CycleCL: Self-supervised Learning for Periodic Videos Supplementary Material

Matteo Destro
Cerrion, Inc.
matteo@cerrion.com

Michael Gygli
Cerrion, Inc.
michael@cerrion.com

1. Supplementary Material

We provide additional implementation details, ablations and additional visualizations in the following sections.

A. Implementation Details

In addition to the pre-processing discussed in Sec. 4, each frame is resized to 224 x 224 and normalized before being fed to the network. The last convolutional layer produces a 7 x 7 x 512 feature maps for each frame, which are then stacked in chunks of 64 and forwarded to the projection head. We use 512 filters with $3 \times 3 \times 3$ kernels for the 3D convolution, followed by batch normalization and Leaky ReLU activations. To maintain the input dimensionality on the temporal dimension, we set the padding to 1. The spatial dimensions are collapsed with either an adaptive max or average pooling. After the final linear layer, the embeddings are L_2 normalized. For optimization, we use Adam [3], with a learning rate of 10^{-4} and weight decay of 10^{-3}. The triplet loss margin α is set to 0.5. When sampling the video clips, the temporal stride is set to 2 for the Industrial dataset and to 5 for the others. The experiments were run on an Nvidia Tesla T4 GPU.

B. Additional Experiments

This section discusses the additional experiments we performed to better understand the robustness of our method to varying hyperparameter choices.

Embeddings dimensionality. The dimensionality of the final output embedding can have a significant effect on the downstream task of nearest neighbor classification. In Tab. A we compare different dimensionalities and show that they do not have a big influence on the results: overall our method performs best across ranges from 128 to 256. There seems to be a slight trade-off though: smaller embeddings suffer less from the curse of dimensionality when performing k-NN classification w.r.t. larger embeddings, but at the same time they have lower capacity to encode all the relevant information.

Output dim.	**AP**	**k-NN F_1**
ImageNet	46.25	46.31
64	58.13	58.55
128	61.35	63.86
256	62.25	62.14

Table A. Ablation study on the output dimensionality, for the Industrial dataset.

Layers	**Pooling**	**AP**	**k-NN F_1**
ImageNet	-	46.25	46.31
2D Conv. + FC	Mean Pool	51.71	54.32
2D Conv. + FC	Max Pool	56.09	57.73
3D Conv. + FC	Mean Pool	56.46	56.98
3D Conv. + FC	Max Pool	61.35	63.86

Table B. Effect of the projection head on the performance, for the Industrial dataset.

Projection head. We compare different head architectures besides the one presented in Sec. 3. In particular, we evaluate the contributions given by the 3D convolutional and the spatial max pooling layers w.r.t. their 2D and average counterparts. Tab. B reports the results. They indicate that both, the type of pooling and convolutional layers, are important choices for maximizing performance. The 3D convolution helps the model to capture temporal context. The spatial max pooling allows the model to focus on a specific region of the input and ignore background regions that have no information regarding the cycle.

L_2-normalization. In the context of nearest neighbor classification, constraining the loss can help improve the quality of the embeddings [5]. In particular, triplet loss is known to be sensitive to the magnitude of the embeddings [4]. Thus, we test the contribution of the L_2 normalization bottleneck in stabilizing the training, reported in Tab. C. In line with other self-supervised frameworks [1, 2], we find that L_2-normalization leads to significantly better performance.

Data amount. While our method requires no annotation, reducing the amount of data needed for training can greatly reduce training time. Therefore, we also perform an ablation study on the amount of data used to train the model, reported in Tab. D. While using more data can be effective in increasing the robustness of the learned representations, the improvements are minimal, and even using only 10% of the data provides significant gains w.r.t. the ImageNet baseline.

C. Additional Visualizations

Temporal self-similarity matrix (TSM). Here, we analyze the TSM shown in Fig. 4 of the main paper in more detail. The TSM shows the similarity of each frame to all (other) frames, thus encoding the temporal similarity patterns. We visualize the TSM and the corresponding frames for the anomalous video in Fig. A. From this, it can be seen that our CycleCL features clearly capture the difference in the cyclic signal when an anomaly occurs. In particular, it detects the different types of anomalies that occur in sequence as distinct clusters, represented by the four lighter blocks along the diagonal.

The first cluster depicts the start of an anomaly. At the beginning of the second cluster, the anomaly becomes more severe, and only one bottle is still produced. A cyclic pattern is still visible in both cases, but as the anomalies are different, the similarity between clusters is lower. During the third cluster, the bottle production is stopped entirely, while the robotic arm is still moving: the similarity is therefore very high within the cluster because most of the periodic process is stopped.

Features projection. We apply PCA to the embeddings produced by the CycleCL model to map them to a 1-dimensional space, by taking the first principal component. Fig. B shows the result of this operation for a normal and an anomalous video clip. The PCA projection of the normal video is smooth and quasi-sinusoidal, clearly showing the temporal cycle of the video, while the projection of the anomalous video is discontinuous. This validates that our methods leads to features that are sensitive to the phase of the input.

Nearest neighbor distance. In the context of nearest neighbor classification, we are interested in understanding whether the model produces clearly separable embeddings for anomalous *vs.* normal frames. Fig. C shows an example of the nearest neighbor distances of an anomalous video w.r.t. a normal one. For each frame of the anomaly video, the distance to the nearest frame in the normal video is computed and used as its anomaly score. When a frame represents a normal state, a good match can be easily found in the normal video. On the contrary, when a frame represents an anomaly, the distance to the nearest neighbor is higher. This validates that the features are sensitive to deviations from normal repetitions.

Norm.	**AP**	k-**NN** F_1
ImageNet	46.25	46.31
L_2-norm	61.35	63.86
None	57.13	56.01

Table C. Effect of l_2-normalization after the projection head, for the Industrial dataset.

Data %	**AP**	k-**NN** F_1
ImageNet	46.25	46.31
10	59.09	59.71
50	60.27	61.93
100	61.35	63.86

Table D. Effect of the data quantity on the performance of our method, on the Industrial dataset.

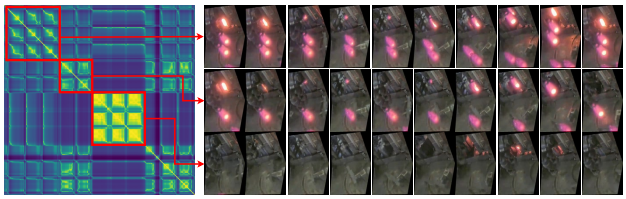

Figure A. Analysis of the repeating patterns appearing in the TSM, extracted from an abnormal video. Three different phases can be identified (marked in red): (i) initial small anomaly, (ii) more severe anomaly, (iii) section running empty without producing any bottles.

Figure B. 1D PCA projection of embeddings for a normal (left) and abnormal (right) video clip. The frames depicting the anomaly are represented in red.

Figure C. Nearest neighbor distances of abnormal frames (in red) and normal frames (in blue) with respect to a reference video depicting the normal process. While the process in the query video is running normally, correspondences with the reference can be found with a small distance. Instead, during an anomaly the visual difference is high and the distance to the nearest neighbor is a strong sign of abnormality.

References

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

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

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

[5] Kilian Q Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *NeurIPS*, 2005.