Supplementary Materials: Self-Supervised Procedure Learning from Instructional Videos using DNNs

Anonymous ECCV submission

Paper ID 2830

Key-step Localization Network. To build the key-step localization network (KLN), we make connection between our goal, which is to take a T_ℓ input vectors and output T_ℓ outputs each of dimension M, and semantic image segmentation whose goal is to take an input image and produce an output image where each pixel takes one of few discrete values corresponding to a category. Thus, we take the network in [1] and, given that we are working with sequential data instead of 2D images, we convert 2D convolutions, 2D pooling and 2D deconvolutions to 1D temporal convolutions, 1D pooling and 1D deconvolutions, respectively, see Figure 1.

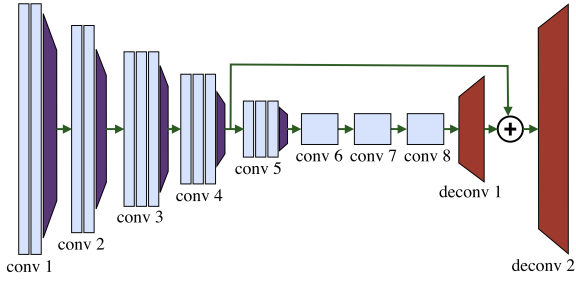

Fig. 1: Our key-step localization network (KLN) takes T_ℓ input vectors, corresponding to feature vectors from T_ℓ segments/frames of video ℓ, and generates T_ℓ outputs each of dimension M, where the t-th output encodes the assignment probabilities of segment/frame t to each of the M latent states obtained from videos.

As the figure shows, our KLN consists of multiple consecutive layers of 1D convolutions and 1D pooling, forming the encoding part of the network, followed by 1D deconvolutions, forming the decoding part of the network. The KDN is organized similar to [1]. The first five convolutional subnetworks (conv1 to conv5) each consist of multiple temporal convolutional layers, where each temporal convolution is followed by batch normalization and ReLU activation. Each convolutional subnetwork is followed by a temporal max-pooling. The subnetworks conv6 and conv7, each consists of a temporal convolution followed by ReLU and dropout. The subnetwork conv8 consists of 1×1 convolution with batch normalization. We apply deconvolution along the time axis on the output of conv8. We also apply a 1×1 convolution and batch normalization to the output of pool4 and add (element-wise) the result with deconv1 features. This skip connection, used in semantic segmentation to produce better visual features,

Fig. 2: Visualization of the self-supervised learned attention model on two videos from the task 'assemble clarinet' (left) and 'perform CPR' (right) from ProceL. Notice that our method successfully learns to focus on important region of each frame. For example, for clarinet, it focused on cork, ligature, screws, lower and upper joints in the associated key-steps.

is also useful in key-step discovery, as it helps to recover temporal information for key-step and video classification. Finally, we apply a temporal deconvolution and obtain the final predictions, which are T_ℓ outputs each of dimension M.

More Results from Visual Attention. Figure 2 shows more results for the visualization of our self-supervised learned attention model on two videos from the task 'assemble clarinet' (left) and 'perform CPR' (right) from ProceL. Notice that our method successfully learns to focus on important region of each frame. For example, for clarinet, it focuses on cork, ligature, screws, lower and upper joints in the associated key-steps. This is particularly vital for localization of key-steps and recognition of the task, as also demonstrated by our quantitative results.

References

1. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition (2015)