Supplementary Material for: Video Representation Learning by Recognizing Temporal Transformations

Simon Jenni[0000−0002−9472−0425], Givi Meishvili[0000−0002−0984−7078], and Paolo Favaro[0000−0003−3546−8247]

University of Bern, Switzerland
{simon.jenni,givi.meishvili,paolo.favaro}@inf.unibe.ch

1 Additional Implementation Details

We provide some additional implementation details regarding network architectures and data pre-processing. For our C3D network we followed the implementation of [2]. We list the layer definitions in Table 1. For the 3D-ResNet18 experiments we follow the implementation of [1]. Concretely, this means we use videos of resolution 128 × 128 and apply more aggressive data augmentation, *i.e.*, we add random resizing and color jittering.

Table 1: **The C3D network architecture.** The first dimension in the kernels and strides refers to the temporal dimension. **BN** indicates the use of batch normalization. We assume the use of SAME padding throughout.

Small C3D Network
conv3d: kernel=$3 \times 3 \times 3$, strides=(1, 1, 1), channels=64, activation=`ReLU`, **BN**
maxpool3d: kernel=$1 \times 2 \times 2$, strides=(1, 2, 2)
conv3d: kernel=$3 \times 3 \times 3$, strides=(1, 1, 1), channels=128, activation=`ReLU`, **BN**
maxpool3d: kernel=$2 \times 2 \times 2$, strides=(2, 2, 2)
conv3d: kernel=$3 \times 3 \times 3$, strides=(1, 1, 1), channels=256, activation=`ReLU`, **BN**
maxpool3d: kernel=$2 \times 2 \times 2$, strides=(2, 2, 2)
conv3d: kernel=$3 \times 3 \times 3$, strides=(1, 1, 1), channels=256, activation=`ReLU`, **BN**
maxpool3d: kernel=$2 \times 2 \times 2$, strides=(2, 2, 2)
conv3d: kernel=$3 \times 3 \times 3$, strides=(1, 1, 1), channels=256, activation=`ReLU`, **BN**
maxpool3d: kernel=$2 \times 2 \times 2$, strides=(2, 2, 2)
fully-connected: channels=2048, activation=`ReLU`, **BN**
fully-connected: channels=2048, activation=`ReLU`, **BN**
fully-connected: channels=$n_{classes}$

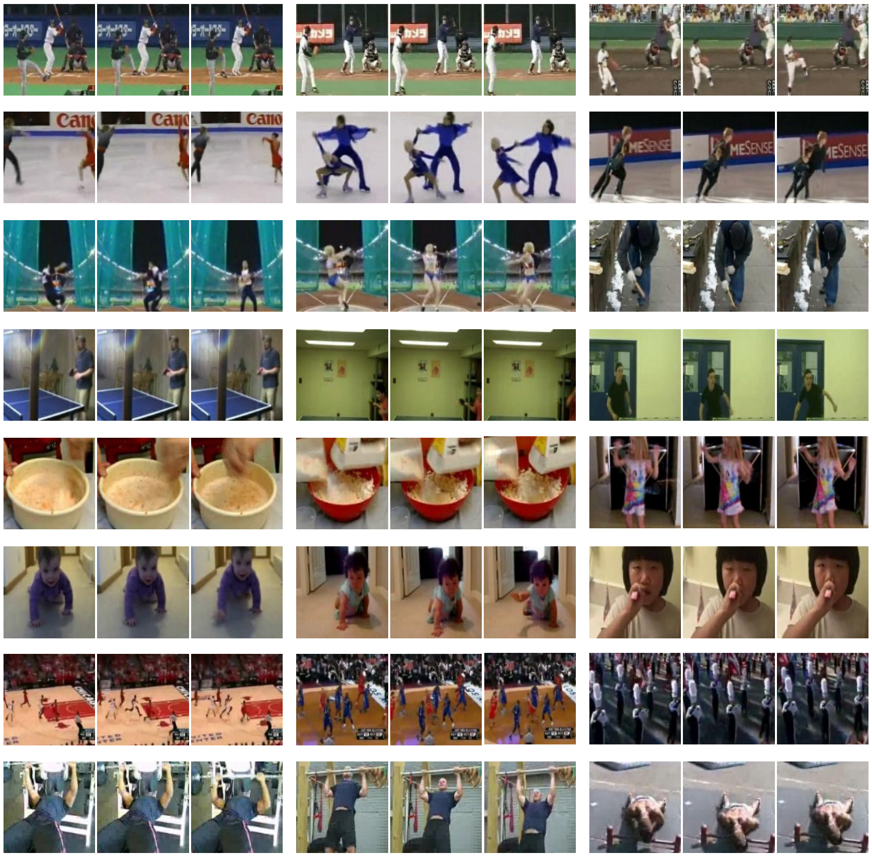

Fig. 1: **Examples of Retrievals in UCF101.** Leftmost block of 3 columns: Frames from the query sequences. Second and third blocks of 3 columns: Frames from the two nearest neighbors.

2 Qualitative Nearest-Neighbor Results

We show some examples of nearest neighbor retrievals in Fig. 1. Frames from the query test video are shown in the leftmost block of three columns. The second and third blocks of three columns show the top two nearest neighbors from the training set. We observe that the retrieved examples often capture the semantics of the query well. This is the case even when the action classes do not agree (*e.g.*, last row).

References

1. Han, T., Xie, W., Zisserman, A.: Video representation learning by dense predictive coding. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019)
2. Wang, J., Jiao, J., Bao, L., He, S., Liu, Y., Liu, W.: Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4006–4015 (2019)