2022 26th International Conference on Pattern Recognition (ICPR)
August 21-25, 2022, Montréal, Québec, Canada

Online Deep Clustering with Video Track Consistency

Alessandra Alfani, Federico Becattini, Lorenzo Seidenari, Alberto Del Bimbo
University of Florence

2022 26th International Conference on Pattern Recognition (ICPR) | 978-1-6654-9062-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICPR56361.2022.9956284

***Abstract*—Several unsupervised and self-supervised approaches have been developed in recent years to learn visual features from large-scale unlabeled datasets. Their main drawback however is that these methods are hardly able to recognize visual features of the same object if it is simply rotated or the perspective of the camera changes. To overcome this limitation and at the same time exploit a useful source of supervision, we take into account video object tracks. Following the intuition that two patches in a track should have similar visual representations in a learned feature space, we adopt an unsupervised clustering-based approach and constrain such representations to be labeled as the same category since they likely belong to the same object or object part. Experimental results on two downstream tasks on different datasets demonstrate the effectiveness of our Online Deep Clustering with Video Track Consistency (ODCT) approach compared to prior work, which did not leverage temporal information. In addition we show that exploiting an unsupervised class-agnostic, yet noisy, track generator yields to better accuracy compared to relying on costly and precise track annotations.**

I. Introduction

Convolutional Neural Networks (CNNs) are being widely used as the main framework in many computer vision applications due to their great ability to learn different levels of general visual features. Over the past 10 years, their performance has continued to improve, thanks to complex architectures and large-scale datasets. The success of CNNs is, in fact, highly dependent on their capabilities to model highly non-linear patterns and the amount of training data available. The deeper and more expressive the network is, the larger the required dataset will be to effectively train such model.

Models trained on large-scale, fully-supervised and diverse image datasets, such as ImageNet [1] and YFCC100M [2], have been demonstrated to capture common and generic visual features, making them also ideal as a starting point for diverse subsequent learning tasks. According to this view, collecting and annotating in a precise way larger and more diversified datasets to be used in supervised machine learning seems a natural approach to go forward. However, on the other hand, this approach is extremely costly, time consuming and requires a massive amount of manual annotations. As a consequence, unsupervised learning has recently received attention in the field of machine learning and methods for clustering, dimensionality reduction, and density estimation are commonly used in computer vision applications [3], [4], [5], [6].

The goal of unsupervised representation learning is indeed to learn transferable image or video representations without the need for manual annotations [7], [8], [9], [10], [11], [12], [13]. Clustering-based representation learning approaches, which jointly optimize clustering and feature learning, stand out as a potential trend in this field [14], [15], [16], [17], [18], [19].

In this work we focus the attention on two unsupervised clustering-based learning methods, DeepCluster (DC) [17] proposed by Caron *et al.* and Online Deep Clustering (ODC) [19] proposed by Zhan *et al.*. Both these methods alternate between deep feature clustering and CNN parameters update, with the difference that ODC proposed to decompose the clustering procedure into mini-batch-wise label updates and incorporate these updates into network update iterations.

Despite the success of such clustering-based methods, they are mostly focused on still images and do not exploit additional sources of supervision such as temporal consistency in videos. Following this idea, we propose a novel unsupervised approach to exploit temporal information from videos, allowing the network to use tracks proposals as a form of self-supervision. Our method, dubbed Online Deep Clustering with Video Track Consistency (ODCT), is fully unsupervised and can be trained starting from any video database, without any prior knowledge regarding the nature of their content. To this end we leverage an unsupervised object discovery approach that extracts class-agnostic track proposals and exploit such data to constrain a feature clustering phase and generate temporal coherent pseudo-labels.

The main contributions of our work are the following:

- We present a fully unsupervised, class-agnostic method exploiting video temporal consistency of object tracks using feature clustering to generate pseudo-labels.
- We show how adding constrains on the clustering phase we can account for intra-track variability and obtain a more effective supervision signal.
- We test our model on two downstream tasks on different datasets, obtaining significant improvements compared to prior work which leverages only still images.

II. Related Work

Unsupervised and Self-Supervised learning are used in representation learning methods to generate feature representations merely from images, without the need for time-consuming semantic annotations. Self-Supervised methods, in particular, employ pre-designed pretext tasks and need no labeled data in order to train the weights of a convolutional neural network. Instead, the visual features are learned by minimizing the pretext task's objective function.

Several pretext tasks have been proposed in the literature and each of them was carefully designed to use various cues

from images or videos. Noroozi *et al.* proposed to learn image representations by solving Jigsaw puzzles [10]. *RotNet* [20] relies on predicting image rotations. Other approaches exploit temporal or image context prediction. Doersch *et al.* leverage image context [7] while Misra *et al.* seek supervision via temporal order verification [21].

A different line of work relies instead on generative or completion tasks. Zhang *et al.* learns representations via image colorization [9]; Pathak *et al.* use image inpainting instead [22]; Tulyakov relies on video generation with GANs [23] to learn features.

Several strategies for combining multiple cues have been proposed recently, Doersch *et al.* in [24] proved that combining diverse pretext tasks to simultaneously train a single trunk network, using a multi-task objective function, can improve the overall performance. However, these methods are domain-specific and require expert knowledge to properly design a pretext task that may result in transferable features.

Moreover, many approaches, such as inpainting or colorization, have the disadvantage of learning features on modified images, which may impair generalization to unmodified ones. Colorization, for example, takes a grayscale image as input, so the network cannot learn to extract color information, which might be useful for other tasks.

Clustering is often employed as a means to generate supervision via pseudo-labels [17], [25], [19]. Noroozi *et al.* propose to use k-means after a jigsaw based pretraining [25], a simple classifier is then trained from scratch on the gathered pseudo-labels. Caron *et al.* embed the clustering step in the training loop, thus updating the pseudo-labels during all training[17]. Zhang *et al.* improved over [17] by updating labels continuously rather than in a pulsating manner, enabling the representations to evolve steadily.

Some works try to avoid the use of top-down clustering. Instead of generating synthetic classes via grouping, Wu *et al.* propose to learn a representation capturing the similarity among instances by requiring learned features to be discriminative of individual instances [26]. Huang *et al.* propose a neighborhood discovery approach based on a divide-and-conquer strategy [27]. The proposed method is able to find class consistent neighbourhoods anchored to individual training samples which result in highly compact sample clusters.

To get the best of both worlds, Vangan *et al.* [28], combine a non-clustering based pre-training step based on a pretext task. The obtained features are used as a prior in a learnable clustering approach. This allows to remove the need for clustering to rely on low-level features.

In this work we follow the clustering-based pre-training strategies, in particular building upon the Online Deep Clustering (ODC) approach [19]. Differently from previous work though, we exploit videos to perform the pre-training stage and leverage temporal coherence as a source of supervision. By extracting object tracks from videos we are able to formulate a novel constrained clustering which incrementally builds pseudo-labels attempting to assign samples belonging to the same track to the same cluster in the feature space.

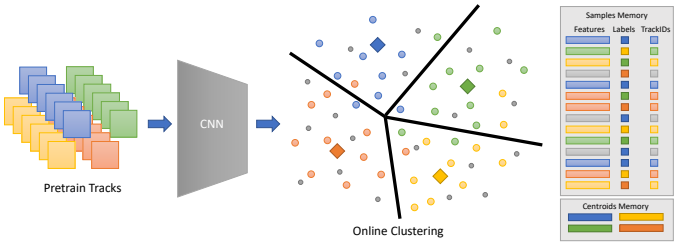

Fig. 1. Pseudo-labels are generated during training with online clustering. Track identities are stored in an external memory to constrain the clustering stage and make features of the same track belong to the same cluster.

III. Online Deep Clustering with Video Track Consistency

In this paper we extend Online Deep Clustering [19] by exploiting video temporal coherency as a self-supervised source for constraining the clustering stage (Fig. 1).

Given a video $V = f_0, ..., f_{N-1}$, where f_i is the i-th frame, we define a track as a sequence of K bounding boxes b_i^k, $k = 0, ..., K-1$. A track spatially encloses in its boxes the same object across a set of adjacent frames $[f_j, f_{j+K}]$. We do not assume to have any information regarding the class of such object, but we assume to have access to an oracle, capable of yielding tracks associated with different objects in a video.

Our goal is to train a classifier using pseudo-labels that are consistent at track level. The advantage of using tracks is that they naturally provide a supervision signal that links different views of the same object, therefore accounting for intra-class variability stemming from pose and viewpoint.

ODC originally relies on two external memories, the Samples Memory and the Centroids memory. Such memories are used to store features for training samples and their correspondence with the centroids of each cluster, i.e. the pseudo-labels used for training the model. We extend this formulation by adding a Track Memory where we store a track identifier for each sample. As in standard ODC, we populate the Samples Memory by extracting features from a randomly initialized CNN. Differently from [19], however, we consider each bounding box in a frame as a different sample. Therefore, features are extracted from image crops representing objects instead of from whole images. For the initial clustering step, we draw a representative at random from each track and perform K-means. We store the centroids in the Centroids Memory, propagating the cluster of each representative to all the other samples in the track. After the initialization step, we train the network following five steps:

(i) *Network Forward Propagation*: given a batch of samples x_i, the network extracts compact feature vectors $f_\theta(x_i)$ and a prediction $\tilde{y}_i$.

(ii) *Network Backward Propagation*: pseudo-labels p_i, associated with each sample, are retrieved from the Samples Memory and are used to compute the loss. The parameters of the CNN are updated accordingly after gradient computation.

(iii) *Weight Distances*: for each feature $f_\theta(x_i)$ extracted in the current batch, we retrieve from the Samples Memory

all the features belonging to samples x_j in the same track $F_m(x_j)$. Let x' be the sample in the track with the closest feature vector $F_m(x')$ to $f_\theta(x_i)$. We define weight coefficients d_j, that will be used to update the Samples Memory, as $d_j = \frac{d(x_i,x')}{d(x_i,x_j)}, \forall x_j \in T$

(iv) *Samples Memory Update*: each sample in the current batch is used to update the Samples Memory exploiting the weight coefficients d_j:

$$F_m \leftarrow m\frac{f_\theta(x)}{\parallel f_\theta(x) \parallel_2} + (1-m)\frac{\sum_j d_j F_m(x_j)}{\sum_j d_j} \quad (1)$$

where $m \in (0, 1]$ is a momentum coefficient. At the same time, each sample's pseudo-label is updated by finding its nearest centroid:

$$\arg\min_{p\in\{1..C\}} \|F_m - C_p\|_2^2, \quad (2)$$

where C_p indicates the centroid feature of class p in the Centroids Memory.

(v) *Centroids Memory Update*: every k-th iteration the centroids are updated as the mean of the feature vectors assigned to each cluster.

As in [19], to avoid trivial solutions where all samples are assigned to a single clusters, we prevent small clusters to become empty. Let C_s be the set of clusters with size less than a certain threshold. For $c \in C_s$, we start by assigning samples in c to the nearest centroids belonging to the remaining clusters $C_n = C \setminus C_s$, thus making c empty. Then, the largest cluster $c_{max} \in C_n$ is partitioned into two sub-clusters using *K-Means*. This process is repeated until the set of small clusters C_s is empty.

IV. Track Generation

To train our model, we assumed to have access to a set of video tracks. For the purpose of our work, these can be provided by some source of track oracle yielding linked bounding boxes in video frames. In the following Sec. V we will discuss experiments performed using ground truth tracks. Nonetheless, since we operate under an unsupervised data-regime, we also rely on an object discovery pipeline, derived from [29], capable of producing a set of class-agnostic tracks which can be exploited to train our model. The idea of such approach is to track spatio-temporal consistency in frame-wise object proposals. We use EdgeBoxes [30] to generate proposal boxes for each frame. These boxes are likely to enclose an object based on low level image characteristics, such as edges, which provide an *objectness* score for regions of interest. Note that this method does not require any class-specific training, thus can be applied to discover any kind of object or part of it. As in [29], we then use a greedy box-association tracker to link proposals across frames based on their spatial consistency through time. To reduce mismatches, we register each frame onto the next by shifting boxes using optical flow. More formally, we can define a track as a succession of bounding boxes b_i^k, for which the intersection over union (IoU) between two boxes of consecutive frames, b_i^n, b_{i+1}^m, is greater than a threshold θ_τ. We can define as well the track score S_{t_j} for track t_j, as the mean objectness score of its boxes $S_{t_j} = \frac{\sum_{i\in t_j} s_i^k}{|t_j|}$ where $|t_j|$ indicates the total number of boxes in track t_j. This naturally leads to a track ordering.

An issue with matching bounding boxes is that there might be missing frames and therefore track fragmentation due to occlusion or appearance changes. As in [30] we use a Time to Live counter (TTL), which represents the maximum number of frames without match before truncating a track. Finally, we apply a post-processing step to remove the tracks that are unlikely to represent objects. Particularly, we consider only tracks with a length between l and L, as we believe that shorter, more consistent tracks are preferable to longer tracks that are likely to be subject to errors, due to static background, noise, object occlusion or camera movement. l and L are empirically set to 40 and 100, respectively. In order to to retain only tracks that are likely to represent objects, we retain only the first 15 tracks in each snippet.

V. Experiments

In this Section, we present experimental results. In addition to our *ODC with Video Track Consistency* (ODCT) model, for the purpose of evaluation we report results from two additional methods. The first is vanilla ODC [19], trained on the same video data as ODCT. This does not integrate temporal knowledge from video snippets in any way and relies purely on clustering visually similar items. Then we propose a simple ODC variant with our modified *K-Means* procedure used for initialization, which constrains crops belonging to the same track to be assigned to the same cluster. We will refer to this model as ODC$_{TrackInit}$.

We apply a two-step training procedure, commonly adopted for unsupervised learning [17], [19]: (i) First, we perform unsupervised pre-training using the unsupervised pretext methods on a pre-training dataset. Usually, such dataset is large yet with limited or no annotation. (ii) We then perform transfer learning by fine-tuning the model on some downstream task, i.e. harder tasks with access to some form of supervision.

A. Datasets and Data Preprocessing

We conduct our pre-training experiments on the ImageNet VID 2015 dataset [31], a dataset designed for object tracking and object detection. It includes videos of 30 basic object categories, subset of the 1,000 ILSVRC-2012[1] classes. In total, ImageNet VID consists of 3,862 snippets for training, 555 snippets for validation, and 937 snippets for testing. All snippets include 56 to 458 frames of images, with a median frame rate and duration of 29*fps* and 12*s*, respectively. Each frame is annotated with labels indicating the presence of the 30 object classes and their corresponding bounding boxes. The dataset comprises a total of 7,857 ground truth tracks for training. In our experiments we will initially discard such annotations in order to work in a fully unsupervised regime, but we will also train our model using ground truth tracks as a control experiment in Sec. V-F. As downstream tasks we follow commonly adopted benchmarks as in [9], [17], [19]

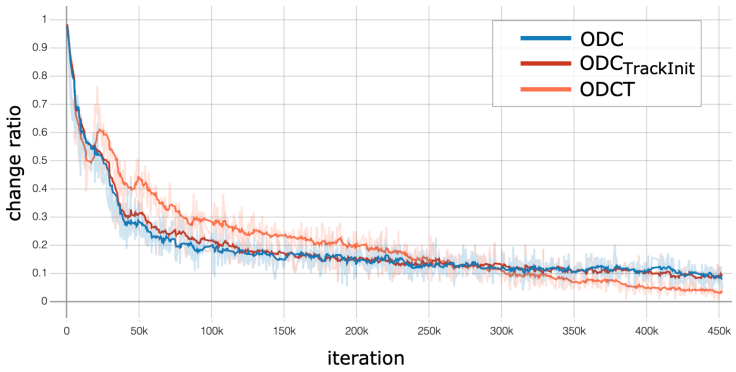

Fig. 2. Change ratio for sample-cluster assignments during training.

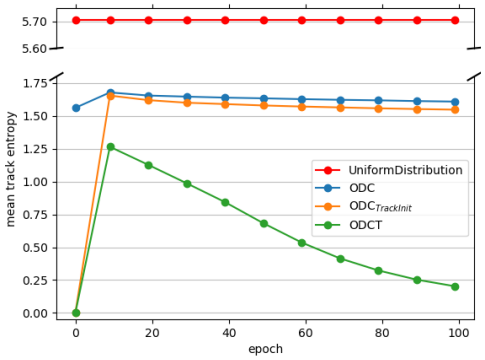

Fig. 3. Mean track entropy for cluster assignments during training. We also report the entropy for a Uniform Distribution over clusters as an upper bound.

and evaluate our model on ImageNet [1] and Pascal VOC 2007 [32].

We scale all frames to the same maximum dimension of 600px. As described in Sec. IV, we perform the object proposal phase with EdgeBoxes. To reduce noise, we limit the number of proposals per frame to 300, before applying non-maximal suppression (NMS) to select the best bounding box for each object. In the box tracking phase, we set the Time To Live counter (TTL) to 3 frames and the IoU threshold for bounding box matching to 0.35. Furthermore, we removed bounding boxes with an area smaller than 10,000 square pixels. We collected 57,879 tracks, each of which has been subsampled by picking 10 frames, equally spaced in time, for a total of 578,790 training samples. Object crops are resized to a resolution of 224x224 and data augmentation is applied including random flipping, rotation (±2°) and color jittering. In addition, we randomly convert images to grayscale with a probability of 0.2. Applying random color jitter and grayscale transformations on the training samples, discourages the network from exploiting trivial information from color to group samples snippet-wise.

B. Training Details

We employ ResNet-50 as model backbone and introduce a non-linear head to reduce hidden size dimensionality for storing features in the external memories. The head is composed of a 512-dimensional fully connected layer, followed by batch normalization, another 256-dimensional fully connected layer with relu activation and dropout. The head layer is removed for downstream tasks.

All the models are trained from scratch on a single GeForce RTX 2080 Ti GPU with mixed precision training for efficiency. The batch size and the learning rate are respectively set to 128 and 0.015 for 100 epochs using the SGD optimizer with momentum set to 0.5. The threshold to identify small clusters is set, as in ODC, to 20. Zhan *et al.* [19] proved that changing this threshold does not affect the results significantly, as long as it does not exceed the average number of samples in a cluster. Regarding the Centroids Memory update, we perform it every 10 iterations to balance learning efficacy and efficiency. Following Caron *et al.* [17], we use a number of clusters equal to 10 times the number of annotated categories in the dataset (K=300). However, as an ablation study, we also trained our model using K=30 and K=1000.

C. Preliminary evaluation

In a preliminary set of experiments, we analyze the models' behavior during training.

1) Cluster reassignments: To assess the models' stability, we keep track of the change ratio, i.e. the fraction of samples whose labels change at each iteration. Intuitively, fewer label switches indicate better stability. In Fig. 2 we illustrate the change ratio during training for ODC, ODC$_{TrackInit}$ and ODCT, trained from scratch, with $K = 300$ clusters. Models with $K = 30, 1000$ presented similar trends. Initially, almost 100% of samples undergo a label switch. The ratio decreases gradually and converges to ~ 0.1, thus indicating stability for 9 samples out of 10. Interestingly, ODC$_{TrackInit}$ and ODC have almost identical curves, while ODCT initially exhibits a higher change ratio but is able to eventually converge to a more stable solution (~ 0.04).

2) Cluster entropy: To assess how well the models assigns samples from the same track to the same cluster, we rely on an entropy based evaluation. Given a cluster assignment, for each track T we can determine the sample distribution over the K clusters, counting the occurrences of track samples per each cluster, $X_T = \{x_1, x_2, ..., x_K\}$. We then compute the Shannon entropy H, for X_T, as $H(X_T) = -\sum_{i=1}^{K} x_i log(x_i)$. This measures how noisy the distribution of track samples is over clusters. Ideally, samples of a same track should belong to the same cluster, yielding entropy $H(X_T) = 0$.

In Fig. 3 we report the mean track entropy computed for all the ImageNet VID track proposals during training. At the first epoch, we have $H(X_T) = 0$ for ODC$_{TrackInit}$ and ODCT since all the samples in a track are associated to the same cluster and same pseudo-label. ODC$_{TrackInit}$ mostly follows the same trend as ODC, meaning that initialization alone does not suffice to obtain an effective clustering during training.

TABLE I

TOP-1 CLASSIFICATION ACCURACY ON IMAGENET (LEFT) AND MAP ON PASCAL VOC 2007 (LEFT). ODC*[19]: ODC PRE-TRAINED ON IMAGENET. *ImageNet labels*: FULLY SUPERVISED RESNET-50 PRE-TRAINING ON IMAGENET. *Random*: RESNET-50 WITH RANDOM WEIGHTS. ALL OTHER MODELS ARE PRE-TRAINED ON IMAGENET VID WITHOUT SUPERVISION.

ImageNet	**Method**	Stage1	Stage2	Stage3	Stage4	Stage5
	ImageNet labels[19]	15.18	33.96	47.86	67.56	76.17
	Random[19]	11.37	16.21	13.47	9.07	6.54
	ODC*[19]	14.76	31.82	42.44	55.76	57.70
	ODC	10.63	21.74	23.01	22.86	16.65
K = 30	ODC$_{TrackInit}$	10.61	22.27	24.10	24.07	17.34
	ODCT	**10.91**	**23.50**	**27.37**	**30.26**	**23.38**
	ODC	9.95	21.92	23.70	23.35	16.74
K = 300	ODC$_{TrackInit}$	10.39	21.74	25.56	25.86	19.20
	ODCT	**11.02**	**24.54**	**28.06**	**32.23**	**26.00**
	ODC	11.14	21.25	23.65	23.14	16.80
K = 1000	ODC$_{TrackInit}$	11.39	23.11	24.82	25.93	19.53
	ODCT	**11.44**	**24.79**	**28.72**	**32.95**	**26.04**

VOC07	**Method**	Stage1	Stage2	Stage3	Stage4	Stage5
	ImageNet labels[19]	26.84	47.56	58.94	78.94	87.17
	Random[33]	9.60	8.30	8.10	8.00	7.70
	ODC*[19]	27.33	46.16	56.22	68.06	78.42
	ODC	24.80	39.36	40.02	36.72	30.85
K = 30	ODC$_{TrackInit}$	**24.86**	38.76	40.00	37.88	31.32
	ODCT	24.58	**39.74**	**42.80**	**43.83**	**38.28**
	ODC	24.01	38.67	39.57	37.55	31.62
K = 300	ODC$_{TrackInit}$	**24.87**	38.85	41.47	38.70	33.10
	ODCT	24.77	**41.00**	**44.56**	**45.55**	**41.38**
	ODC	24.22	38.36	39.56	37.57	31.58
K = 1000	ODC$_{TrackInit}$	24.74	39.63	40.50	38.77	33.37
	ODCT	**25.08**	**41.28**	**44.57**	**46.23**	**41.74**

Nonetheless, it still obtains a slightly lower entropy value than vanilla ODC, indicating that a good initialization indeed has a beneficial effect. ODCT, on the other hand, due to the constrained Samples Memory update, exhibits a completely different trend, managing to effectively lower the mean track entropy epoch by epoch.

D. Transfer Learning Experiments

As in [34], we extract image features from our ResNet-50 pretrained on Imagenet VID, truncating the model after the last layer of every residual stage in ResNet-50. In the following we discuss the downstream tasks we adopted to evaluate our pre-trained model and the quality of such features.

1) ImageNet Linear Classification: As proposed by Zhang *et al.* in [9], we test the task generalization of the representation by freezing the backbone's weights and training linear classifiers on top of each layer to perform 1000-way ImageNet classification. We report top-1 center-crop accuracy on the validation split of ImageNet. Results are shown in Tab. I (left). We compare our model against ResNet-50 trained directly on ImageNet in a fully supervised setting and againts a version with random weights, as proposed in [35]. Such methods act as an upper bound and lower bound for the task, respectively. We also report results for ODC pretrained on ImageNet, as reference. We compare our model to ODC pretrained on ImageNet VID for different number of clusters values. All models are trained for 100 epochs, using SGD with momentum of 0.9 and batch size of 64. The learning rate is initialized as 0.01 and decayed by a factor of 10 after every 30 epochs.

We can observe that ODCT Stage1 results in slightly higher accuracy than ODC and ODC$_{TrackInit}$, with the best improvement using $K = 300$. However, this small performance gap is immediately widened at Stage2, and our ODCT network achieves much higher results compared to standard ODC and ODC$_{TrackInit}$ pre-trained on ImageNet VID. As we use features extracted from deeper layers of the network, we can observe a consistent increase up to Stage4 for all methods. In particular, ODTC exhibits and improvement across all layers, with the best gain ($\sim$33%-40%) obtained with Stage4 and Stage5 and $K = 300, 1000$. These results indicate that exploiting video track information encourages representations that linearly separate semantic classes in the trained data distribution.

However, there is no significant improvement using $K = 1000$ compared to $K = 300$. This suggests that, after a certain value of K, the accuracy saturates. Interestingly, we can see that also for ODC$_{TrackInit}$ there is a considerable performance gain compared to vanilla ODC, proving that a good initialization can have a significant impact on the whole training process.

2) VOC2007 SVM Classification: We test our model on the classification benchmark PASCAL VOC2007[32]. Following the setup of Goyal *et al.* [33], we train linear SVMs on features extracted from the ResNet-50 backbone on the `trainval` split of VOC07. We follow the same configuration of [33] and report mean Average Precision (mAP) on the test set. The results in Tab. I (right), show that ODCT surpasses ODC, when both are pre-trained on ImageNet VID, by a significant margin on the VOC2007 SVM classification task.

We can observe that, ODCT yields comparable results to ODC$_{TrackInit}$ for $K = 30, 300$ for Stage1. However, starting from Stage2, ODCT achieves higher results throughout the remainder of the network compared to other unsupervised methods pretrained on ImageNet VID. Significantly, with $K = 1000$, ODCT achieves 46.23% mAP with Stage4, which is 6,67% higher than the best performing layer of ODC and 5,73% higher than the best performing layer of ODC$_{TrackInit}$. Also, our method gains over 10% mAP points with $K = 300, 1000$ compared to ODC on the last layer.

E. Qualitative Analysis

In Fig. 4 we show portions of the clusters obtained training ODCT using $K = 1000$. The number of clusters is much larger than that of the annotated classes and interestingly there are some clusters that represent new semantic categories beyond the annotated ones. In addition to the clusters that represent existing classes in ImageNet VID, shown in the orange box, we find additional subclasses (blue box), e.g. "white car", "car on racing track" or "dog with white snoug".

ODCT also groups images with similar relations between objects. As shown in the yellow box, the method discovers clusters representing "animal on a lawn" and "busses and trains", which have a similar visual appearance. Moreover, due

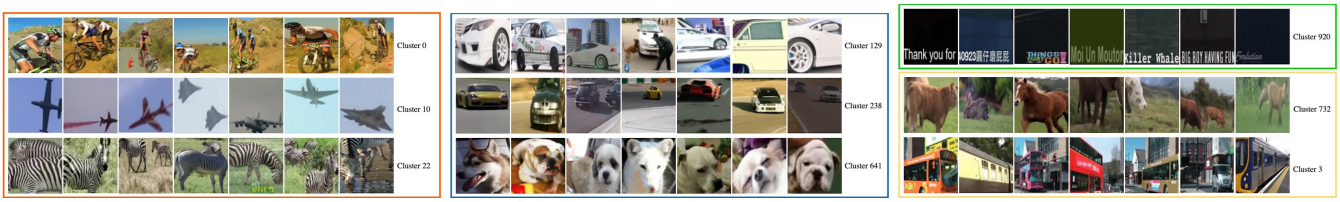

Fig. 4. Samples assigned to different clusters. Orange: existing classes. Blue: subclasses. Green: new classes. Yellow: object relations.

TABLE II
INTRA-TRACK AND INTRA-CLASS ENTROPY COMPUTED WITH GT TRACKS FROM IMAGENET VID. THE LOWER THE BETTER.

	Method	Intra-Track H	Intra-Class H
K = 30	ODC_{GT}	0.868	2.986
	$\text{ODC}_{GT-TrackInit}$	0.873	3.013
	ODCT_{GT}	**0.061**	**2.813**
K = 300	ODC_{GT}	1.368	4.963
	$\text{ODC}_{GT-TrackInit}$	1.360	4.969
	ODCT_{GT}	**0.064**	**4.099**

TABLE III
IMAGENET TOP-1 ACCURACY AND PASCAL VOC07 MAP FOR ODCT TRAINED ON UNSUPERVISED TRACKS AND GT TRACKS.

	Method	ImageNet top-1 acc	VOC07 mAP
K = 30	ODC	22.86	36.72
	$\text{ODC}_{TrackInit}$	24.07	37.88
	ODCT	**30.26**	**43.83**
K = 30	ODC_{GT}	19.75	33.77
	$\text{ODC}_{GT-TrackInit}$	20.01	33.36
	ODCT_{GT}	**22.49**	**36.14**
K = 300	ODC	23.35	37.55
	$\text{ODC}_{TrackInit}$	25.86	38.70
	ODCT	**32.23**	**45.55**
K = 300	ODC_{GT}	21.33	35.25
	$\text{ODC}_{GT-TrackInit}$	21.42	35.59
	ODCT_{GT}	**24.18**	**37.29**

to the unsupervised pre-training on videos, ODCT is capable to detect new classes such as "text on a dark background" (green box in Fig. 4).

F. Training with Ground Truth Tracks

Previous experiments demonstrated that ODCT is effective and achieves a significant improvement in transfer learning tasks when compared to vanilla ODC. Such results are obtained relying on a set of tracks generated without supervision, as outlined in Sec. IV. Since ImageNet VID also contains manually annotated tracks, we retrain our model using GT tracks (discarding class information) to highlight the difference between the two approaches. Our unsupervised track generation method yields a total of 57,879 tracks compared to the 7,857 ground truth tracks annotated in the dataset. Our tracks however, despite being an order or magnitude more, are likely to contain noise, be fragmented or focus on object parts or groups of objects. On the other hand, GT tracks are precise, clean and represent single objects in their entirety.

Following the same setting described in section V-B we train from scratch the three models on the GT ImageNet VID tracks[1], with $K = 30, 300$, discarding class labels. We refer to these models as ODC_{GT}, $\text{ODC}_{GT-TrackInit}$ and ODCT_{GT}.

We evaluate the models using two entropy measures:

(i) *Intra-Track Entropy* is used to evaluate the distribution of tracks over clusters. For each track T, we measure how a track is distributed over K clusters by computing the Shannon entropy as in Sec. V-C2, counting the occurrences of track samples per each cluster.

(ii) *Intra-Class Entropy* is a measure of how well a model assigns samples from the same class to the same cluster. For each class, we compute the class distribution over K clusters, counting the occurrences of class samples per each cluster, and then compute the Shannon entropy.

[1]We use 10% of equally spaced samples per track, a total of 77,714 samples.

In Tab. II we report the Intra-Track and Intra-Class entropies computed on all tracks. In both the experiments, the entropy for ODCT is lower, thus indicating that our approach assigns samples from the same tracks and classes to the same clusters better than the others. In particular, it is interesting to notice the significant drop in Intra-Track entropy compared to the other methods. This underlines the effectiveness of our clustering strategy, which manages to keep together samples belonging to the same track.

We additionally evaluate the effectiveness of the model trained on the GT tracks using the ImageNet and VOC07 downstream classification tasks. We use the same settings indicated in Sec. V-D1 and Sec. V-D2, considering only Stage4. Results are shown in Tab. III. Interestingly, there is not much difference among all methods using the ground truth annotations, with a gap of at most 2-3 points in either accuracy and mAP. At the same time, it is surprising to notice that unsupervised tracks yield much higher results when comparing models to their counterparts trained with GT tracks. This hints to the fact that, even if manually annotated tracks are clean and precise, it is better to train with more, possibly noisy, data. In addition, our unsupervised track generation can provide tracks from unseen classes that can help to perform a better pre-training.

VI. CONCLUSIONS

In this paper we have introduced an unsupervised clustering-based approach that exploits temporal consistency of video tracks to model intra-class variation. It emerges that simple track based initialization has beneficial effects and overall exploiting temporal information leads to important gains in accuracy compared to prior work based on still images.

REFERENCES

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[2] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m," *Communications of the ACM*, vol. 59, no. 2, p. 64–73, Jan 2016. [Online]. Available: http://dx.doi.org/10.1145/2812802

[3] A. Joulin, F. R. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1943–1950, 2010.

[4] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[5] F. Becattini, L. Seidenari, and A. Del Bimbo, "Indexing quantized ensembles of exemplar-svms with rejecting taxonomies," *Multimedia Tools and Applications*, vol. 76, no. 21, pp. 22 647–22 668, 2017.

[6] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[7] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1422–1430.

[8] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[9] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*. Springer, 2016, pp. 649–666.

[10] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European conference on computer vision*. Springer, 2016, pp. 69–84.

[11] L. Berlincioni, F. Becattini, L. Seidenari, and A. Del Bimbo, "Multiple future prediction leveraging synthetic trajectories," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 6081–6088.

[12] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782*, 2016.

[13] L. De Divitiis, F. Becattini, C. Baecchi, and A. D. Bimbo, "Disentangling features for fashion recommendation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2022.

[14] C. Huang, C. C. Loy, and X. Tang, "Unsupervised learning of discriminative attributes and visual representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[15] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*. PMLR, 2016, pp. 478–487.

[16] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5147–5156.

[17] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 132–149.

[18] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin, "Unsupervised pre-training of image features on non-curated data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[19] X. Zhan, J. Xie, Z. Liu, Y.-S. Ong, and C. C. Loy, "Online deep clustering for unsupervised representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6688–6697.

[20] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

[21] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: unsupervised learning using temporal order verification," in *European Conference on Computer Vision*. Springer, 2016, pp. 527–544.

[22] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2536–2544.

[23] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1526–1535.

[24] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2051–2060.

[25] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash, "Boosting self-supervised learning via knowledge transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9359–9367.

[26] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742.

[27] J. Huang, Q. Dong, S. Gong, and X. Zhu, "Unsupervised deep learning by neighbourhood discovery," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2849–2858.

[28] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool, "Scan: Learning to classify images without labels," in *European Conference on Computer Vision*. Springer, 2020, pp. 268–285.

[29] G. Cuffaro, F. Becattini, C. Baecchi, L. Seidenari, and A. Del Bimbo, "Segmentation free object discovery in video," in *European Conference on Computer Vision*. Springer, 2016, pp. 25–31.

[30] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European conference on computer vision*. Springer, 2014, pp. 391–405.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[32] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, p. 303–338, Jun. 2010. [Online]. Available: https://doi.org/10.1007/s11263-009-0275-4

[33] P. Goyal, D. Mahajan, A. Gupta, and I. Misra, "Scaling and benchmarking self-supervised visual representation learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6391–6400.

[34] R. Zhang, P. Isola, and A. A. Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1058–1067.

[35] P. Krähenbühl, C. Doersch, J. Donahue, and T. Darrell, "Data-dependent initializations of convolutional neural networks," *arXiv preprint arXiv:1511.06856*, 2015.