	8.0	5.6	12.9	16.6	15.5	11.6	9.0
Jigsaw [57]*	12.4	28.0	39.9	45.7	34.2	15.1	28.8	36.8	41.2	34.4
Colorization [57]*	10.2	24.1	31.4	39.6	35.2	14.7	27.4	32.7	37.5	34.8
NPID [58]	**15.3**	18.8	24.9	40.6	54.0	18.1	22.3	29.7	42.1	45.5
Rotation [56]*	41.7 (best layer)					38.1 (best layer)				
BigBiGAN [34]	55.4 (best layer)					-				
DeepCluster [13]†	14.4	29.6	39.9	52.2	50.3	19.3	31.9	39.0	46.1	43.6
LA [61]	9.3	23.2	38.0	48.6	**58.8**	18.3	31.5	39.2	46.3	49.1
ODC (Ours)	14.8	**31.6**	**42.5**	**55.7**	57.6	**21.4**	**35.0**	**41.3**	**47.4**	**49.3**

task. With ODC, we achieve 78.2% mAP performance, which is 9.1% higher than DC. However, We also note that there is still a significant 9.8% performance gap between our ODC and the supervised model pre-trained with ImageNet labels, leaving room for further exploration.

Low-shot VOC2007 SVM Classification. Following [57], we also transfer our learned representations to a low-shot setting of VOC2007 SVM classification to test the quality of features when there are few training examples per category. We vary the number of positive samples in each class and train linear SVMs on the frozen ResNet-50 backbone using the same setting from VOC2007 SVM classification. We use the standard "trainval" split of VOC2007 in training and the test split in testing. We report the mean average precision (mAP) across five independent samples for various low-shot values in Figure 3. The final mAP results shown in Table 3 are observed as the averages of all low-shot values and all independent runs. The per-shot results are shown in Figure 3. ODC has a consistent improvement over DC for each shot, with the performance gap further increasing when more positive examples per class are allowed. We also observe that the performance gap between ODC and

Table 3. ResNet-50 SVM classification and low-shot SVM classification mAP on VOC07. Numbers for methods with† are produced by us. Numbers for other methods are taken from [57].

Method (ResNet-50)	best layer	VOC07 SVM (% mAP)	VOC07 SVM Low-shot (% mAP)
ImageNet labels	5	88.0	75.4
Random	1	9.6	12.7
Jigsaw [4]	4	64.5	39.2
Colorization [3]	4	55.6	33.3
Rotation [8]†	4	67.4	41.0
DeepCluster [13]†	5	69.1	46.9
ODC (Ours)	5	**78.2**	**57.1**

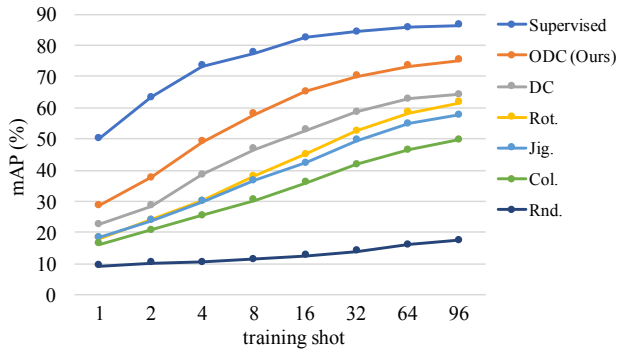

Figure 3. Low-shot Image Classification on VOC07 with linear SVMs trained and tested on the features from the best layer respectively for each method. We show the average performance for each shot across five runs.

the supervised model pre-trained with ImageNet labels is gradually narrowed down with the increase of training shot values. Table 3 shows that ODC achieves 57.1% mAP performance in low-shot SVM classification on VOC2007, 10.2% higher than our counterpart DC. The low-shot results of ODC in this benchmark suggest that the learned features through ODC generalize well to low-shot classification.

4.2. Further Analysis

In this section, we further analyze our ODC model from different perspectives.

ODC as a Fine-tuning Scheme. The high efficiency enables ODC to easily serve as a rapid unsupervised fine-tuning scheme. To assess the fine-tuning ability of ODC, we also use our reimplemented DC to fine-tune other self-supervised models. The improvements over different self-supervised approaches are shown in Table 4. Compared with DC, we observe that ODC boosts the performance of each self-supervised approach by a significant margin. With ODC fine-tuning, we achieve 16.7% improvements for Col., 9.9% for Jig., 7.1% for Rot., and 7.9% for DC, respectively, on the VOC2007 SVM classification benchmark. By contrast, DC also yields fine-tuning improvements but lags far behind ODC.

Table 4. Improvements over previous self-supervised approaches. Each model is fine-tuned for 120 epochs. We report VOC07 SVM classification mAP for ResNet-50. Pre-trained models marked* are provided by [57], hence the original results are also taken from [57]. For methods marked†, we reimplement them to obtain the results.

	Col. [3]*	Jig. [4]*	Rot. [8]†	DC [13]†
Original	55.6	64.5	67.4	69.1
DC [13]†	61.2	68.5	68.6	70.0
ODC	**72.3**	**74.4**	**74.5**	**77.0**

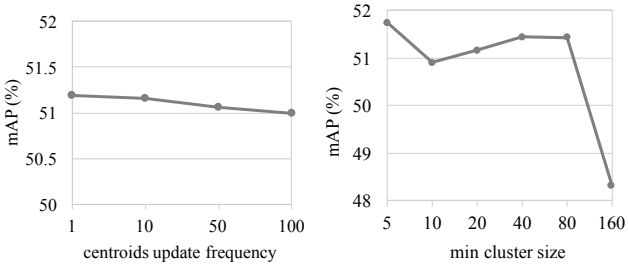

Figure 4. Influence of centroids update frequency (*left*) and minimal small cluster size (*right*) on the quality of features learned by ODC. We study these hyper-parameters on uniformly sampled 90K ImageNet within 300 random classes. We report mAP on VOC07 SVM classification task with ResNet-50.

Influence of the Hyper-parameters. The hyper-parameters of ODC include the frequency of updating the centroids memory, and the minimal size of clusters. To study the influence of the aforementioned two hyper-parameters, we train models with 90K images that are uniformly sampled from the original 1.28M ImageNet dataset, and evaluate the performance on VOC2007 SVM classification benchmark. Figure 4 shows the influence of the update frequency of centroids memory. We observe no significant decrease in the performance of ODC when the update frequency becomes lower, indicating that our method is insensitive to this hyper-parameter provided that it is within a reasonable range. The influence of the minimal size of small clusters is shown in Figure 4. The results show that a large threshold (*i.e.* 160) on clusters size would lead to a performance drop. The result is not surprising. A cluster whose size is smaller than the minimal size is identified as a "small cluster". An overly frequent processing of such small clusters (see Sec. 3.2) introduces instability in feature learning. The large threshold would also group images that should not have belonged to the same class. It is noteworthy that ODC does not experience a significant change in performance within a reasonable range of minimal cluster sizes.

Stability and Convergence. Figure 1 already demonstrates the superior stability of ODC over DC from the aspect of the loss curve. In Figure 5, we show the training stability and

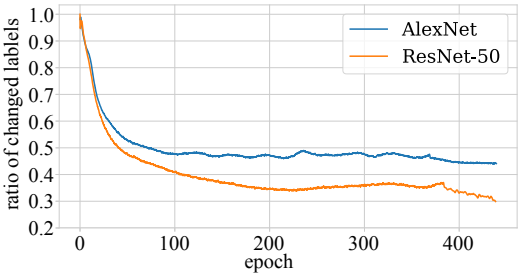

Figure 5. The ratio of changed labels in each batch gradually declines, indicating ODC tends to be stable during training.

Figure 6. The efficacy of ODC trained on downsampled 300-class 100K long-tail ImageNet, with the ratio of the size of largest class to smallest class ranging from 1 (*non-long-tail*) to 64 (*highly long-tail*). We report mAP on VOC07 SVM task with ResNet-50.

convergence of ODC throughout the full training iterations. To measure the stability of our models, we record the ratio of samples whose labels are changed in a batch. Intuitively, fewer label switchings suggest a higher stability. We report the ratio when different backbones are trained from scratch with ODC. The curves begin with the highest label-switching ratio, *i.e.*, nearly 100% of samples in a batch experience a switch in their labels. Gradually, the label-switching ratio declines and converges to a relatively low value. Though there is always a small portion of samples altering their labels at last, ODC reaches a stable state.

Training on Long-Tailed Data. In all previous experimens, we train our models on the class-balanced ImageNet dataset. To evaluate the learning efficacy of ODC on long-tailed data, we perform experiments on downsampled long-tail ImageNet following [64]. Specifically, we randomly downsample 300 classes with 100K images from the original ImageNet dataset to make different levels of long-tail ImageNet datasets, where the ratio of the largest class to the smallest class ranges from 1 (the non-long-tail level) to 64 (the highest long-tail level). Figure 6 shows the performance of ODC trained on different levels of long-tail ImageNet. We observe no significant performance drop even in the conditions with large long-tail degrees, suggesting the robustness of our method on long-tailed data.

Visualization of Clusters. We visualize some selected clusters as shown in Figure 7. Since the number of clusters is much larger than that of the original annotations, there will certainly be some clusters that represent new semantics beyond the annotated classes. We find new classes, *e.g.*, "hand" and "feet", and new relations, *e.g.*, "animal in cage", "person holds dog" and "person leads dog with a rope", that are discovered by ODC. The phenomenon reveals the potential of unsupervised learning to capture new semantics beyond manual annotations.

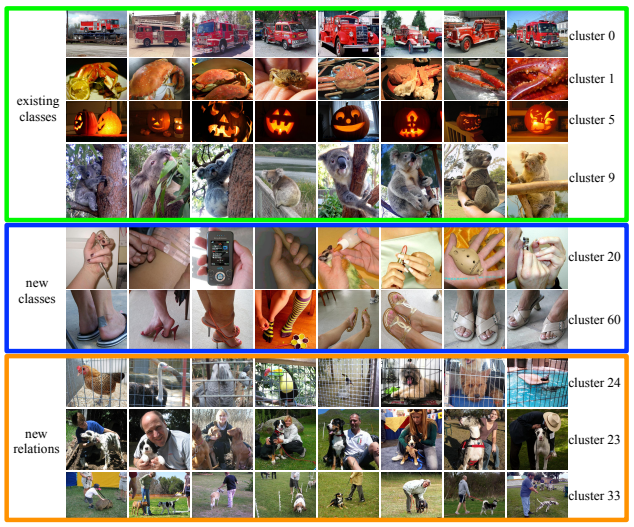

Figure 7. This figure shows part of selected clusters. Each row represents a cluster. Apart from the clusters that represents existing classes in ImageNet annotations, shown in the green box, we also find some new classes discovered by ODC. For example, the two rows in the blue box group "hand" and "feet" respectively, while "hand" or "feet" is not a category in ImageNet annotations. ODC also surprisingly groups images with similar relations between objects. As shown in the orange box, the clusters represent "animal in cage", "person holds dog" and "person leads dog with a rope" respectively.

5. Conclusion

We have proposed an effective joint clustering and feature learning paradigm for unsupervised representation learning. The proposed approach, Online Deep Clustering (ODC), attains effective and stable unsupervised training of deep neural networks, via decomposing feature clustering and integrating the process into iterations of network update. ODC performs compellingly as an unsupervised representation learning scheme alone. It can also be used to fine-tune and substantially improve previous self-supervised learning methods.

Acknowledgements. This work is supported by the SenseTime-NTU Collaboration Project, Singapore MOE AcRF Tier 1 (2018-T1-002-056), NTU SUG, NTU NAP, the Max Planck-NTU Joint Lab for Artificial Senses and Data Science and Artificial Intelligence Research Lab. We thank Yue Zhao for his participation in discussing the idea.

References

[1] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 1, 2, 6

[2] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 1, 2, 6

[3] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 1, 2, 6, 7

[4] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 1, 2, 6, 7

[5] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *ICLR*, 2017. 1, 2

[6] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, 2017. 1, 2

[7] Simon Jenni and Paolo Favaro. Self-supervised feature learning by learning to spot artifacts. In *CVPR*, 2018. 1

[8] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 1, 2, 6, 7

[9] Xiaohang Zhan, Xingang Pan, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised learning via conditional motion propagation. In *CVPR*, 2019. 1, 2

[10] Huang Huang, Chen Change Loy, and Xiaoou Tang. Unsupervised learning of discriminative attributes and visual representations. In *CVPR*, pages 5175–5184, 2016. 1

[11] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016. 1, 2

[12] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, 2016. 1, 2

[13] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 1, 2, 3, 5, 6, 7

[14] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, pages 2959–2968, 2019. 1, 2

[15] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 1

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 1

[17] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 1

[18] Xiaohang Zhan, Ziwei Liu, Junjie Yan, Dahua Lin, and Chen Change Loy. Consensus-driven propagation in massive unlabeled data for face recognition. In *ECCV*, 2018. 1

[19] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to cluster faces on an affinity graph. In *CVPR*, pages 2298–2306, 2019. 1

[20] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008. 2

[21] Quoc V Le. Building high-level features using large scale unsupervised learning. In *ICASSP*, 2013. 2

[22] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 2

[23] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, pages 609–616, 2009. 2

[24] Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. Robust boltzmann machines for recognition and denoising. In *CVPR*, pages 2264–2271, 2012. 2

[25] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014. 2

[27] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NeurIPS*, pages 1486–1494, 2015. 2

[28] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 4681–4690, 2017. 2

[29] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, pages 5907–5915, 2017. 2

[30] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. 2019. 2

[31] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, pages 613–621, 2016. 2

[32] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, pages 1526–1535, 2018. 2

[33] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. 2

[34] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *arXiv preprint arXiv:1907.02544*, 2019. 2, 6

[35] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *ECCV*, pages 577–593, 2016. 2

[36] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017. 2, 5, 6

[37] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, pages 5898–5906, 2017. 2, 6

[38] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *ICML*, pages 737–744, 2009. 2

[39] Dinesh Jayaraman and Kristen Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *CVPR*, pages 3852–3861, 2016. 2

[40] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, pages 2794–2802, 2015. 2

[41] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, pages 4463–4471, 2017. 2

[42] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, pages 667–676, 2017. 2

[43] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, pages 527–544, 2016. 2

[44] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *CVPR*, pages 8052–8060, 2018. 2

[45] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *CVPR*, pages 2701–2710, 2017. 2

[46] Aravindh Mahendran, James Thewlis, and Andrea Vedaldi. Cross pixel optical-flow similarity for self-supervised learning. In *ACCV*, pages 99–116, 2018. 2

[47] Jacob Walker, Abhinav Gupta, and Martial Hebert. Dense optical flow prediction from a static image. In *ICCV*, pages 2443–2451, 2015. 2

[48] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, pages 835–851, 2016. 2

[49] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NeurIPS*, pages 766–774, 2014. 2

[50] Renjie Liao, Alex Schwing, Richard Zemel, and Raquel Urtasun. Learning deep parsimonious representations. In *NeurIPS*, pages 5076–5084, 2016. 2

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

[52] T Nathan Mundhenk, Daniel Ho, and Barry Y Chen. Improvements to context based self-supervised learning. In *CVPR*, pages 9339–9348, 2018. 2

[53] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, pages 9359–9367, 2018. 3

[54] Xiaolong Wang, Kaiming He, and Abhinav Gupta. Transitive invariance for self-supervised visual representation learning. In *ICCV*, pages 1329–1338, 2017. 3

[55] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *ICCV*, pages 2051–2060, 2017. 3

[56] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. *arXiv preprint arXiv:1901.09005*, 2019. 3, 6

[57] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. *arXiv preprint arXiv:1905.01235*, 2019. 3, 5, 6, 7

[58] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 6

[59] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *CVPR*, pages 2547–2555, 2019. 6

[60] Zeyu Feng, Chang Xu, and Dacheng Tao. Self-supervised representation learning by rotation feature decoupling. In *CVPR*, pages 10364–10374, 2019. 5, 6

[61] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, pages 6002–6012, 2019. 5, 6

[62] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NeurIPS*, pages 487–495, 2014. 5

[63] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 5

[64] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019. 8