SLIC: Self-Supervised Learning with Iterative Clustering for Human Action Videos

Salar Hosseini Khorasgani* Yuxuan Chen* Florian Shkurti
University of Toronto
{salar.hosseinikhorasgani, yuxuansherry.chen}@mail.utoronto.ca, florian@cs.toronto.edu

Abstract

Self-supervised methods have significantly closed the gap with end-to-end supervised learning for image classification [13, 24]. In the case of human action videos, however, where both appearance and motion are significant factors of variation, this gap remains significant [28, 58]. One of the key reasons for this is that sampling pairs of similar video clips, a required step for many self-supervised contrastive learning methods, is currently done conservatively to avoid false positives. A typical assumption is that similar clips only occur temporally close within a single video, leading to insufficient examples of motion similarity. To mitigate this, we propose SLIC, a clustering-based self-supervised contrastive learning method for human action videos. Our key contribution is that we improve upon the traditional intra-video positive sampling by using iterative clustering to group similar video instances. This enables our method to leverage pseudo-labels from the cluster assignments to sample harder positives and negatives. SLIC outperforms state-of-the-art video retrieval baselines by +15.4% on top-1 recall on UCF101 and by +5.7% when directly transferred to HMDB51. With end-to-end finetuning for action classification, SLIC achieves 83.2% top-1 accuracy (+0.8%) on UCF101 and 54.5% on HMDB51 (+1.6%). SLIC is also competitive with the state-of-the-art in action classification after self-supervised pretraining on Kinetics400.

1. Introduction

Self-supervision tasks have emerged as effective pretraining methods for image classification, retrieval, and other downstream tasks. They have also been shown to outperform end-to-end supervised learning in a number of settings [13, 34]. A key assumption in many self-supervised methods is the ability to do instance discrimination, by sampling or generating similar and dissimilar data, given a query. While this is a reasonable assumption for training image representations, for example by generating similar data through augmentations and dissimilar data by random sampling, it becomes much more challenging in the case of video representations as we need to account for motion-based as well as appearance-based similarities.

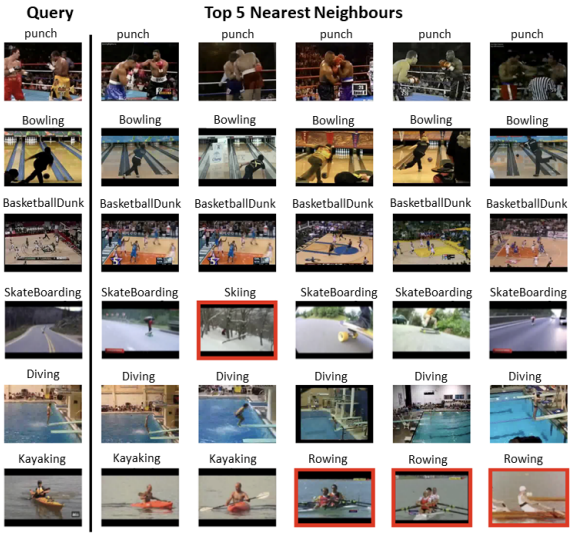

Figure 1. **Nearest neighbor retrieval results** of SLIC after pretraining on UCF101. The leftmost column is the query video from the UCF101 test set. On the right are the top 5 nearest neighbours from the training set in UCF101. Incorrect retrieval results are highlighted in red.

Different clips might have dissimilar appearances and the same motion (e.g. running at different locations), or similar appearances but different motions (e.g. playing cricket or golf with a green field as the background). In instance discrimination we sample positive video instances conservatively, typically by assuming that they only occur temporally close within a single video, and sample negatives randomly from different videos. Even though some methods attempted to improve upon instance discrimination by sampling harder positives using additional views, self-supervised pretraining of video representations is still not as effective as fully supervised pretraining (when evaluating on downstream clas-

*The two authors contributed equally to this paper.

sification using only visual inputs) [28, 52, 58].

To mitigate this, we propose SLIC, a self-supervised learning method for videos. SLIC alternates between periodically clustering video representations to produce pseudo-labels, and using those pseudo-labels to inform the sampling of positive and negative pairs to update the video representations, by minimizing a triplet margin loss. SLIC also combines iterative clustering with multi-view encoding and a temporal discrimination loss to learn view-invariant embeddings and fine-grained motion features, in order to distinguish the additional aspect of similarity that arises from the temporal dimension. Figure 2 shows an overview of our method.

Our main contributions are twofold. First, we show that iterative clustering significantly improves upon traditional instance discrimination in self-supervised learning for video representation. While this has already been established for image representations [8, 9], it has not been examined carefully for videos. Our method is the first to leverage efficient iterative clustering for video representation, specifically for sampling harder positives and negatives for contrastive learning. Second, we integrate iterative clustering with multi-view encoding and a temporal discrimination loss to sample harder positives and negatives during pretraining. We demonstrate that the interaction of these components, which has not been carefully examined by previous methods, is beneficial.

Our experiments show that SLIC achieves state-of-the art results on video retrieval ($+15.4\%$ improvement on top-1 recall on UCF101 and $+5.7\%$ on HMDB51 respectively, as shown in Table 1), and action classification when pretrained on UCF101. When finetuning end-to-end for action classification, we observe significant gains from pretraining instead of using a random initialization of weights (about $+24\%$ on top-1 accuracy on both UCF101 and HMDB51, as shown in Table 2). We demonstrate through additional experiments that all three components (i.e. iterative clustering, multi-view encoding, and temporal discrimination loss) complement each other, and result in larger performance improvements when combined. We evaluate the individual contribution from each component in the ablation studies, and identify that iterative clustering and multi-view encoding are the main contributing factors in SLIC (shown in Table 3).

2. Related work

Pretext Tasks and Losses for Self-Supervision: Several pretext tasks have been used in the past to provide training signals for image-based self-supervision, including predicting the position of an image patch relative to the first [17], solving jigsaw puzzles with shuffled image patches [40], performing colorization [18, 54, 65], inpainting of missing pixels [42], or predicting image rotations [23]. In the case of video data, these tasks can be extended to include predicting the direction of time in the shuffled frames of a video [57], and predicting the playback rate of the video [6, 56, 62]. Our proposed method does not rely on any of these pretext tasks.

Many of the tasks mentioned above were superseded by contrastive learning approaches, using the triplet loss [45], multiclass N-pairs loss [49], and variants of noise contrastive estimation [25, 26, 31, 41, 44, 53, 61], where the task is to discriminate between noisy and observed data. In particular, [26, 44] attempt to recurrently predict the representation of future frames using previous frames, then train the network to contrast the predicted representation against the ground truth representation and a pool of distractors. Also, [46] uses a triplet loss to attract images from multiple videos concurrently recording different viewpoints of the same observation. In addition, [24] shows that negatives are not needed to achieve comparable performance with self-supervised contrastive methods in image classification. While we experimented with contrastive losses [13, 14, 28, 53], we observed lower performance on video retrieval accuracy with iterative clustering, so our method relies on two triplet losses instead.

Clustering-based Self-Supervision: Using clustering as part of the pretraining process in self-supervision has been examined in multiple prior works. For example, ClusterFit [60] clusters the features computed by a pretrained CNN using K-means and then trains a new network using the resulting cluster assignments as pseudo-labels, showing notable improvements in transfer tasks. Our method differs in two ways: (a) we do not rely on a pretrained CNN to obtain features for clustering and instead use iterative clustering steps, which Fig. 3 shows to reduce false positives, and (b) we do not rely on K-means and thus we do not need to pre-specify K.

DeepCluster [8, 9] showed that the idea of using iterative clustering through K-means brings significant improvements in large-scale image classification, while our work addresses self-supervised activity recognition and retrieval, instead. Our idea is also similar to Prototypical Contrastive Learning (PCL) [37], which learns cluster prototypes and optimizes the InfoNCE loss [53] in an Expectation Maximization loop. PCL differs from our work in that it addresses image classification only, and uses a different loss for contrastive learning than what we use here. It is worth mentioning here that [32] presents a neural clustering method that does not compute cluster centers and thus does not require a pre-defined distance metric or number of clusters. Similarly, [10, 63] optimize clustering using an optimal transport solver, and a contrastive learning objective, but only address image classification and require a known number of labels, while our method does not. Our method also shares similarities with [47] which clusters face representations, however there is again the difference that we do not cluster the features from any pretrained networks and we progressively update the features used for clustering.

Multimodal and Multiview Self-Supervision: Multiple modalities for self-supervision have been used as different

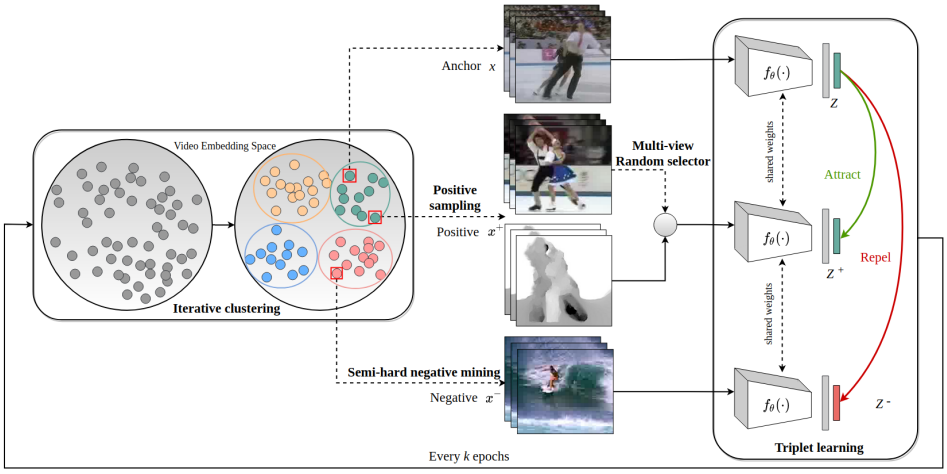

Figure 2. **Overview of the proposed self-supervised learning with iterative clustering framework (SLIC)**. We extract features using a deep 3D CNN and perform clustering every k epochs in the feature space to obtain cluster assignments. The cluster assignments are used as pseudo labels to sample positives and negatives for triplet learning. There is a fixed probability of replacing the positive (RGB view) with its corresponding optical flow view.

views of the data. For example audio and video [2–5, 35, 55], or video, audio and text [1]. In particular, [3, 5] rely on both audio and visual inputs to produce clustering assignments for cross-entropy classification (pre-specified number of clusters). In contrast, our method only uses visual inputs.

Co-training based on RGB videos and optical flow was examined in CoCLR [28], which is one of the best-performing methods for video representation. [44] also adopts a multi-view training scheme similar to that of CoCLR [28] by ensuring that the embeddings of different views such as flow, segmentation masks, and poses are consistent with respect to their distances between the same clips.

Action Recognition from Video: The main approaches for supervised learning for video action recognition rely on 3D-CNN architectures [21, 30, 52] that assume either single-stream networks [20, 52, 58] or multi-stream networks (e.g. separate streams for RGB and optical flow inputs) [16, 21, 22, 48, 66]. Our method belongs in the former category since we use one shared encoder network to process both RGB and optical flow inputs.

3. Method

Our goal is to learn feature representations from videos in a self-supervised manner. We propose an iterative clustering based method for video representation learning. The core elements of the proposed framework consist of the following: i) performing gradient updates using a triplet margin loss, and (ii) acquiring pseudo-labels from the cluster assignments to sample triplets. We incorporate two loss functions to optimize the encoder: an instance-based triplet loss and a temporal discrimination loss. Furthermore, we encourage the clustering to discover motion-based similarities by incorporating multi-view encoding of the RGB and optical flow views (using a single encoder). The pseudo code is presented in Algorithm 1 in Appendix Section D.

3.1. Iterative Clustering

To generate pseudo labels for the training set, we adopt the FINCH [39] algorithm to obtain pseudo-labels from clustering the video embeddings. FINCH discovers groupings in the data by linking the first neighbor relations of each sample, and hence does not require any prior knowledge of the data distribution. FINCH is more suitable for our task compared to other clustering methods such as K-means and DBSCAN [19] as it does not involve any hyper-parameter tuning or prior specification of the number of clusters, and it is considerably faster when handling large datasets.

FINCH computes the first neighbor k_i^1 for each video instance x_i in the feature space using the cosine distance metric. For instance, $k_i^1 = j$ means that x_j is the first neighbor of x_i. It then generates an adjacency link matrix $A(i,j)$ via Equation 1.

$$A(i,j) = \begin{cases} 1, \text{if } j = \kappa_i^1 \text{ or } k_j^1 = i \text{ or } \kappa_i^1 = \kappa_j^1 \\ 0, \text{otherwise} \end{cases} \quad (1)$$

The adjacency matrix links each video instance x_i to its first neighbors via $j = \kappa_i^1$, enforces symmetry through $i = \kappa_j^1$, and links video instances that share a common first neighbor with $\kappa_i^1 = \kappa_j^1$. Clustering is performed by recursively merging the connected components obtained from the adjacency matrix $A(i,j)$ in a hierarchical fashion. The output of the FINCH clustering algorithm is a small hierarchy of partitions that capture groupings in the underlying data structure at different levels of granularity, where each successive partition is a superset of the preceding partitions. Partition 1 is a flat partition of data generated via Equation 1, which

consists of a large quantity of small clusters at high purity. Due to it being the partition with the highest purity and to reduce the likelihood of sampling false positives, we use the cluster labels from the first partition to provide pseudo-labels for positive and negative mining.

SLIC periodically performs FINCH clustering in the feature space during training every k epochs. After applying FINCH, we update the pseudo label set $\{\hat{y}_i\}$ from the first partition P_1, where $\hat{y}_i \in \{1, 2, ..., C_{P_1}\}$, and C_{P_1} is the total number of clusters generated in partition P_1. As an alternative to FINCH, we also experiment with K-means and spherical K-means as baselines (with different values of K). In Section 4.5, we demonstrate that when a large enough number of clusters is used, K-means and spherical K-means result in comparable performance to that of FINCH but at a higher computational cost. We quantify the clustering quality by computing the Normalized Mutual Information (NMI) between the pseudo labels generated by the clustering algorithm $\{\hat{y}_i\}$ and the ground truth labels $\{y_i\}$.

$$NMI(\{\hat{y}_i\}, \{y_i\}) = \frac{I(\{\hat{y}_i\}, \{y_i\})}{\sqrt{H(\{\hat{y}_i\})H(\{y_i\})}} \quad (2)$$

where $I(\cdot, \cdot)$ and $H(\cdot)$ are the mutual information and the entropy respectively. We show in Section 4.4 that clustering quality improves throughout the training.

3.2. Instance-based Triplet Loss

We adopt the triplet margin loss and use it in conjunction with the pseudo labels attained from clustering, which allow us to sample harder positives and negatives during training. Let the dataset be denoted as D with N videos, i.e. $D = \{x_1, x_2, ..., x_N\}$, the goal is to learn an encoder $f_\theta(\cdot)$ that enforces the distance between two similar video clips to be closer than the distance between a dissimilar pair in the feature space. Given a triplet that consists of an anchor x, positive x^+, and negative x^-, we want to compute $f_\theta(\cdot)$ to maximize the distance between x and x^- while minimizing the distance between x and x^+, i.e. $d(f_\theta(x), f_\theta(x^-)) > d(f_\theta(x), f_\theta(x^+))$, where $d(\cdot)$ is the cosine distance $d(f_1, f_2) = 1 - \frac{f_1 \cdot f_2}{\|f_1\| \|f_2\|}$. The triplet margin loss is defined as follows:

$$\mathcal{L}_{triplet}(x, x^+, x^-; \theta, m_1) = \max(0, d(f_\theta(x), f_\theta(x^+)) - d(f_\theta(x), f_\theta(x^-)) + m_1) \quad (3)$$

where m_1 is a hyper-parameter that enforces a margin between anchor-positive and anchor-negative distance pairs.

Positive mining. Let x_i and x_j be two different instances sampled from the dataset D, where x_i is the anchor, and x_j belongs to the same cluster as x_i. Many prior works [15, 43, 51] sample the positive and anchor as two non-overlapping clips from the same video, which is known as instance discrimination. Although augmented differently, these two clips have limited diversity in objects and scenes since they are sampled from the same video. Thus, it is desirable to use different videos from the same semantic class as positives. However, due to the instability of clustering quality in the early stages of training, we might sample false positives (i.e. positives sampled from the same cluster but belonging to different semantic classes) that could be detrimental to the training. Hence, we leverage both methods by sampling positives from the same instance with probability p_α and from different instances with probability $(1 - p_\alpha)$:

$$x^+ = \alpha x_i^+ + (1 - \alpha) x_j^+, \quad \alpha \sim \text{Bernoulli}(p_\alpha) \quad (4)$$

where x_i^+ is a sampled augmented clip from the same video instance as the anchor x_i, and x_j^+ is a sampled augmented clip from a different instance, x_j, which belongs to the same cluster as x_i. p_α is a hyper-parameter that dictates how often the positives are sampled from the same instance or from different instances within the same cluster.

Multi-view positives. We denote video data from two different views as x and $\texttt{view}(x)$. For our method, optical flow is used as the second view in addition to RGB to encourage learning motion-based features. We use a shared encoder to process both input views to achieve faster inference speed with less memory consumption. To learn view-invariant embeddings, the features $f_\theta(x)$ and $f_\theta(\texttt{view}(x^+))$ should be close to each other in embedding space, since the features are extracted from similar videos. Thus, we sample the original RGB clip x^+ as the positive with probability p_β, or replace the RGB clip with another $\texttt{view}(x^+)$ with probability $(1 - p_\beta)$. This is written as:

$$x^+ = \beta x^+ + (1 - \beta)\texttt{view}(x^+), \quad \beta \sim \text{Bernoulli}(p_\beta) \quad (5)$$

Negative mining. Most prior works [15, 28, 43, 51] randomly select a large batch of negatives for contrastive instance discrimination, where the easiest 95% of the negatives do not contribute to the training and the hardest 0.1% are usually in-class negatives [7]. Easy negatives are further apart from the anchor than positives, which would evaluate to a loss of zero, whereas the in-class negatives refer to the negatives that belong to the same semantic class as the anchor, and could be detrimental to the training. On the other hand, hard negatives are closer to the anchor than the positive and satisfy $d(f_\theta(x), f_\theta(x^-)) < d(f_\theta(x), f_\theta(x^+))$. To avoid easy negatives and in-class negatives, we use pseudo labels generated by the cluster assignments to sample semi-hard negatives (within the mini-batch) that satisfy the constraint:

$$d(f_\theta(x), f_\theta(x^-)) \leq d(f_\theta(x), f_\theta(x^+)) + m_1 \quad (6)$$

where x^- belongs to a different cluster than x and x^+.

3.3. Temporal Discrimination Loss

In order to distinguish fine-grained temporal actions, we propose a temporal discrimination loss. This loss is similar

to the local-local contrastive loss proposed by TCLR [15], but has the key difference that a much greater variety of negatives can be sampled across different video instances as a result of the pseudo-labels from clustering. Given an anchor clip x_i, the positive clip is designated as the spatial augmentation $\texttt{aug}(x_i)$ of the anchor clip. The negative is any temporally non-overlapping clip from the same instance or, unlike TCLR [15], from a different instance in the same cluster. The temporal discrimination loss is defined as:

$$\mathcal{L}_{temporal} = \mathcal{L}_{triplet}(x_i, \texttt{aug}(x_i), x^+; \theta, m_2) \quad (7)$$

where m_2 represents the margin parameter for the temporal discrimination loss. It should be noted here that this loss pushes away x^+ from x_i whereas the instance-based triplet loss pulled it closer. To ensure that the difference between the anchor-positive and anchor-negative distances in this loss are smaller than those in the instance-based loss, m_2 is set to be less than m_1. Overall, the combined loss function is:

$$\mathcal{L}_{total} = \mathcal{L}_{triplet} + \lambda \mathcal{L}_{temporal} \quad (8)$$

where λ is a weighting parameter to balance the triplet loss and the temporal discrimination loss.

4. Experiments

4.1. Setup and Settings

Datasets. We conduct various experiments to evaluate the proposed model and its transferability on unseen data. We use two video action recognition datasets for pretraining the model: UCF101 [50] and Kinetics400 [33]. We use UCF101 [50] and HMDB51 [36] for finetuning on downstream evaluation tasks. The UCF101 dataset consists of 13,320 videos spanning 101 classes, HMDB51 consists of 7000 videos in 51 action categories, and Kinetics400 consists of 222,454 training videos in 400 action classes. For self-supervised pretraining, we use the UCF101 training set (split-1) and Kinetics400 training set without any class labels. For downstream evaluation tasks, we benchmark on split-1 of both UCF101 and HMDB51.

Implementation details. Unless stated otherwise, we use R3D-18 [29] as the feature extractor for all experiments due to its common use in the literature. As done in SimCLR [13], we attach a non-linear projection head (2-layer MLP) during training, with a hidden size of 2048 and an output dimension of 128. We maintain the projection head when evaluating action retrieval, and remove it for action classification. For both training and evaluation, we use clips of 16-frames or 32-frames (consecutive frames from 25fps videos) with a spatial resolution of 128 x 128 pixels as inputs. We apply clip-wise consistent spatial augmentations including random cropping, random scaling, horizontal flipping, color jittering, color dropping, and Gaussian blurring. We also apply random temporal cropping that crops the input video clips at random time stamps. This helps the model leverage the natural variation of the temporal dimension. For multi-view positive sampling, optical flow (u) is computed using the unsupervised TV-L1 algorithm [64], with the same pre-processing procedure as in [11]. We duplicate the data from the second view to generate input clips with channel dimension 3. We provide more details on the experimental setup in Appendix Section B.

4.2. Evaluation on Video Retrieval

In this protocol, we directly evaluate the extracted 128-dimensional representations from the self-supervised pretraining by performing nearest neighbour (NN) retrieval without any supervised fine-tuning. We follow the protocol used in prior work [15, 27, 28, 38, 59], where videos from the test set are used as the query for retrieving the k-nearest neighbours from the training set. For every query video, we average the output representations over 10 uniformly spaced clips. For every training video, we use the representation from a clip at a random time step. For both query and training videos, spatial center-crops of the frames are used. The retrieval results are presented for both UCF101 and HMDB51 in Table 1. In Appendix Section F, we also provide the retrieval results from discarding the projection head and using the intermediate embeddings, which attains similar performance on HMDB51 and UCF101.

Comparison with state-of-the-art. Similar to prior methods, we evaluate video retrieval using the top-k retrieval accuracy (recall@k), which is defined as the accuracy that at least one of the k-nearest neighbours from the training set belongs to the same semantic category as the query video. As shown in Table 1, our proposed method achieves 71.6% top-1 accuracy on UCF101, which is 15.4% higher than the current state-of-the-art. The direct transferability of the trained model is evaluated on the HMDB51 dataset. SLIC achieves 28.9% top-1 retrieval accuracy, which is 5.7% higher than the current state-of-the-art performance from a single-stream network. As a reference, we also report the performance from a supervised version of SLIC which does not use iterative clustering and instead samples positives and negatives using semantic labels. The gap with this supervised model has been reduced to 9.4% for R@1 on UCF101 and 4.1% for R@1 on HMDB51 (using 32 frames as the temporal input size). Lastly, we include the results obtained from using S3D [58] as the feature extractor to demonstrate that SLIC generalizes to both ResNet and Inception-based backbones.

Qualitative results. Figure 1 visualizes 6 query videos sampled from the test set and their top 5 nearest neighbors from the training set for UCF101. SLIC is able to retrieve similar videos based on their high-level semantics, and we observe that the incorrect retrievals generally share similar motion patterns and scene dynamics with the query video.

Table 1. **Nearest neighbour video retrieval results** on UCF101 and HMDB51 (both split-1). Testing set clips are used as queries to retrieve the top-k nearest neighbors in the training set, where $k \in [1, 5, 10, 20]$. Models with † use both RGB and optical flow as inputs for retrieval. "Supervised SLIC" is trained using a supervised triplet loss with positives and negatives sampled using semantic labels.

			UCF101				HMDB51			
Method	Input Size	Arch.	R@1	R@5	R@10	R@20	R@1	R@5	R@10	R@20
VCP [38]	16×112^2	R3D-18	18.6	33.6	42.5	53.5	7.6	24.4	36.3	53.6
VCOP [59]	16×112^2	R3D-18	14.1	30.3	40.4	51.1	7.6	22.9	34.4	48.8
IIC [51]	16×112^2	R3D-18	36.5	54.1	62.9	72.4	13.4	32.7	46.7	61.5
IIC† [51]	16×112^2	R3D-18	42.4	60.9	69.2	77.1	19.7	42.9	57.1	70.6
MemDPC [27]	40×128^2	R3D-18	20.2	40.4	52.4	64.7	7.7	25.7	40.6	57.7
CoCLR-RGB [28]	32×128^2	S3D-23	53.3	69.4	76.6	82.0	23.2	43.2	53.5	65.5
CoCLR† [28]	32×128^2	S3D-23	55.9	70.8	76.9	82.5	26.1	45.8	57.9	69.7
TCLR [15]	16×112^2	R3D-18	56.2	72.2	79.0	85.3	22.8	45.4	57.8	73.1
SLIC	16×128^2	S3D-23	60.9	73.5	78.6	84.0	21.8	46.3	59.7	72.4
SLIC	32×128^2	S3D-23	69.8	79.2	83.2	87.2	26.8	**52.9**	**66.2**	**78.1**
SLIC	16×128^2	R3D-18	66.7	77.3	82.0	86.4	25.3	49.8	64.9	76.1
SLIC	32×128^2	R3D-18	**71.6**	**82.4**	**86.6**	**90.3**	**28.9**	52.8	65.4	77.8
Supervised SLIC	16×128^2	S3D-23	67.7	73.2	74.8	76.0	17.3	39.4	53.7	67.7
Supervised SLIC	32×128^2	S3D-23	72.5	79.1	80.9	83.9	19.1	45.1	58.0	70.7
Supervised SLIC	16×128^2	R3D-18	75.3	80.5	82.8	84.9	26.8	51.2	65.1	76.3
Supervised SLIC	32×128^2	R3D-18	81.0	84.9	86.5	88.6	33.0	57.4	70.0	82.7

4.3. Evaluation on Action Classification

We pretrain the model on UCF101 and Kinetics 400, then evaluate the pretrained model for action recognition on UCF101 and HMDB51 following the protocol from prior work [28]. After self-supervised pretraining, we discard the non-linear projection head and attach a linear classifier, then evaluate the model under two settings: i) linear probing with a frozen backbone, and ii) fine-tuning the network end-to-end. The model is trained using the supervised cross-entropy loss, while applying all the data augmentations mentioned in Section 4.1 except for Gaussian blur. We use a learning rate of 10^{-3} for the linear classifier, and 10^{-4} learning rate for the backbone (when not frozen). For both i) and ii), the learning rate is decayed by a factor of 0.1 at epoch 60 and 100, and the training is carried out for 150 epochs. During inference, we follow the same procedure as [27, 28] and average the probabilities from 10 spatial crops (center-crop plus four corners, with horizontal flipping), and all temporal crops from a moving window with half temporal overlap.

Comparison with state-of-the-art. Comparison with the results from prior works are presented in Table 2. When pretrained on UCF101, SLIC outperforms the current state-of-the-art on end-to-end finetuning and linear probing using a temporal window of 32 frames. Compared to our fine-tuning results from using randomly initialized weights, self-supervised pretraining on UCF101 provides significant improvement in classification accuracy for both UCF101 and HMDB51. Furthermore, to examine how action classification performance would vary if semantic label information was used during pretraining, we perform end-to-end finetuning and linear probing on Supervised SLIC. Compared to SLIC, Supervised SLIC pretraining improves linear probing accuracy by 2-5% (depending on temporal input size and dataset) on UCF101 and HMDB51, but only provides marginal improvements for end-to-end finetuning. When pretrained on the larger Kinetics400 dataset, SLIC achieves comparable performance to TCLR [15] but underperforms compared to CoCLR-RGB [28]. It is worth noting that, while 32 frame inputs were used during finetuning, 16 frame inputs were used during Kinetics400 pretraining for faster computation.

4.4. Evolution of Cluster Assignments and Retrieval Accuracy

This section demonstrates the evolution of the clustering quality, and a simplified top-k retrieval accuracy which only uses the center 16 or 32 frames of query (test) and training videos as inputs. This simplified retrieval accuracy was used to monitor training progress since it is faster to compute than the final metric described in Section 4.2. As shown in Figure 3, the retrieval accuracy plateaus by the end of training when the clustering quality (the NMI between the cluster assignments and the ground-truth semantic labels) has nearly stabilized. Thus, the embeddings gradually improve in their ability to represent information relating to semantic classes. Additionally, the false positive rate (percentage of positives sampled using cluster assignments which do not have the same ground-truth label as the anchor) is decreasing during training, which indicates that the sampled positives get better as the clustering quality improves. On the other hand, the false negative rate (negatives with the same ground-truth label as the anchor) increases as training progresses, indicating that some videos with the same semantic label

Table 2. **Linear probing and end-to-end finetuning top-1 accuracy results for action classification** pretrained on UCF101 and Kinetics400, then fine-tuned on UCF101 and HMDB51, using only visual inputs. Methods with * indicate that results were averaged across the 3 splits of each dataset instead of only using split-1. Subscripts of the input sizes indicate the skip rates each method uses. "None (Rand. Init.)" is trained using a supervised cross-entropy loss with a randomly initialized backbone. Methods with † use optical flow or residual frames in addition to RGB as inputs to the action classification model. SLIC$^{\hat{}}$ indicates that pretraining used 16 frame inputs. Dashes indicate unavailable information. "Supervised SLIC" is trained using a supervised triplet loss with positives and negatives sampled using semantic labels. "Cross-Ent." is the state-of-the-art finetuning performance after supervised pretraining (cross-entropy loss) on Kinetics400.

Pretrain. Method	Input Size	Arch.	UCF101	HMDB51
None (Rand. Init.)	16×128^2	R3D-18	53.5	22.3
Linear Probe (Pretrained on UCF101)				
TCLR [15]	$16_{\times 2} \times 112^2$	R3D-18	69.9	-
CoCLR-RGB [28]	32×128^2	S3D-23	70.2	39.1
CoCLR† [28]	32×128^2	S3D-23	72.1	40.2
SLIC	32×128^2	S3D-23	74.8	44.7
SLIC	16×128^2	R3D-18	72.3	41.8
SLIC	32×128^2	R3D-18	**77.7**	**48.3**
Supervised SLIC	16×128^2	R3D-18	76.4	44.9
Supervised SLIC	32×128^2	R3D-18	82.3	50.6
End-to-end Finetuning (Pretrained on UCF101)				
DPC [26]	40×128^2	R3D-18	60.6	-
VCP* [38]	16×112^2	R3D-18	66.0	31.5
VCOP* [59]	16×112^2	R3D-18	64.9	29.5
IIC [51]	16×112^2	R3D-18	61.6	-
IIC†* [51]	16×112^2	R3D-18	74.4	38.3
MemDPC [27]	40×128^2	R3D-18	68.2	-
CoCLR-RGB [28]	32×128^2	S3D-23	81.4	52.1
CoCLR† [28]	32×128^2	S3D-23	**87.3**	**58.7**
TCLR* [15]	$16_{\times 2} \times 112^2$	R3D-18	82.4	52.9
CoCon-RGB* [44]	$(-) \times 224^2$	R3D-18	70.5	38.4
CoCon-E*† [44]	$(-) \times 224^2$	R3D-18	82.4	52.0
SLIC	32×128^2	S3D-23	81.3	**56.2**
SLIC	16×128^2	R3D-18	77.4	46.2
SLIC	32×128^2	R3D-18	**83.2**	54.5
Supervised SLIC	16×128^2	R3D-18	77.4	46.6
Supervised SLIC	32×128^2	R3D-18	81.9	56.1
End-to-end Finetuning (Pretrained on Kinetics400)				
DPC [26]	40×128^2	R3D-18	68.2	34.5
CoCon-RGB* [44]	$(-) \times 224^2$	R3D-18	71.6	46.0
CoCon-E*† [44]	$(-) \times 224^2$	R3D-18	78.1	52.0
VideoMoCo [41]	16×112^2	R3D-18	74.1	43.6
RSPNet [12]	16×112^2	R3D-18	74.3	41.8
SpeedNet [6]	16×224^2	S3D-G	81.1	48.8
TCLR* [15]	$16_{\times 2} \times 112^2$	R3D-18	84.1	53.6
CoCLR-RGB [28]	32×128^2	S3D-23	**87.9**	**54.6**
CoCLR† [28]	32×128^2	S3D-23	90.6	62.9
SLIC$^{\hat{}}$	32×128^2	S3D-23	82.3	49.4
SLIC$^{\hat{}}$	32×128^2	R3D-18	83.2	52.2
Cross-Ent. [52]	16×112^2	R(2+1)D-34	96.8	74.5

are being grouped in different clusters. The inability to detect these false positive and negatives is one of the current limitations of our method.

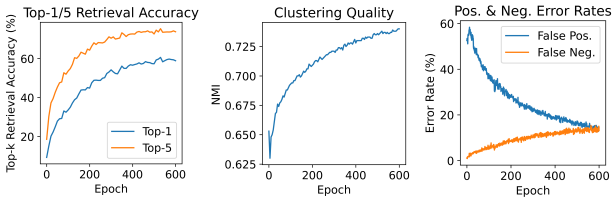

Figure 3. **Retrieval accuracy, clustering quality, and false sampling rates** for the experiment settings described in section 4.1 over 600 epochs on UCF101. (left): evolution of the top-1/5 retrieval accuracy; (middle): evolution of the clustering quality as measured by the NMI between the cluster assignments and the ground-truth labels; (right): percentage of positives and negatives sampled from clusters which are erroneous according to ground truth labels.

4.5. Ablation Studies

Ablation study on major training components. In order to study the impact of the main training components used in SLIC, we test the top-1/5 retrieval performance of R3D-18 models pretrained on UCF101 with a subset of the components. The results from removing iterative clustering, multi-view positive sampling, and the temporal discrimination loss from the self-supervised pretraining are shown in Table 3. With all-else fixed, the addition of iterative clustering results in a significant 21.7% gain (from 45.0%) in R@1 on UCF101 and a 5.8% gain (from 19.5%) in R@1 on HMDB51. The next-best impact on UCF101 R@1 performance comes from mult-view positive sampling (59.2% without it). However, the largest R@1 performance gain on HMDB51 comes from the temporal discrimination loss (19.1% without it), possibly indicating that HMDB51 benefits more from temporal attention than UCF101.

Table 3. **Ablation study on the impact of 3 major training components**: (1) iterative clustering (IC), (2) multi-view positive replacement using optical flow (MV), and (3) temporal discrimination loss (TL).

				UCF101		HMDB51	
IC	MV	TL	Input Size	R@1	R@5	R@1	R@5
✗	✓	✓	16×128^2	45.0	62.3	19.5	45.1
✓	✗	✗	16×128^2	54.7	65.6	18.2	41.5
✓	✓	✗	16×128^2	59.9	69.8	19.1	41.1
✓	✗	✓	16×128^2	59.2	69.8	20.1	43.6
✓	✓	✓	16×128^2	**66.7**	**77.3**	**25.3**	**49.8**

Ablation studies on clustering settings. We perform ablation studies to measure the impact of the clustering settings used during the self-supervised pretraining. In particular, we try clustering with FINCH partition-2, K-means, and Spherical K-means as alternatives to FINCH partition-1, and we measure the impact of the cluster interval (k) as well as the number of clusters in the case of K-means. All experiments

are performed with 16×128^2 input size. As shown in table 4, K-means and Spherical K-means result in comparable performance to FINCH partition-1 (which results in approximately 2200 clusters when clustering UCF101 video embeddings) when at least 1000 clusters are used. However, FINCH is significantly more efficient than K-means (roughly 85x faster) when clustering embeddings from the larger Kinetics400 dataset. Furthermore, the performance of K-means is quite sensitive to the number of clusters. For instance, using only 100 clusters with K-means results in a significant degradation in retrieval accuracy. The hyper-parameter search would pose great challenge when deploying K-means in practice. Hence, FINCH is a more suitable option for this application since it doesn't require additional hyper-parameter tuning. In addition, the performance of FINCH partition-2 (which results in approximately 550 clusters) is lower than that of the first FINCH partition. Thus, it can be hypothesized that over-clustering, or using many clusters with high purity, is beneficial to positive sampling. Additionally, cluster intervals of 1, 2, 5, and 10 result in similar video retrieval performance. Thus, it is not necessary to re-compute all training set embeddings and cluster them every epoch to achieve the best performance.

Table 4. **Ablation study on the impact of clustering settings** on video retrieval (after pretraining on UCF101). 'Num.' indicates the number of clusters and 'k' indicates the clustering interval.

			UCF101		HMDB51	
Clustering Method	Num.	k	R@1	R@5	R@1	R@5
FINCH(partition 1)	-	1	66.8	77.9	24.5	49.0
FINCH(partition 1)	-	2	64.8	76.7	24.2	46.6
FINCH(partition 1)	-	**5**	**66.7**	**77.3**	**25.3**	**49.8**
FINCH(partition 1)	-	10	66.7	77.8	22.4	48.7
FINCH(partition 2)	-	5	62.2	76.1	21.0	44.7
K-means	100	5	51.5	66.4	18.7	43.0
K-means	500	5	63.5	76.2	22.0	47.4
K-means	1000	5	66.0	76.4	21.8	46.4
K-means	2000	5	66.4	77.3	23.2	47.9
Spherical K-means	1000	5	64.9	77.3	23.6	48.2

Ablation study on positive sampling settings. We perform ablation studies to measure the impact of different positive sampling settings on the nearest neighbour retrieval tasks. p_α determines the probability of sampling positives from the same video instance (as opposed to different instances within the same cluster), and p_β determines the probability of using RGB inputs for positives (as opposed to replacing them with the optical flow view). The retrieval results from varying p_α and p_β are shown in Table 5. All experiments are performed with 16×128^2 input size. These results indicate that it is beneficial to sample most, but not all, of the positive clips using clustering assignments. Further, increasing p_β results in improved performance. The best performance is achieved with $p_\alpha = 0.2$ and $p_\beta = 0.75$.

Table 5. **Ablation study on the impact of p_α and p_β** on video retrieval (after pretraining on UCF101). The ablations on p_α are done using RGB views only.

			UCF101		HMDB51	
Pos. View	p_α	p_β	R@1	R@5	R@1	R@5
RGB	0.0	-	54.8	67.1	16.3	36.9
RGB	0.2	-	59.2	69.8	20.1	43.6
RGB	0.5	-	52.3	67.5	15.7	36.5
RGB	0.7	-	52.8	67.6	16.2	36.2
Optical flow	0.2	0.25	59.6	74.3	21.2	45.7
Optical flow	0.2	0.5	62.8	76.2	21.3	46.9
Optical flow	**0.2**	**0.75**	**66.7**	**77.3**	**25.3**	**49.8**

5. Limitations and Future Work

One of the main limitations of our method is that it is heavily reliant on the false positive and false negative rate. This is mainly an issue for the former since it was observed in Figure 3 that the false positive rate starts quite high at 60%, but is reduced due to iterative clustering. This high false positive rate could also potentially be reduced by obtaining multiple clustering assignments (e.g., from multiple clustering algorithms or clustering embeddings from multiple input views) and prioritizing positives which are in agreement across all cluster assignment sets. Additionally, it was observed in Figure 3 that the false negative rate was rising during the training, which indicates that some semantically similar videos are considered as dissimilar pairs when training the model. This could be detrimental to the final performance. We plan to address this issue by continuing to sample positives from one of the lower FINCH partitions but instead sample negatives from a higher partition. Since the higher FINCH partitions capture the data structure at a coarser granularity, it would reduce the likelihood of sampling false negatives when sampling from a higher partition.

6. Conclusions and Societal Impact

We propose a self-supervised, iterative clustering based contrastive learning framework to learn useful video representations. We showed that we can sample harder positives through clustering, and achieve further improvement by incorporating multi-view positives and a temporal discrimination loss. Our proposed method, SLIC, achieves state-of-the-art performance across various downstream video understanding tasks. Finally, while there is both promise and need for self-supervised representations to enable better activity recognition, video retrieval, and other downstream tasks, there exist potential uses of this technology (e.g. surveillance) that could exacerbate societal problems.

Acknowledgment This work is supported by the Vector Scholarship in Artificial Intelligence, provided through the Vector Institute.

References

[1] Hassan Akbari, Linagzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text, 2021. 3

[2] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering, 2020. 3

[3] Humam Alwassel, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *CoRR*, abs/1911.12667, 2019. 3

[4] Relja Arandjelović and Andrew Zisserman. Look, listen and learn. *arXiv:1705.08168*, 2017. 3

[5] Yuki Markus Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. *CoRR*, abs/2006.13662, 2020. 3

[6] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 7

[7] Tiffany Tianhui Cai, Jonathan Frankle, David J. Schwab, and Ari S. Morcos. Are all negatives created equal in contrastive instance discrimination?, 2020. 4

[8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018. 2

[9] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. *arXiv:1905.01278*, 2019. 2

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, volume 33, pages 9912–9924. Curran Associates, Inc., 2020. 2

[11] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017. 5

[12] Peihao Chen, Deng Huang, Dongliang He, Xiang Long, Runhao Zeng, Shilei Wen, Mingkui Tan, and Chuang Gan. Rspnet: Relative speed perception for unsupervised video representation learning. In *The AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 7

[13] 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, 2, 5, 12

[14] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[15] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. *arXiv:2101.07974*, 2021. 4, 5, 6, 7, 13

[16] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. Dynamonet: Dynamic action and motion network. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6191–6200, 2019. 3

[17] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[18] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2070–2079, 2017. 2

[19] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996. 3

[20] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[21] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition, 2019. 3

[22] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[23] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. 2

[24] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. 1, 2

[25] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, *International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. 2, 13

[26] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. *CoRR*, abs/1909.04656, 2019. 2, 7

[27] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *European Conference on Computer Vision*, 2020. 5, 6, 7

[28] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*,

volume 33, pages 5679–5690. Curran Associates, Inc., 2020. 1, 2, 3, 4, 5, 6, 7, 12, 13

[29] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?, 2018. 5, 12

[30] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Towards good practice for action recognition with spatiotemporal 3d convolutions. In *24th International Conference on Pattern Recognition (ICPR)*, pages 2516–2521, 2018. 3

[31] R. Devon Hjelm and Philip Bachman. Representation learning with video deep infomax. *CoRR*, abs/2007.13278, 2020. 2

[32] Yen-Chang Hsu and Zsolt Kira. Neural network-based clustering using pairwise constraints. *arXiv:1511.06321*, 2016. 2

[33] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv:1705.06950*, 2017. 5

[34] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020. 1

[35] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 7774–7785, 2018. 3

[36] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563, 2011. 5

[37] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C.H. Hoi. Prototypical contrastive learning of unsupervised representations. *ICLR*, 2021. 2

[38] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning, 2020. 5, 6, 7

[39] Vivek Sharma M. Saquib Sarfraz and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2019. 3

[40] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 2

[41] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. *CoRR*, abs/2103.05905, 2021. 2, 7

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

[43] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *arXiv:2008.03800*, 2021. 4

[44] Nishant Rai, Ehsan Adeli, Kuan-Hui Lee, Adrien Gaidon, and Juan Carlos Niebles. Cocon: Cooperative-contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3384–3393, June 2021. 2, 3, 7, 13

[45] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015. 2

[46] Pierre Sermanet, Corey Lynch, Jasmine Hsu, and Sergey Levine. Time-contrastive networks: Self-supervised learning from multi-view observation. *CoRR*, abs/1704.06888, 2017. 2

[47] Vivek Sharma, Makarand Tapaswi, M. Saquib Sarfraz, and Rainer Stiefelhagen. Clustering based contrastive learning for improving face representations. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 109–116, 2020. 2

[48] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, page 568–576, Cambridge, MA, USA, 2014. MIT Press. 3

[49] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 2

[50] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012. 5

[51] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Self-supervised video representation learning using inter-intra contrastive framework. *Proceedings of the 28th ACM International Conference on Multimedia*, Oct 2020. 4, 6, 7

[52] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *arXiv:1711.11248*, 2018. 2, 3, 7

[53] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2019. 2

[54] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. *arXiv:1806.09594*, 2018. 2

[55] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Wei Liu, and Yunhui Liu. Self-supervised video representation learning by uncovering spatio-temporal statistics. *CoRR*, abs/2008.13426, 2020. 3

[56] Jiangliu Wang, Jianbo Jiao, and Yunhui Liu. Self-supervised video representation learning by pace prediction. In *European Conference on Computer Vision*, 2020. 2

[57] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Computer Vision and Pattern Recognition (CVPR)*, pages 8052–8060, 2018. 2

[58] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature

learning: Speed-accuracy trade-offs in video classification. *arXiv:1712.04851*, 2018. 1, 2, 3, 5

[59] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Computer Vision and Pattern Recognition (CVPR)*, June 2019. 5, 6, 7

[60] Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Mahajan. Clusterfit: Improving generalization of visual representations. *arXiv:1912.03330*, 2019. 2

[61] Ting Yao, Yiheng Zhang, Zhaofan Qiu, Yingwei Pan, and Tao Mei. Seco: Exploring sequence supervision for unsupervised representation learning. In *35th AAAI Conference on Artificial Intelligence*, 2021. 2

[62] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. *arXiv:2006.11476*, 2020. 2

[63] Asano Y.M., Rupprecht C., and Vedaldi A. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2020. 2

[64] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. In *Proceedings of the 29th DAGM Conference on Pattern Recognition*, page 214–223, Berlin, Heidelberg, 2007. Springer-Verlag. 5

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

[66] Jiaojiao Zhao and Cees G. M. Snoek. Dance with flow: Two-in-one stream action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3