STEP: Spatio-Temporal Progressive Learning for Video Action Detection

SUPPLEMENTARY MATERIAL

Xitong Yang[1*] Xiaodong Yang[2] Ming-Yu Liu[2]
Fanyi Xiao[3*] Larry Davis[1] Jan Kautz[2]
[1]University of Maryland, College Park [2]NVIDIA [3]University of California, Davis

In this supplementary material, Section A summarizes the details of our two-branch architecture and how to generate the initial proposals. Section B presents more evidences of the spatial displacement problem in action detection. Section C provides more algorithm and result analysis.

A. Implementation Details

UCF101 Dataset. Table 4 shows the details of our two-branch architecture. The network takes as inputs a sequence of $512 \times 25 \times 25$ feature maps from the backbone network (i.e., VGG16) as well as a set of proposal tubelets. For each proposal, an RoI pooling layer extracts a sequence of fixed-length regional features from the feature maps. For temporal modeling in the global branch, we first spatially extend each proposal tubelet to incorporate more scene context, as described in Section 3.5 of the paper. We then forward the extended features to three 3D convolutional layers to obtain the global features. To perform action classification, the global features are flatten and fed into a sequence of fully connected (`fc`) layers, which finally output the softmax probability estimates over C classes plus background. To perform tubelet regression, the global features are concatenated along channel dimension with the regional features at each frame and then fed into another sequence of `fc` layers, which produce a class-specific regression output with the shape $4 \times (C+1)$ for each frame.

AVA Dataset. The overall architecture is the same as the one in Table 4 except that we do not introduce extra 3D convolutional layers for temporal modeling. Instead, the `Mixed_5b` and `Mixed_5c` in I3D are used and followed by a $1 \times 1 \times 1$ convolutional layer to downsample the channel dimension to 256.

We use 34 initial proposals in the experiments on AVA since this datasets involves more actions on average at each frame than UCF101. We define the 34 initial proposals following the practice in [24]. In details, we generate the initial proposals using a two-level spatial pyramid with [$4/3$, 2] scales and [$5/6$, $3/4$] overlap for each spatial scale. In other words, a sliding window with size $3W/4 \times 3H/4$ pixels and overlap ratio $5/6$ is used for the first level, and a sliding window with size $W/2 \times H/2$ pixels and overlap ratio $3/4$ is used for the second level. Here, W and H denote the width and height of the frames, respectively. We extract video frames in 12 fps and resize them to 400×400.

*Work done during an internship at NVIDIA Research.

Layer		Output size
Global branch		
conv1	3×3×3, 1024	6×7×7
	max pool	
conv2	3×3×3, 512	3×7×7
	max pool	
conv3	3×3×3, 256	1×7×7
	average pool	
fc1(2)	4096	1×1×1
out	C+1, softmax	1×1×1
Local branch		
fc1(2)	4096	1×1×1
out	4×(C+1)	1×1×1

Table 4: Architecture of the two-branch network, where $T \times H \times W$, N represent the dimensions of convolutional kernels and output feature maps.

B. Spatial Displacement

The spatial displacement problem occurs in an action tube when the sequence is long and or involves rapid movement of people or camera. Here we analyze the spatial displacement problem on UCF101 by calculating the minimum IoU within tubes (MIUT). Given a ground truth action tube, MIUT is defined by the minimum IoU overlap between the center bounding box (i.e., the box of the center frame) and the other bounding boxes within the tube. Figure 10 demonstrates the statistics of different actions with different length using ground truth action tubes in the validation set. We observe that the spatial displacement problem is not very obvious for short clips (e.g., $K = 6$), as most

Figure 9: Examples of the spatial displacement problem. Red boxes indicate the ground truth bounding boxes and blue ones the spatial grids. From top to bottom are LongJump (ID: 12), FloorGymnastics (ID: 8) and CliffDiving (ID: 4).

Figure 10: MIUT of ground truth action tubes on UCF101. K denotes different tube lengths, and red dash line coorresponds to MIUT $= 0.5$.

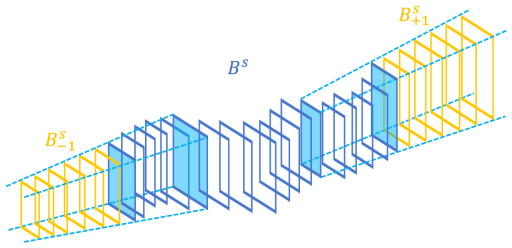

Figure 11: Illustration of extrapolation for adaptive temporal extension. Blue shaded boxes are the first and last bounding boxes of the corresponding tubelets.

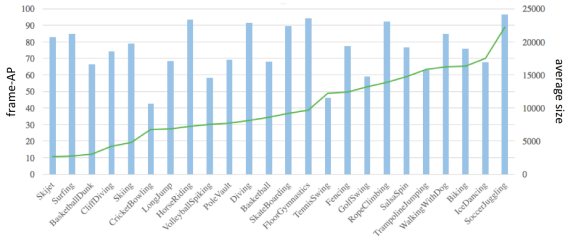

Figure 12: Analysis of the detection accuracy (blue) and the average size (green) of each action class.

action classes have high MIUT values. However, the spatial displacement problem becomes more severe for most actions when the sequence length increases. For example, "Skijet" (ID: 18) has a 0.12 MIUT and "CliffDiving" (ID: 4) has a 0.17 MIUT when $K = 30$, indicating both actions encounter large spatial displacements within the tubes. We also show some examples to illustrate the spatial displacement problem in Figure 9.

C. More Analysis

In order to tackle the spatial displacement problem, we introduce two methods to adaptively perform the temporal extension, i.e., extrapolation and anticipation as defined in Eqs.(4-5) of the paper. Figure 11 illustrates the extrapolation: for each of the current proposals, following its first and last tubelets (one tubelet with 6 bounding boxes), the extrapolation linearly estimates the directions and scales of the extended tubelets. As for the impact of different action scales, we qualitatively show the examples in Figure 8 of the paper, and we report the frame-APs and average sizes of different action classes of UCF101 in Figure 12. Thanks to the progressive learning, STEP is found to be robust to handle the actions with small scales, though it starts with coarse-scale proposals. Figure 13 demonstrates the per-class breakdown frame-AP on AVA.

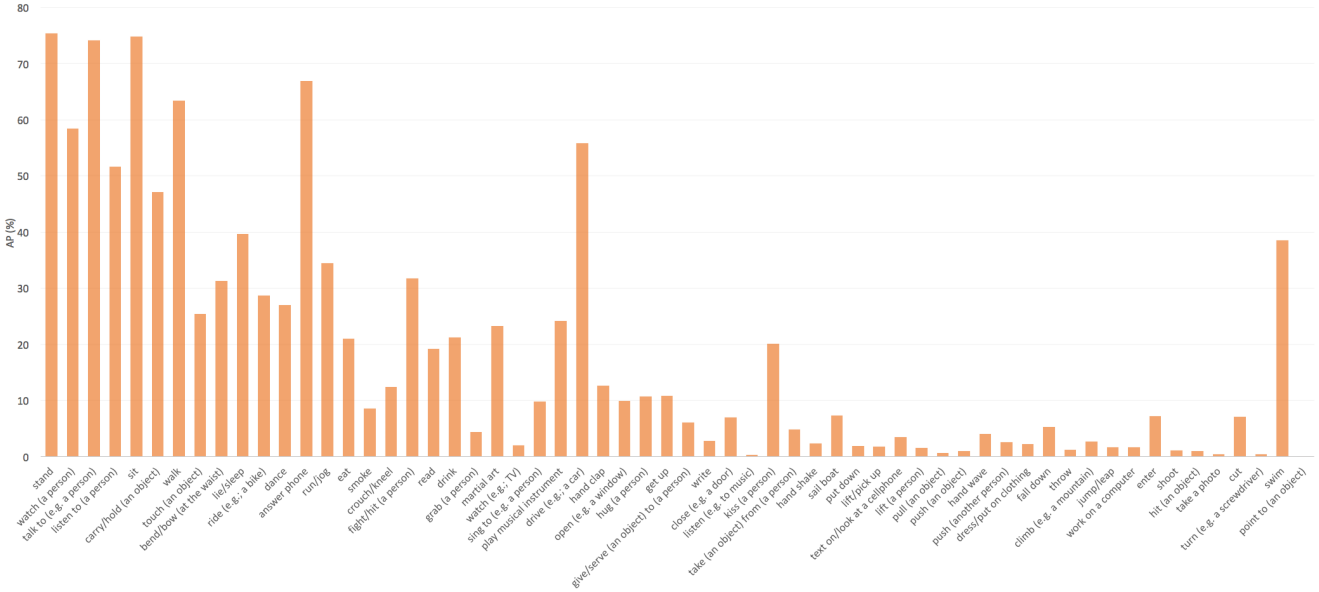

Figure 13: Comparison of the per-class breakdown frame-AP at IoU threshold 0.5 on AVA.