CycleCL: Self-supervised Learning for Periodic Videos

Matteo Destro
Cerrion, Inc.
matteo@cerrion.com

Michael Gygli
Cerrion, Inc.
michael@cerrion.com

Abstract

Analyzing periodic video sequences is a key topic in applications such as automatic production systems, remote sensing, medical applications, or physical training. An example is counting repetitions of a physical exercise. Due to the distinct characteristics of periodic data, self-supervised methods designed for standard image datasets do not capture changes relevant to the progression of the cycle and fail to ignore unrelated noise. They thus do not work well on periodic data.

In this paper, we propose CycleCL, a self-supervised learning method specifically designed to work with periodic data. We start from the insight that a good visual representation for periodic data should be sensitive to the phase of a cycle, but be invariant to the exact repetition, i.e. *it should generate identical representations for a specific phase throughout all repetitions. We exploit the repetitions in videos to design a novel contrastive learning method based on a triplet loss that optimizes for these desired properties. Our method uses pre-trained features to sample pairs of frames from approximately the same phase and negative pairs of frames from different phases. Then, we iterate between optimizing a feature encoder and re-sampling triplets, until convergence.*

By optimizing a model this way, we are able to learn features that have the mentioned desired properties. We evaluate CycleCL on an industrial and multiple human actions datasets, where it significantly outperforms previous video-based self-supervised learning methods on all tasks.

1. Introduction

When analyzing temporal data, repetitions are omnipresent in most types of applications: repetitions of physical exercises [15]; discrete sets of repeating steps in automatic production processes [3]; seasonal changes in remote sensing [44]; periodic patterns in medical applications, where they play a key role when analyzing vital signs such as heart rate or respiration [1, 27].

Thus, feature representations that capture the main aspects of periodic data are of key interest. In most cases, such data is very different from what can be found in standard datasets like ImageNet [11] or the Kinetics dataset [8], which are visually and semantically extremely diverse. Instead, in periodic data diversity is limited, and small differences are often fundamental. For instance, satellite images might always capture the same region with the goal of finding when seasonal patterns deviate from normality, *e.g.* during a drought. For models to perform well on such data, they need to be able to detect subtle changes. This limits the success of the de-facto standard of employing transfer learning from existing datasets [12, 29, 35, 47]. At the same time, labeled data for these applications is typically scarce and hard to acquire. Thus, in this paper we propose to employ self-supervision to learn strong representations for periodic videos. These features can then be used for a set of downstream tasks such as repetition counting [15] or anomaly detection. Focusing on feature learning is motivated by works such as [37], which showed that with sufficiently general representations, a simple model like k-nearest neighbors achieves strong performance on tasks such as anomaly detection.

A plethora of self-supervised methods have been proposed to tackle general feature learning, *e.g.* [9, 10, 14, 16, 18, 19, 30]. These methods, however, are typically designed for datasets such as ImageNet and struggle with periodic data. The prevalent contrastive methods [30], such as SimCLR [9], use random augmentations of the anchor as positive samples, and randomly sampled images as negative samples. Therefore, the model is trained to be invariant to pre-defined transformations of the *same* instance. This approach works well on highly diverse datasets, but becomes less effective on periodic data, as evidenced in Sec. 4.1, where a nearly identical image may be repeated once every cycle. In such cases, randomly selecting negative samples and forcing the representation to be dissimilar impedes the learning process, making a smart selection mechanism necessary.

We start from the following insight: a good visual representation for periodic data should be sensitive to the pro-
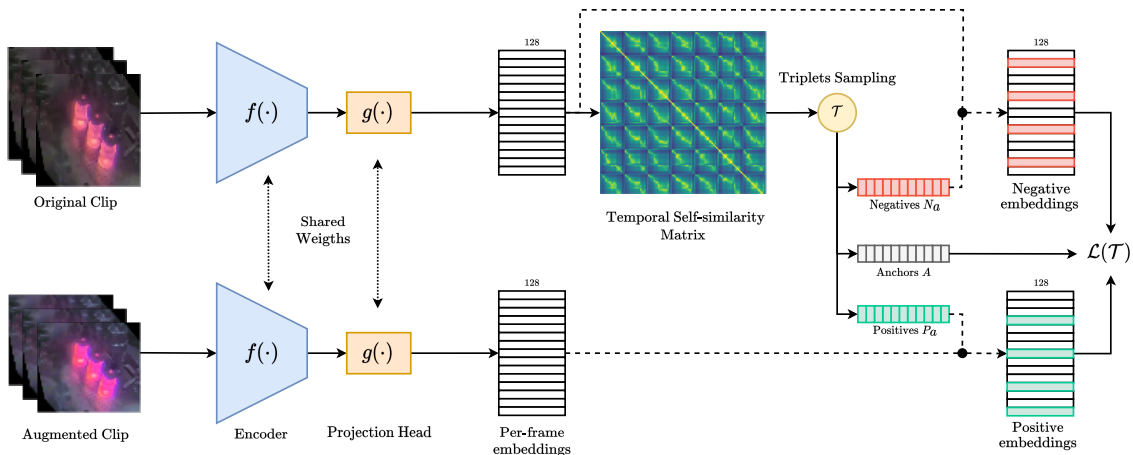
Figure 1. **Overview of our SSL method for Periodic Videos:** frames are encoded with a CNN and used to build a Temporal Self-Similarity Matrix (TSM) between the frames in a clip. From this matrix, we sample positive and negative frames for each anchor by selecting the most similar or dissimilar frames to build triplets. We then minimize a triplet loss, thereby pulling positives closer to the anchor and pushing negatives away. We iterate this process until convergence. To make the model more robust, we augment the positives with a set of random transformations and use them as inputs when computing and minimizing the loss.

gression/phase in the cycle. At the same time, it should be invariant to the exact repetition, *i.e.* produce the same representation for the same phase across cycles. We exploit this property to design our novel self-supervised learning method with smart triplet sampling, which is trained via a triplet loss [40]. Specifically, we hypothesize that even with a suboptimal feature representation it is possible to find frames that are clearly showing a different phase of the cycle. At the same time, it should be possible, with better than random accuracy, to find pairs of frames showing roughly the same phase. We use this idea to compute similarities between frames and, from these, sample initial positive and negative pairs, and start the optimization process. Our method iterates between computing new triplets and optimizing the encoder, until convergence. This is similar to deep clustering [6], which alternates between assigning images to clusters and optimizing the feature representation by predicting the cluster index of an image. In our experiments, we show that by optimizing the feature representation this way, we are able to learn frame embeddings that have the desired properties described above (see Fig. 2).

Concretely, we make the following contributions: (i) A novel self-supervised learning method designed for periodic video data and applicable over a wide range of domains. (ii) An augmentation strategy adapted to periodic data, further boosting the performance of our method. (iii) Evaluation on an industrial and on three human actions dataset over two tasks. Our approach outperforms previous self-supervised learning methods such as SimCLR [9], RepNet [15], Temporal Cycle Consistency [14] and DINO [7] on nearest neighbor classification and unsupervised anomaly detection. On the indutrial dataset, for example, our method re-

duces the gap to a fully supervised method to 8%, from 18% for the second best method (TCC). The remainder of the paper is structured as follows: in Sec. 2 we discuss related work. Sec. 3 introduces our method. Sec. 4 experimentally validates the benefits of our approach. Finally, Sec. 5 concludes our paper.

2. Related Work

Our method relates to existing works for self-supervised representation learning, in particular to methods that learn representations from videos via temporal signals.

Self-supervised learning (SSL). This is an effective way for pre-training deep models without the need for human supervision. SSL is based on the idea that a model can learn general visual representations by solving a *pretext task*. Such a task uses pseudo-labels extracted programmatically from the data itself, rather than human-provided labels. Through using a representation learned this way, little to no supervision is necessary to solve a *downstream task*, such as anomaly detection or image classification. An early canonical example of SSL is based on rotating images and predicting their rotation angle [16]. Recent methods use contrastive learning [9, 10, 18, 21], a form of metric learning, where positive pairs are created based on augmentations of an image itself. [43] proposes an extension to contrastive methods, tailored to diverse and heavy-tailed image datasets. Via clustering the dataset into semantically-similar subsets, they create relevant hard negatives by sampling within these clusters. When learning on repetitive videos we encounter the opposite problem, making this method unsuitable for periodic data (see Sec. 1)
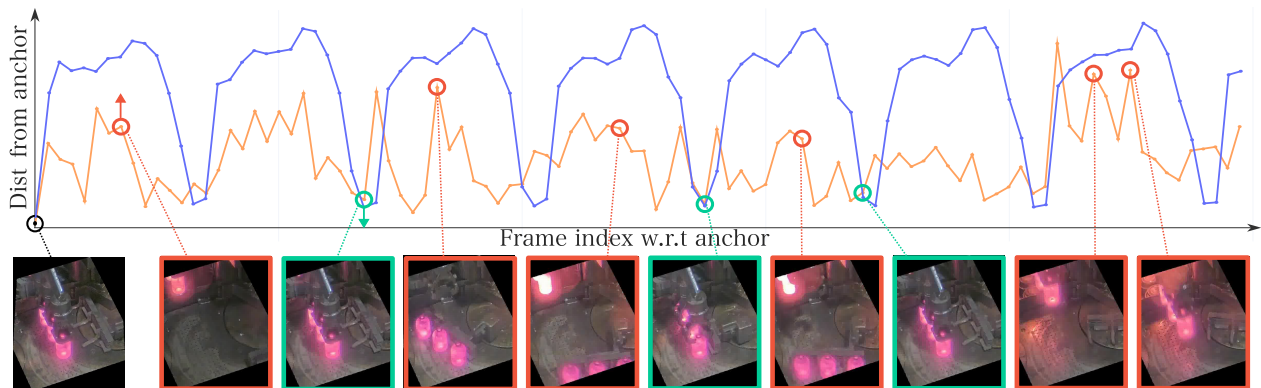
Figure 2. **Our triplet selection mechanism:** the feature similarity between consecutive frames is exploited to estimate a set of triplets such that the positives (green) belong to the same phase as the anchor (black) but in successive repetitions, while the negatives (red) belong to different phases. At the beginning of the training process the similarity (orange line) is noisy and the selection is inaccurate, but after just five epochs (blue line) the model learns representations that are sensitive to the phase but invariant to the specific repetition.

An alternative to contrastive approaches is DINO [7], a self-supervised method for vision transformers [13], and its extensions [31, 34]. DINO is a form of knowledge distillation with no labels, where the student is trained to predict the same features as the teacher, whose weights are the moving average of the student's weights, rather than directly optimized weights. Recently, masked autoencoders (MAE) [20] have been proposed for feature learning. They also rely on vision transformers [13], but instead aim to reconstruct a masked input. While generally performing well, MAE models are worse in anomaly detection tasks than previous self-supervised methods, as shown by Reiss *et al.* [38].

Apart from these most relevant works, a plethora of other SSL methods exist. For a more in-depth review of self-supervised learning methods, please refer to [2, 24, 28].

Learning representations from videos. Several methods rely on the temporal dimension of videos to design self-supervised objectives [14,15,19,23,25,33]. Temporal Cycle Consistency (TCC) [14] proposes an objective over pairs of videos. It optimizes the representation such that if a frame b in one video is the nearest neighbor for frame a in another video, the inverse holds as well. [26] extends [14] by additionally maximizing the similarity between frame and video representations (which are average pooled frame representations). Instead of using separate local and global objectives, [19] learns a representation by globally aligning video pairs using a differentiable version of Dynamic Time Warping [4]. Both methods however require class labels for each video to build pairs and struggle with periodic data, since the alignment becomes ambiguous.

More related, [15,23,46] also learn from periodic videos, specifically to perform repetition counting. SimPer [46] is based on contrastive prediction similar to SimCLR [9]. Instead, RepNet [15] is trained through creating synthetic rep-

etitions and using the period length as a supervision signal. Jacquelin *et al.* [23] sample triplets from periodic videos to optimize a metric learning objective, as in our work. However, they use a fixed sampling scheme relying on adjacent frames. Our method instead dynamically samples frames across different repetitions within the same video, thus producing more general features. This leads to more powerful representations, as they are invariant to the exact repetition of a periodic input, while increasing the sensitivity to the progression in a period.

In our experiments (Sec. 4) we compare our approach to [7, 9, 14, 15, 23], covering multiple canonical categories of methods designed to address this problem. Our findings show that CycleCL outperforms these alternatives on periodic video datasets.

3. Our method: CycleCL

We introduce CycleCL, a self-supervised learning method for learning feature representations on periodic data (Fig. 1). Periodic video or image sequences have a set of key characteristics: (i) They exhibit a repetitive signal. (ii) Variations during a repetition might be visually subtle, *e.g.* when alternatingly lifting a foot during a planking exercise. (iii) Variations irrelevant to the repetition might be prominent, *e.g.* when the camera is not static, there are camera flashes or partial occlusions of the repeating pattern, like in Fig. 5. Therefore, an effective visual representation for periodic data should accurately capture the phase of a cycle while remaining unaffected between repetitions. In other words, it should yield the same representation for a given phase across all repetitions. Our novel self-supervised learning method exploits the periodicity in videos to directly optimize a feature encoder to have these properties. Our method iterates between:

(i) Sampling triplets: for each frame, finding other frames that represent the same phase but in a different repetition (positives), and frames that depict a different phase (negatives).

(ii) Optimizing the encoder $f(\cdot)$, such that the distance from the anchor to the positives becomes smaller compared to the distance from the anchor to the negatives.

By iterating this process, our method learns features that capture the variations that are relevant to the periodic signal and ignore other irrelevant changes.

To optimize the encoder we specifically minimize a *triplet loss* [40] during training:

$$\mathcal{L}(\mathcal{T}) = \sum_{(v_a, v_p, v_n) \in \mathcal{T}} \left[\|f(v_a) - f(v_p)\|_2^2 \right.$$

$$\left. - \|f(v_a) - f(v_n)\|_2^2 + \alpha \right]_+ \quad (1)$$

where $\mathcal{T}$ is the set of valid triplets (v_a, v_p, v_n). The margin parameter α guarantees that only (semi-)hard samples play a role in the training process. Triplets where the negative sample is farther from the anchor than the positive sample by more than α do not contribute to the loss. We select triplet loss as it operates on a specific set of positive and negative examples. In contrast, more recent loss functions, such as *InfoNCE* [30], use a single positive and treat all other samples in a batch as negatives, which leads to an incorrect labeling for periodic data.

Next, we discuss our triplet sampling strategy and network architecture in detail.

Triplet sampling. When training with the loss in Eq. (1), the key is sampling adequate triplets. The goal is to select frames as positives if they depict the same phase as the anchor and as negatives otherwise. Devoid of ground truth, this can not initially be done perfectly. But even with an unoptimized feature encoder, it is possible to find such samples with better than random accuracy [6]. We do this by sampling positive and negative pairs based on the similarity of their feature representations. Initially, these features are produced by a model trained for ImageNet classification and a randomly initialized head, but as the encoder $f(\cdot)$ gets updated in consecutive steps, the feature representations evolve and allow more accurate sampling.

Fig. 2 shows how our sampling strategy provides a relevant learning signal.

The prerequisite to sampling triplets is a temporal self-similarity matrix (TSM). Given a representation $f(v_i)$ for each frame v_i of the input video, we compute the self-similarity matrix $\mathcal{S}$ using the *negative squared Euclidean distance*. Since the input videos can be very long, and we are interested in modeling short-term cycles, S is computed

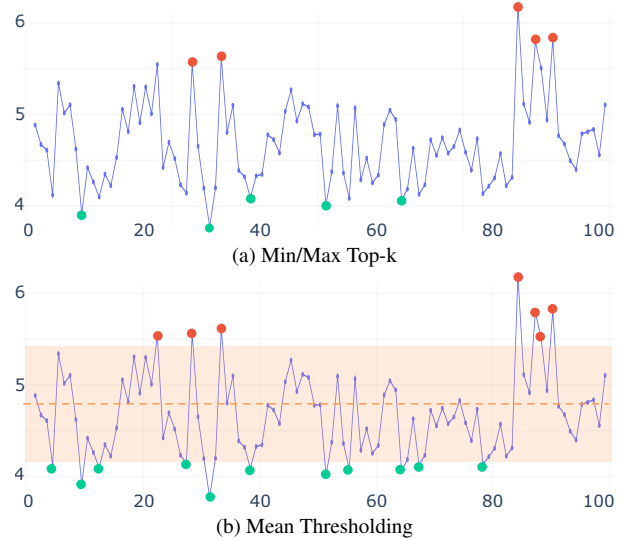(a) Min/Max Top-k

(b) Mean Thresholding

Figure 3. Visualization of the proposed triplets sampling methods. The green and red dots represent, respectively, the selected positive and negative samples for the current frame at index 0.

only on a *chunk* of the video, i.e. a window of C subsequent frames. The result is a matrix with shape $|C| \times |C|$, where $\mathcal{S}_{ij}$ is the similarity between the embeddings of frames v_i and v_j, computed as:

$$\mathcal{S}_{ij} = -\mathcal{D}_{ij} = -\|f(v_i) - f(v_j)\|_2^2 \quad (2)$$

An example of self-similarity matrix is shown in Fig. 4. We propose two different sampling strategies based on the similarity matrix $\mathcal{S}$:

(i) Min/Max Top-k: given an anchor frame v_a from a video chunk C, the set of positive frames P_a is sampled using the k *most similar* frames, *i.e.* $P_a = k\text{-argmax}_{v_p \in C} \mathcal{S}_{ap}$, while the negatives N_a are taken from the k *most dissimilar* ones.

(ii) Mean Thresholding: the mean similarity between the anchor v_a and all the frames in C is used as threshold for the selection, computed as: $\mu_a = \frac{1}{|C|} \sum_{v \in C} \mathcal{S}_{av}$, where $\mathcal{S}_{av}$ is the feature similarity between the anchor and video v. Frames with similarity below the threshold are taken as positives and above as negatives. An additional hyperparameter β controls the selection margin between the positives and the negatives, to ensure that the selected frames only have a strong similarity or dissimilarity. The final sets of positives and negatives are defined as:

$$P_a = \{v_p \in C \mid \mathcal{S}_{ap} > \mu_a(1 + \beta)\} \quad (3)$$
$$N_a = \{v_n \in C \mid \mathcal{S}_{an} < \mu_a(1 - \beta)\} \quad (4)$$

Fig. 3 presents a visual comparison of the selection methods. While the number of positives and negatives is a fixed

hyperparameter for *Min/Max Top-k*, their number is automatically set in *Mean thresholding* based on relative similarity, allowing it to select more accurate triplets.

Frame Encoder. The frame encoder $f(\cdot)$ is composed of two main components:

(i) Feature Extractor: each frame v_i is fed separately to a convolutional encoder to extract intermediate embeddings. The spatial dimensionality is kept. We used a ResNet model in our experiments, but this component is architecture-agnostic.

(ii) Projection Head: The convolutional features undergo dimension reduction through a 3D convolutional layer, global max pooling, and a linear layer. The 3D convolution helps the network to model short-term temporal patterns to better distinguish similar-looking frames but with different motion contexts.

Augmentations. We propose to augment the positive frames with random image transformations when optimizing the model. As the triplet loss (Eq. (1)) forces the positives to be closer to the anchor than the negatives by some margin, adding augmentations on the positives, but not the negatives, makes the task harder by forcing the network to become invariant to these transformations and thus makes it more robust [42]. The augmentations are used only when computing the loss. The triplets selection is performed on the unaltered frames to avoid changing the TSM and thus reducing the quality of the sampled triplets.

4. Experiments

In this section, we compare our approach with previous state-of-the-art methods. We arrange the evaluation in two steps, namely *feature learning* and *anomaly detection*, and focus on their results separately to get a better overview of the single contributions.

Datasets. We evaluate our method on four diverse datasets. The first one, which we refer to as *Industrial* dataset, consists of videos from production lines in the glass packaging industry, which we collected. This dataset consists of 575 unlabeled videos for training and 707 labeled videos for testing, with an average duration of 2 min. We collected these samples from cameras installed in production lines, and annotated anomalous events leading to loss of products.

The second is *Countix* [15], a subset of *Kinetics* [8] containing repetitive actions usually used as a benchmark for repetition counting tasks (examples in Fig. 5).

To ensure a comprehensive evaluation, we also include the *QUVA* [39] and *PERTUBE* [32] datasets in our evaluation. These datasets are closely related to Countix in terms of domain but are significantly smaller in size. QUVA is composed of 100 videos, while PERTUBE consists of only 50 videos. Given that, we follow [15] and use them as test

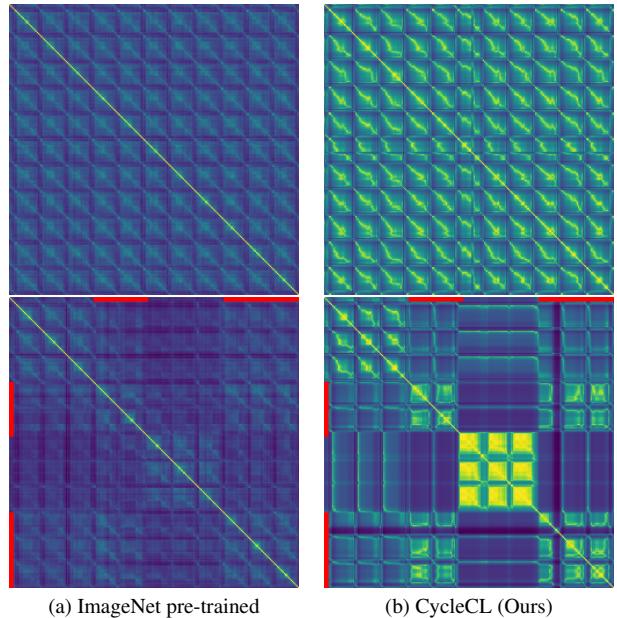(a) ImageNet pre-trained   (b) CycleCL (Ours)

Figure 4. Temporal Self-similarity Matrices (TSM) produced with a model trained on ImageNet and with CycleCL. A lighter color represents a higher similarity. In the video depicting a *periodic* process (top), the embeddings from the ImageNet baseline do not show strong similarity patterns at regular intervals. The model is not invariant to insignificant changes in the input. Our CycleCL model instead captures the periodicity accurately. For the *non-periodic* video (bottom), the TSM from CycleCL clearly shows a difference in similarity when an anomaly occurs (marked by the red lines), while the ImageNet model is not able to capture this change as clearly.

datasets on which we evaluate the models trained on Countix.

All the datasets have labeled intervals from two classes: *periodic* and *non-periodic*. For the Industrial dataset, the samples were manually annotated by an expert. For Countix, QUVA and PERTUBE the labels were provided by the authors.

Implementation details. We employ a ResNet-18 [22] model as encoder, pre-trained on ImageNet. The outputs of the last convolutional layer are stacked along the temporal dimension in chunks of 64 frames and forwarded to the projection head, followed by L_2 normalization. During training, we train to convergence, *i.e.* we stop the training process when the loss drops to zero for multiple consecutive iterations, which typically happens after around 40 epochs. We use the Adam optimizer, with a learning rate of 10^{-4}. We sample long video clips of 100 frames, to maximize the number of valid triplets at each iteration. More details in the supp. material.

Method	Industrial		Countix		QUVA		PERTUBE	
	AP	F_1	AP	F_1	AP	F_1	AP	F_1
Random	04.05	03.95	36.62	30.67	06.07	02.06	15.95	11.42
ImageNet pre-trained	46.25	46.31	40.23	37.18	10.30	10.12	27.89	27.05
SimCLR [9]	27.45	29.62	39.12	38.84	12.38	09.24	30.45	29.51
RepNet [15]	-	-	41.24	**43.22**	12.88	12.85	35.39	32.22
DINO [7]	50.65	50.92	38.24	37.40	12.47	12.03	33.45	32.59
TCC [14]	53.88	54.14	39.76	39.41	11.42	11.61	31.29	30.61
CycleCL (Ours)	**61.35**	**63.86**	**45.27**	43.16	**13.79**	**13.65**	36.98	**37.47**

Table 1. Comparison against baselines and current state-of-the-art approaches. Our CycleCL method significantly outperforms the ImageNet baseline and previous state-of-the-art methods. Notably, it significantly outperforms the canonical SimCLR [9] which struggles with periodicity, where nearly identical images are repeated once every cycle. In such cases, randomly selecting negative samples, as done in this method, is suboptimal. It also compares favorably with RepNet [15], which is specifically designed for periodic data.

4.1. Feature Quality

We first evaluate the robustness of the learned representations in terms of separation between *non-periodic* and *periodic* samples.

Evaluation protocol. We evaluate our method on the four datasets and compare it against previous approaches discussed in Sec. 2. In line with other recent works [7, 45], we use a weighted k-NN classifier on top of the features $f(x_i)$ to determine if a frame is taken from an interval depicting a periodic process or not. Since video data has a strong temporal redundancy between neighboring frames, we do leave-one-out evaluation, *i.e.* we exclude frames belonging to the same video of x_i when computing the nearest neighbors. The set N_k of top k nearest neighbors is determined using the Euclidean distance. The final classification is computed via weighted voting. Following the evaluation setting of Wu et al. [45], a class c gets a total weight $w_c = \sum_{i \in N_k} \alpha_i \mathbf{1}_{c_i=c}$, where $\alpha_i = \frac{1}{d_i^2 + \epsilon}$ and d_{xi} is the distance between the target sample x and sample i. We report F_1-*score*, *i.e.* the harmonic mean between recall and precision, as well as *Average Precision* (AP).

Compared methods. We compare against SimCLR [9], DINO [7], Temporal Cycle Consistency (TCC) [14], and RepNet [15], all state-of-the-art approaches for self-supervised learning. We further include a *random* baseline and a baseline with features of a ResNet-18 model trained on *ImageNet*.

For a fair comparison, all methods use the same 3D backbone as ours and were trained on the same data. The only exception is RepNet, where we use the original model trained on Countix, as released by the authors, since the training code was not made available. Moreover, since the Industrial dataset significantly differs from Countix, RepNet results for it are not included.

Results (Tab. 1). Our method outperforms the current state of the art on all four datasets on both metrics, except RepNet on Countix, where CycleCL outperforms RepNet in AP and

Figure 5. Example from Countix [15]. Even for frames showing the same phase of a bike spin, there is large variation due to position differences, camera zoom, illumination changes and a camera flash, highlighting the difficulty of learning relevant features.

is on par in terms of F_1. On the industrial dataset, the F_1 score is improved by 17% compared to the ImageNet pre-trained baseline, and by 13% compared to the DINO model.

In terms of previous methods, SimCLR does not perform well on Countix and on the Industrial dataset as it struggles with periodic data, as discussed in Sec. 1. Optimizing for *Temporal Cycle Consistency* [14] performs well on the Industrial dataset, but worse than SimCLR [9] and RepNet [15] on the other datasets, especially so on Countix. Countix depicts more diverse actions with larger visual variations, and TCC struggles with repetitions as aligning across videos becomes ambiguous in this case. RepNet's performance is only slightly worse than our method. But it also requires a more complex augmentation pipeline for generating synthetic videos compared to ours. To conclude, these strong results show that CycleCL is able to produce robust representations.

Apart from a comparison against other SSL methods, we also evaluated a *supervised* baseline on the industrial dataset. There, the features are extracted from a ResNet-18 model, trained with a supervised signal on the available annotations of the training split. This model achieved 70.81 of Average Precision and 71.91 of F_1-score. While the performance gap w.r.t. a supervised model persists, our method
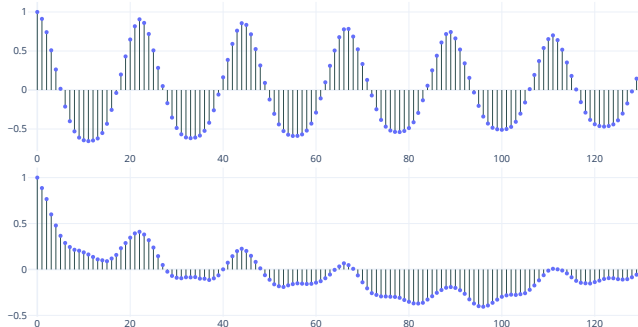
Figure 6. Autocorrelation plot of a periodic (top) and non-periodic (bottom) video. The autocorrelation of two periodic signals measures their correlation as they shift in time, revealing any repeating patterns between them. The cyclic nature of the input is clearly visible during a normal process but disappears during an anomaly.

reduces it to 8%, compared to more than 18% for all the other methods.

Visualizations. Fig. 4 shows an example of TSMs extracted from a periodic and non-periodic video. Our CycleCL model encodes the progression of the cycles better than the ImageNet baseline. Fig. 6 depicts two autocorrelation plots from the same videos. This visualization shows how good our features are: the period of the cycles can be easily determined by using simple techniques like Fourier transforms.

4.2. Ablation study

We perform a set of ablation studies on the Industrial dataset to justify some of the decisions we made when designing CycleCL and show the robustness of our method w.r.t. specific parameter values. Here, we focus on the key design choices (more ablation experiments are provided in the supp. material).

Triplets sampling strategy. We compare our sampling against other strategies found in the literature (Tab. 2). Our *Mean thresholding* approach with $\beta = 0.3$ outperforms all the other methods. Compared to the best-competing strategies proposed in [14, 23], its F_1 score is 9% higher.

Building triplets by using the nearest frame as positive and the second nearest as negative, as done in the *Adjacent* [23] method, is not effective on these datasets. Depending on the framerate, the next and the second next frames could have high similarity, and learning to separate them leads to the model focusing on less relevant local changes.

Min/Max Top-k sampling consistently chooses an equal amount k of positive and negative samples from each video clip, irrespective of how similar they are to the anchor. Therefore, it does not account for potential differences in the distribution of valid positives and negatives across vari-

Method	AP	F_1
Random features	04.05	03.95
ImageNet features	46.25	46.31
Adjacent [23]	41.32	41.65
TCC [14]	53.88	54.14
Min/Max Top-k (Ours)	53.76	57.54
Mean Thresholding (Ours)	**61.35**	**63.86**

Table 2. Ablation study on the triplets sampling strategy for the Industrial dataset. Our best sampling strategy (Mean Thresholding) outperforms the other strategies and baselines.

ous clips. In contrast, *Mean thresholding* does not rely on a fixed k value, but instead takes into account the similarities to determine the number of triplets to sample. This dynamic approach allows for adjustments based on the varying distributions and the relative balance of positive and negative samples across clips (e.g. a video tends to have more negative pairs than positives). As a result, it leads to a more effective and less noisy selection process.

Data augmentations. When minimizing the triplet loss (Eq. (1)), our method quickly learns to correctly separate matching from non-matching pairs and converges. As mentioned in Sec. 3, adding augmentations introduces harder samples and helps to produce more robust representations [36]. We test applying the following set of augmentations: brightness and contrast alterations, color jittering, Gaussian blurring, and random cropping. The triplets selection is performed on the original frames to avoid altering the similarity values and thus reducing the quality of the sampled triplets. Therefore the augmented versions are used only when computing the loss.

We analyze different strategies to apply these augmentations (Tab. 3): augmenting all samples, augmenting only the positives, or not using augmentations. We find that augmenting only the positives works best and it improves the F_1-score by 7.5% compared to not using augmentations. By augmenting only the positive samples and keeping the negatives unaltered, the model needs to learn to focus on the relevant information that encodes the current phase and ignore the changes introduced by the augmentations. Augmenting also the negatives makes the task again easier and performs worse than not using augmentations at all.

Sequence length. The length of the sequence for sampling triplets is an important hyperparameter. During training, using a longer video sequence can provide more valid positive and negative pairs and improve the training signal by including more complex cases. Thus, performance is generally higher for longer input sequences (see Tab. 4). But there is a trade-off between sampling from long sequences and a reasonable batch size, due to memory constraints of the GPU. We see a small performance drop for a sequence

Data Aug.	AP	F_1
All	50.19	51.98
Positives Only	61.35	63.86
None	55.64	56.39

Table 3. Ablation study on the data augmentations for the Industrial dataset. Augmenting only the positives outperforms the other approaches by a significant margin.

Seq. Length	Batch Size	AP	k-NN F_1
ImageNet	-	46.25	46.31
10	32	48.82	48.30
25	16	55.89	54.37
50	8	60.68	60.02
100	4	61.35	63.86
200	2	61.19	63.57

Table 4. Ablation study on the sequence length of sampled video clips, for the Industrial dataset.

Encoder	Features	AD Method	AP	F_1
ImageNet	Raw	LOF	12.8	19.8
ImageNet	Cycle	k-NN	26.2	39.8
AE	-	-	08.6	09.3
MemAE [17]	-	-	09.1	13.2
CycleCL	Raw	LOF	19.4	27.9
CycleCL	Cycle	k-NN	44.3	52.9

Table 5. Comparison of different feature representations and anomaly detection methods according to the frame average precision (AP) and oracle F_1 scores.

of 200 frames with a batch size of 2. We believe that this is because there are too few samples for Batch Normalization to work well.

4.3. Anomaly Detection

We now evaluate how well the learned features transfer to an anomaly detection task.

Evaluation protocol. We focus on the Industrial dataset and use the same ground truth as Sec. 4.1. Any interval annotated as non-periodic, which is rare, is considered as *anomaly*. Anomalies are classified fully unsupervised based on the feature distance to other frames, in contrast to the previous section, where frames were classified based on the labels of other videos. We use our frame embeddings as input to standard anomaly detection methods, namely k-*Nearest Neighbors*, *Local Outlier Factor* [5] and *One-Class SVM* [41]. Using the available annotations we then compute the precision-recall curve and the corresponding AP. We also report the *oracle F_1-score*, the maximum attainable F_1-score for the given anomaly ranking if the optimal classification threshold was chosen.

Feature representation. We use the learned features directly (*Raw*). We also propose a version where a frame is represented by its similarity to other frames in the clip (*Cycle*), *i.e.* by its row in the temporal similarity matrix, as $c(v_i) = \mathcal{S}_{i:}$.

Compared methods. As baseline, we apply the same techniques to features extracted from a model pre-trained on ImageNet, as done for the feature quality evaluation. We also compare our methods against an autoencoder (AE) and Mem-AE [17], a state-of-the-art approach for anomaly detection based on reconstruction error. Since these methods directly provide an anomaly score, no additional post-processing techniques are needed.

Results. Tab. 5 shows the results. Our model outperforms all compared methods by a significant margin: it exceeds the ImageNet baseline by 13% in F_1, even when using our cycle representation to improve it. The gap between the plain and memory-augmented autoencoders is even more extreme (13% F_1 for Mem-AE *vs.* 53% for our method). These methods do not perform well on this dataset, likely because the videos are visually rather static, and the high-

capacity models used are able to learn to reconstruct the input successfully even in the presence of anomalies. The cycle representation provides significant improvements in the anomaly detection task compared to the raw embeddings (with our method, from 28% F_1 for raw to 53% for cycle features). This is because they capture a larger temporal context at a lower dimensionality, thus making it easier to represent changes in temporal dynamics.

5. Conclusions

In this work, we proposed CycleCL, a self-supervised learning method for periodic data. We designed a novel contrastive learning method based on a triplet loss. It iterates between (i) sampling frames from approximately the same phase as an anchor and negative frames from different phases and (ii) optimizing a feature encoder to increase their separation, until convergence. We evaluated CycleCL on frame-based nearest-neighbour classification and unsupervised anomaly detection on a new Industrial dataset and on three human actions datasets. By achieving state-of-the-art results on such different domains, we demonstrate the effectiveness of our method on static and dynamic camera scenes, including both subtle and large visual differences. Our extensive evaluation against the previous state of the art such as SimCLR [9], RepNet [15], Temporal Cycle Consistency [14] and DINO [7] shows the benefits of our approach: it significantly outperforms all compared methods.

References

[1] Sean Bae, Silviu Borac, Yunus Emre, Jonathan Wang, Jiang Wu, Mehr Kashyap, Si-Hyuck Kang, Liwen Chen, Melissa Moran, Julie Cannon, et al. Prospective validation of smartphone-based heart rate and respiratory rate measurement algorithms. *Communications medicine*, 2022. 1

[2] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning. *arXiv*, 2023. 3

[3] Yaser Banad, Nariman Razaviarab, Hadi Fekrmandi, Guoqiang Li, Patrick Mensah, Shuju Bai, and Safura Sharifi. Automated quality and process control for additive manufacturing using deep convolutional neural networks. *Recent Progress in Materials*, 2021. 1

[4] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. *AAAIWS*, 1994. 3

[5] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: Identifying density-based local outliers. *ACM SIGMOD Record*, 2000. 8

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. *ECCV*, 2018. 2, 4

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *ICCV*, 2021. 2, 3, 6, 8

[8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 1, 5

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *ICML*, 2020. 1, 2, 3, 6, 8

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv*, 2020. 1, 2

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. *CVPR*, 2009. 1

[12] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *ICML*, 2014. 1

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR 2021*, 2021. 3

[14] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal Cycle-Consistency Learning. *CVPR*, 2019. 1, 2, 3, 6, 7, 8

[15] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Counting out time: Class agnostic video repetition counting in the wild. *CVPR*, 2020. 1, 2, 3, 5, 6, 8

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

[17] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton Van Den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. *ICCV*, 2019. 8

[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *NeurIPS*, 2020. 1, 2

[19] Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram Najam Syed, Andrey Konin, Muhammad Zeeshan Zia, and Quoc-Huy Tran. Learning by aligning videos in time. *CVPR*, 2021. 1, 3

[20] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 3

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016. 5

[23] Nicolas Jacquelin, Romain Vuillemot, and Stefan Duffner. Periodicity counting in videos with unsupervised learning of cyclic embeddings. *Pattern Recognition Letters*, 2022. 3, 7

[24] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *ICAI*, 2022. 3

[25] Simon Jenni, Givi Meishvili, and Paolo Favaro. Video representation learning by recognizing temporal transformations. *ECCV*, 2020. 3

[26] Quan Kong, Wenpeng Wei, Ziwei Deng, Tomoaki Yoshinaga, and Tomokazu Murakami. Cycle-contrast for self-supervised video representation learning. *NeurIPS*, 2020. 3

[27] Benjamin Kossack, Eric L Wisotzky, Anna Hilsmann, Peter Eisert, and Ronny Hänsch. Local blood flow analysis and visualization from rgb-video sequences. *Current Directions in Biomedical Engineering*, 2019. 1

[28] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *TKDE*, 2021. 3

[29] Thomas Mensink, Jasper Uijlings, Alina Kuznetsova, Michael Gygli, and Vittorio Ferrari. Factors of influence for transfer learning across diverse appearance domains and task types. *PAMI*, 2022. 1

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, 2018. 1, 4

[31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez,

Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv*, 2023. 3

[32] Costas Panagiotakis, Giorgos Karvounas, and Antonis Argyros. Unsupervised detection of periodic segments in videos. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 923–927, 2018. 5

[33] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *CVPR*, 2021. 3

[34] Kanchana Ranasinghe, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Michael S Ryoo. Self-supervised video transformer. In *CVPR*, 2022. 3

[35] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. *CVPR Workshop*, 2014. 1

[36] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A. Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Data augmentation can improve robustness. *NeurIPS*, 2021. 7

[37] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. PANDA: Adapting pretrained features for anomaly detection and segmentation. *CVPR*, 2021. 1

[38] Tal Reiss, Niv Cohen, Eliahu Horwitz, Ron Abutbul, and Yedid Hoshen. Anomaly detection requires better representations, 2022. 3

[39] Tom F H Runia, Cees G M Snoek, and Arnold W M Smeulders. Real-world repetition estimation by div, grad and curl. In *CVPR*, 2018. 5

[40] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. *CVPR*, 2015. 2, 4

[41] Bernhard Schölkopf, John Platt, John Shawe-Taylor, Alexander Smola, and Robert Williamson. Estimating support of a high-dimensional distribution. *Neural Computation*, 2001. 8

[42] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 2019. 5

[43] Yonglong Tian, Olivier J Henaff, and Aäron van den Oord. Divide and contrast: Self-supervised learning from uncurated data. In *CVPR*, 2021. 2

[44] Jan Verbesselt, Rob Hyndman, Glenn Newnham, and Darius Culvenor. Detecting trend and seasonal changes in satellite image time series. *Remote sensing of Environment*, 2010. 1

[45] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv*, 2018. 6

[46] Yuzhe Yang, Xin Liu, Jiang Wu, Silviu Borac, Dina Katabi, Ming-Zher Poh, and Daniel McDuff. SimPer: Simple Self-Supervised Learning of Periodic Targets. *arXiv*, 2022. 3

[47] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv*, 2019. 1