Human Action Segmentation using 3D Fully Convolutional Network

Pei Yu[1]
pyi980@eecs.northwestern.edu
Jiang Wang[2]
wangjiang@google.com
Ying Wu[1]
yingwu@eecs.northwestern.edu

[1] Northwestern University
2145 Sheridan Road
Evanston, IL, USA

[2] Google Research
1600 Amphitheatre Pkwy,
Mountain View, CA, USA

Abstract

Detailed action analysis, such as action detection, localization and segmentation, has received more and more attention in recent years. Compared to action classification, action segmentation and localization are more useful in many practical applications that require precise spatio-temporal information of the actions. However, performing action segmentation and localization is more challenging, because determining the pixel-level locations of action not only requires a strong spatial model that captures the visual appearances for the actions, but also calls for a temporal model that characterizes the dynamics of the actions. Most existing methods either use hand-crafted spatial models, or can only extract short-term motion information. In this paper, we propose a 3D fully convolutional deep network to jointly exploit spatial and temporal information in a unified framework for action segmentation and localization. The proposed deep network is trained to combine both information in an end-to-end fashion. Extensive experimental results have shown that the proposed method outperforms state-of-the-art methods by a large margin.

1 Introduction

Action segmentation and localization is an important and challenging task. It provides precise action boundaries both in temporal and in spatial domain, which is required in many applications, *e.g*. robotic surgery, autonomous vehicle. This task is challenging because of the difficulties in representing action visual appearances to locate the precise spatial boundary, modeling action motion to find the temporal position, and combining both information in a unified model.

Action segmentation is very different from conventional action recognition task. Compared to action classification, which answers "what" action for a video, *i.e*. action label for an entire video, the tasks of action segmentation, localization and detection answer more challenging questions of "where" and "when" for action understanding. For action detection, it provides a spatio-temporal bounding box for each action. In contrast, action segmentation provides a more precise pixel-frame-level action boundary. As shown in [10, 19], a precise action boundary can further improve classification accuracy. Recently, impressive progresses

have been made in addressing action classification [1, 23]. Nevertheless, the task of action segmentation has still been largely unexplored.

Action appearance modeling and motion modeling are the keys to approach to high quality action segmentation task. In recent years, Convolutional Neural Network (CNN) has shown its powerfulness in modeling action appearance [6]. CNN is also applied on optical flow fields to model short-term temporal dependency [6, 35]. In [19], supervoxel is extracted based on short-term motion. It has been shown that long-term modeling of temporal dependency is crucial to obtain accurate action class information [4], but most existing methods only model short-term dependency. Moreover, they generally model spatial and temporal information separately, and adopt a later fusion strategy for integration. This simple combination fails to fully exploit the correlations between visual and motion information.

In this paper, we propose a novel 3D fully convolutional network (3DFCN) for action segmentation. This new model represents spatial and temporal information jointly in a unified model that is trained in an end-to-end fashion. In addition, with 3D pooling layers, this new 3D fully convolution network is able to capture both short-term and long-term temporal information. We advance the 3D convolutional network [30] to a fully convolutional 3D network so as to create dense pixel-frame-level segmentation masks, inspired by fully convolutional network [18]. The output of this new model is the action segmentation mask, whose values indicate the action class for each pixel in each frame. With the prediction from 3DFCN as the action state posterior, we further investigate action state inference using dynamics modeling via Hidden Markov Model (HMM), and fully connected conditional random fields (CRFs) for accurate action localization.

The proposed method has following contributions:

1. To the best of our knowledge, this is the first work to propose a 3D fully convolutional deep network architecture for the task of action segmentation and recognition. Compared to the sliding window methods, the proposed method is computationally much more efficient, because the computation is shared spatially and temporally with 3D convolution on full videos.

2. The 3D fully convolutional network is able to capture spatio-temporal information jointly, rather than separately modeling spatial and temporal feature as done in the existing methods.

3. The network directly outputs the mask for different action class, rather than producing an action foreground mask to perform action classification inside this action foreground mask.

This paper is organized as follows. Sec. 2 introduces related work. Sec. 3 describes our proposed method. Sec. 4 evaluates the proposed method and the baseline methods. Sec. 5 concludes this paper.

2 Related Work

Action analysis includes several different tasks. Action classification is to classify the action category of one input video. Action detection and localization answer the questions of "where" and "when". It provides spatio-temporal bounding box for certain action. Action segmentation provides the mask of action, *e.g.*, "puppet" [10].

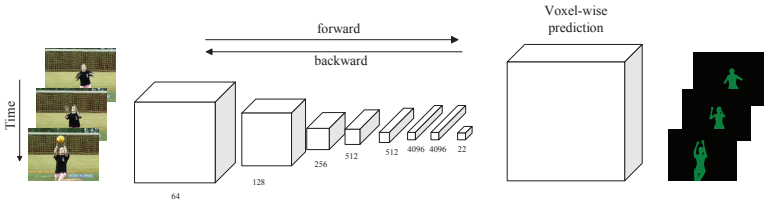

Figure 1: Illustration of framework.

Action classification has been intensively studied in the literature. Surveys can be found in [1, 23]. Different representations of video are proposed to classify actions. Examples of low level representations include histogram of 3D spatio-temporal gradients [15], Bag-of-Words method for encoding spatio-temporal interest points [17], dense point trajectories [32], local apperance and contextual information [33]. Besides low-level features, high-level features have also been explored, *e.g*. action bank [25], human pose [31].

In contrast, the tasks of action localization, detection and segmentation, have not been well explored compared to action classification. These tasks are very challenging, yet greatly beneficial to action classification. It is shown that precise human silhouette boundary can improve action classification accuracy [10, 19]. In order to localize action, one kind of method is to treat moving human body as a 3D spatio-temporal model, *e.g*. 3D shapes induced by motion of silhouettes [7], sliding window matching to an action template [3]. Instead of treating human body as a holistic template, other methods seek to use body parts to model action, *e.g*. 3D spatio-temporal deformable part model [28], dynamic-poselets [34], two-level hierarchy for space-time segments[20]. In a finer scale of initial segmentation, supervoxel is adopted for action localization and segmentation. In [14], 3D supervoxel and shape matching is proposed for video and action analysis. To handle over-segmented supervoxels, tree of sptio-temporal supervoxels for video segmentation [8], uniform entropy slice [37], MRF graph structure [19] are explored. Xu *et al*. propose a suite of 3D volumetric quality metrics to evaluate supervoxel quality [36]. To improve computational efficiency, spatio-temporal proposals [22] and tubelets [9] are proposed.

Recently, more and more methods turn to CNN for action recognition. 3D CNN is adopted to learn spatio-temporal feature for action classification of surveillance videos [11] . In [13], different fusion strategies are proposed to extend network connectivity of 2D CNN to temporal dimension. In [4], a novel recurrent convolutional architecture is proposed. In [30], modern 2D image classification network structure is extended to 3D convolution. Recently, these endeavors have been put in the task of action localization and segmentation. In [6, 35], two-stream 2D CNN structure, including one stream for visual appearance and the other for motion information, is explored for action detection. Although previous attempts [6, 35] have utilized CNN for spatio-temporal feature extraction, the learning process is not end to end, and they both need post processing for temporal localization. [18] applies CNN to end-to-end image segmentation and achieves impressive progress using fully convolutional networks (FCN). Deconvlution layer is used to upsample the prediction map to full scale. With FCN, many methods improve segmentation result with finer details [2]. This kind of method has potential for the task of video segmentation.

3 The Proposed Approach

3.1 Problem formulation

The task of action segmentation is to answer the questions of "where", "when" and "what" in action understanding for an input video. As shown in Fig. 1, with a video as the input volume, the output is the pixel-frame level prediction of action labels. Denote a video with T frames by $\{I_t\}, t = 1 \dots T$. The prediction can be treated as a function $f(I_t) = \mathbf{l}_t$, where $\mathbf{l}_t$ is the pixel-wise action label for frame t. $\mathbf{l}_t(x,y) = c \in \mathcal{C}$ denotes that pixel location (x,y) at frame t has action class c, where $\mathcal{C}$ is the set of action classes. For simplicity, we use $l_{t,i}$ to denote the label of pixel i at time t in this paper.

3.2 Spatio-temporal feature via 3D convolution

Action class is a high-level information, which depends on both visual appearances and temporal dynamics. Many efforts have been made to apply 2DCNN to learn spatio-temporal information. However, we believe that 3DCNN is more suited to capture this information by learning visual appearances and temporal dynamics within a unified framework.

2DCNN has inherent disadvantages for this task. At first, spatial and temporal feature are learned separately. In [6, 35], two-stream 2DCNN is adopted to separately learn the information of static images and motion images represented by optical flow. Appearance features and motion features are simply stacked for final classifier. Secondly, 2DCNN is unable to capture high level temporal information, *e.g.* 2D convolution on optical flow image cannot capture long-term temporal dynamics [6, 35]. In contrast, 3DCNN does not have these limitations since 3D convolution extracts 3D spatio-temporal feature jointly. Moreover, deeper 3D convolution layer is able to extract long-term temporal information with 3D pooling layers. This is completely different from 2D pooling of motion image, which is still in spatial dimension.

3.3 Network structure

Inspired by [5, 18, 26, 29], we propose a 3D fully convolutional network (3DFCN) for this task, by convolutionalizing fully connected layers in 3D classification network, which enables the network to handle input video with different sizes, and to yield spatio-temporal output prediction map. To obtain the full scale prediction, 3D deconvolution is added at the end of the network. Compared with naive sliding window method for pixel-frame-level prediction, the proposed 3DFCN is more efficient as it performs convolution on the entire volume, without redundant computation in the overlapping area of sliding windows.

The basic 3D classification network is inspired by C3D [30]. The proposed 3DFCN network structure is shown in Fig 2. With shorthand notation, the network structure is C(64, 3, 3, 1, 1)-P_1-C(128, 3, 3, 1, 1)-P_2-C(256, 3, 3, 1, 1)-C(256, 3, 3, 1, 1)-P_2-C(512, 3, 3, 1, 1)-C(512, 3, 3, 1, 1)-P_2-C(512, 3, 3, 1, 1)-C(512, 3, 3, 1, 1)-P_2-C(4096, 4, 1, 1, 1)-C(4096, 1, 1, 1, 1)-C($\mathcal{C}$, 1, 1, 1, 1)-*DeConv*($\mathcal{C}$, 64, 32, 32, 16). $C(d, f_1, f_2, s_1, s_2)$ denotes a 3D convolution layer with d filters of spatio-temporal kernel size $f_1 \times f_1 \times f_2$ and stride length $s_1 \times s_1 \times s_2$, where f_1 and s_1 is along the spatial axis, f_2 and s_2 is along temporal axis. Each convolution layer is followed by a ReLU layer. Two types of 3D pooling layers are used. P_1 is a 3D pooling layer with spatio-temporal kernel size $2 \times 2 \times 1$, and spatio-temporal stride length $2 \times 2 \times 1$, *i.e.*, it only performs pooling on spatial axes. The second

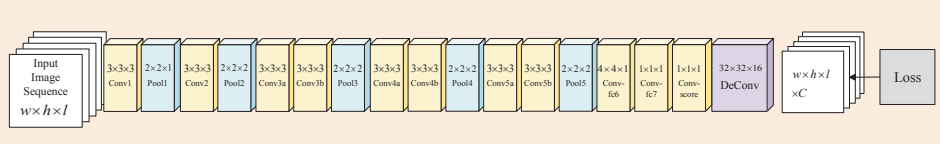

Figure 2: Network structure of 3D fully convolutional network.

one P_2 has spatio-temporal kernel size $2 \times 2 \times 2$ and spatio-temporal stride length $2 \times 2 \times 2$. $DeConv(\mathcal{C}, f_1, f_2, s_1, s_2)$ is a 3D deconvolution layer with kernel size $f_1 \times f_1 \times f_2$ and stride length $s_1 \times s_1 \times s_2$. Note that the last three convolution layers, C(4096, 4, 1, 1, 1), C(4096, 1, 1, 1, 1) and $C(\mathcal{C}, 1, 1, 1, 1)$ correspond to the fully connected layers in C3D [30] network. Let $FC(n)$ denote a fully connected layer with n output nodes. In C3D network, the input of $fc6$ layer $FC(4096)$ has spatio-temporal size of $4 \times 4 \times 1$. Hence, it can be converted to convolution layer C(4096, 4, 1, 1, 1). Similarly, C(4096, 1, 1, 1, 1) corresponds to $fc7$ in C3D network. These two convolution layers are followed by a ReLU layer and a dropout layer. $C(\mathcal{C}, 1, 1, 1, 1)$ corresponds to the class scoring fully connected layer in C3D network, where $\mathcal{C}$ denotes the number of prediction classes.

For an input video of 40 frames and each frame with a spatial resolution 320×240, the size of the output for the fully convolutional network is $7 \times 5 \times 3$. Equivalently, sliding window method of C3D network can generate same size of output with a spatio-temporal window size of $112 \times 112 \times 16$ and stride length $32 \times 32 \times 16$. 3DFCN network only takes 910ms to produce the $7 \times 5 \times 3$ output map, while C3D network takes 12075 ms. 3DFCN is more than 13 times faster than naive sliding window approach.

3.4 Dense sampling by dilated kernel

The model proposed in Sec. 3.3 has an equivalent $32 \times 32 \times 16$ down-sampling stride length of the output of the class scoring layer. With $32 \times 32 \times 16$ deconvolution layer, the output prediction still loses fine details. In order to refine the prediction detail, the sampling stride length needs to be decreased. Moreover, for each layer, the receptive field size should remain the same to ensure that the output gets the same amount of input information. By s_l, denote the stride length of $l-$th pooling layer. For example, when the stride length of pooling layer is reduced from s_l to $\frac{s_l}{2}$, the kernel size f_1 and f_2 of the following convolution layers needs to be increased to $2 \times f_1$ and $2 \times f_2$.

Inspired by [2, 5, 26], we apply dilation operation to 3D convolution kernels to obtain larger convolution kernel, rather than using normal large kernel, to avoid computation overhead. Without loss of generality, we illustrate it with 2D convolution in Fig. 3. The pooling stride length is decreased from 2 to 1. The convolution kernel is dilated from the original 2×2 2D convolution kernel to 4×4 by inserting zeros, while keeping the kernel parameters $w_{00} \sim w_{11}$ unchanged. As zero values in the convolution kernel have no computation, the actual amount of computation is the same of 2×2 convolution on a larger input feature map.

Here we evaluate different sampling stride settings of the class scoring layer, including spatio-temporal stride length $32 \times 32 \times 16$ and $8 \times 8 \times 4$. The model proposed in Sec. 3.3 has spatio-temporal sampling stride length of $32 \times 32 \times 16$. For the model with sampling stride length of $8 \times 8 \times 4$, the stride size of *Pool*4 and *Pool*5 layer is reduced to 1, *Conv*5*a*

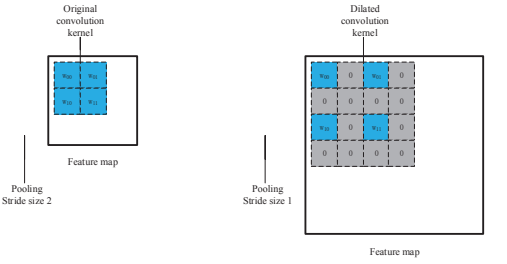

Figure 3: Illustration of kernel dilation.

layer and $Conv5b$ layer are $2\times$ dilated, $Conv-fc6$ layer is $4\times$ dilated. For the model with sampling stride length $32\times32\times16$, the deconvolution layer has kernel size $64\times64\times32$ and spatio-temporal stride length $32\times32\times16$. For the model with sampling stride length $8\times8\times4$, the deconvolution layer kernel size is $16\times16\times8$ with stride length $8\times8\times4$.

3.5 Modeling temporal dynamics with HMM belief

Although the prediction from 3DFCN is based on spatio-temporal feature, 3DFCN network lacks modeling of the dynamics of the output prediction of action state at different time. To bring temporal dynamics to the final prediction, we apply a typical HMM model to the network prediction. Denote the spatio-temporal observation by $z_{t,i}$. State transition probability $P(l_{t+1,i}|l_{t,i})$ can be estimated from ground truth labeling. We use the same state transition probability for all pixel locations.

The output softmax probability of 3DFCN provides posterior $P(l_{t,i}|z_{t,i})$ based on spatio-temporal observation $z_{t,i}$ at time t. With T frames of a video, we can obtain the belief of hidden state based on all observations $P(l_{t,i}|\underline{Z}_{T,i})$, where $\underline{Z}_{t,i}=\{z_{1,i},\ldots,z_{t,i}\}$. The class label can be inferred via maximum a posteriori (MAP) using belief probability.

Based on HMM model, we can easily get

$$P(l_{t,i}|\underline{Z}_{T,i}) \propto \frac{P(l_{t,i}|\underline{Z}_{t,i})P(l_{t,i}|\bar{Z}_{t,i})}{P(l_{t,i})} \tag{1}$$

$$P(l_{t,i}|\underline{Z}_{t,i}) \propto \frac{P(l_{t,i}|z_{t,i})}{P(l_{t,i})}\int_{l_{t-1,i}} P(l_{t,i}|l_{t-1,i})P(l_{t-1,i}|\underline{Z}_{t-1,i}) \tag{2}$$

$$P(l_{t,i}|\bar{Z}_{t,i}) \propto P(l_{t,i})\int_{l_{t+1,i}} \frac{P(l_{t+1,i}|z_{t+1,i})P(l_{t+1,i}|l_{t,i})}{P^2(l_{t+1,i})}P(l_{t+1,i}|\bar{Z}_{t+1,i}) \tag{3}$$

where $\bar{Z}_{t,i}=\{z_{t+1,i},\ldots,z_{T,i}\}$, $P(l_{t,i})$ is the prior distribution. We use the same prior for all pixel locations.

$$P(l_{t,i}) = \int_{l_{t-1,i}} P(l_{t,i}|l_{t-1,i})P(l_{t-1,i}) \tag{4}$$

$P(l_{1,i})$ and $P(l_{t,i}|l_{t-1,i})$ are estimated from ground truth label.

3.6 Accurate localization with fully connected conditional random fields

Recent semantic image segmentation literatures apply fully connected CRFs for accurate localization [2, 21], with an efficient method proposed in [16]. We also apply it on single image to refine the action segmentation boundary. The energy function of label $\mathbf{l_t}$ is

$$E(\mathbf{l}_t) = \sum_i \psi_u(l_{t,i}) + \sum_{i<j} \psi_p(l_{t,i}, l_{t,j}) \tag{5}$$

where $\psi_u(l_{t,i})$ is the unary potential of label $l_{t,i}$, $\psi_p(l_{t,i}, l_{t,j})$ is the pairwise potential. We evaluate two unary potentials. One is $\psi_u(l_{t,i}) = -\log P(l_{t,i}|z_{t,i})$ using output softmax probability of 3DFCN. The other one is $\psi_u(l_{t,i}) = -\log P(l_{t,i}|\underline{Z}_{T,i})$ using belief of HMM. The pairwise potential $\psi_p(l_{t,i}, l_{t,j})$ we use is

$$\mu(l_{t,i}, l_{t,j})(w^{(1)} \exp(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_{t,i} - I_{t,j}|^2}{2\theta_\beta^2}) + w^{(2)} \exp(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2})) \tag{6}$$

where $\mu(l_{t,i}, l_{t,j}) = 1$, if $l_{t,i} \neq l_{t,j}$, otherwise zero, p_i is the pixel position of pixel i, $I_{t,i}$ is the color vector at pixel i at time t. Label $\mathbf{l}_t$ is obtained by minimizing the energy function. For details of optimization, please refer to [16].

4 Experiments

4.1 Experimental setup

Dataset Experiments are performed on joint-annotated human motion data base (J-HMDB), which provides detailed action segmentation mask. Note that other action recognition datasets, *e.g.*, UCF-Sports[24, 27], only provide bounding box as ground truth, which does not well fit action segmentation task. J-HMDB contains 21 categories of action. Each action class contains 36-55 video clips, where each clip has 15-40 frames. In total, there are 31838 annotated frames. Training and testing splits follow the splits provided in J-HMDB dataset. The training testing ratio of distinct video sources is close to $7:3$.

Evaluation protocol Two metrics are used to evaluate the performance, foreground Intersection Over Union (Foreground IOU) and localization Intersection Over Union (Localization IOU), same as [19]. Action foreground IOU takes action mask of all categories of action as the foreground mask, and calculates the intersection over union between the prediction mask and the ground truth mask. Compared with foreground IOU, localization IOU counts the IOU of prediction masks with the same action label of the ground truth action label. Note that action segmentation does not provide action label for an entire video, but pixel-frame-level action label, hence we do not compare the performance of recognition of an entire video.

Baseline methods Recently reported segmentation performance on J-HMDB dataset includes [19]. Moreover, no public datasets except J-HMDB dataset provide action mask. [19] is chosen as baseline method. We also compare the performance of different network settings. We compare the network with sampling stride of $32 \times 32 \times 16$ and $8 \times 8 \times 4$, denoted as 3DFCN-32s and 3DFCN-8s. Moreover, we investigate the influence of belief inference and fully connected CRFs. We evaluate three different cases, 3DFCN-8s+HMM, 3DFCN-8s+CRFs and 3DFCN-8s+HMM+CRFs. In addition, we evaluate the performance of 2D

Method	Foreground IOU	Localization IOU
3DFCN-32s	47.1	19.7
3DFCN-8s	54.5	21.7
3DFCN-8s + HMM	53.3	21.0
3DFCN-8s + CRFs	55.6	**23.8**
3DFCN-8s + HMM + CRFs	**56.2**	23.0
2DFCN-8s	32.0	7.1
2DFCN-8s + HMM	30.0	6.5
2DFCN-8s + CRFs	29.8	8.8
2DFCN-8s + HMM + CRFs	34.1	7.8
[19]	48.8	-

Table 1: Performance of proposed methods and baseline methods.

CNNs, which learn spatial and temporal feature separately and perform frame-wise segmentation. We adopt a two-stream design similar to [6]. Optical flow is calculated using the same method in [6]. Each stream of 2D CNN is 2D FCN by replacing 3D layers in 3DFCN-8s with 2D layers. For fair comparison, we use the same dilation operation. The 2D CNN baseline is denoted as 2DFCN-8s.

4.2 Network settings

Loss function As this problem is inherently a multi-class classification, we adopt multinomial logistic loss after a softmax layer. The loss is weighted since action classes are unbalanced, *e.g.* the background usually covers more pixels than action foreground. The weight for background class is set to 0.1, while the weight for each action class is set to 10.

Optimization The network is trained using SGD with momentum. The batch size is set to 1 since different videos have different temporal lengths, same as model provided by [18] . Note that with batch size 1, the loss is still an average of the output prediction map, not a single prediction. The momentum is set to 0.99. Weight decay is set to 0.0005. Learning rate is fixed to 10^{-6}. The learning rate of biases is double of base learning rate. Training stops after 160000 iterations, roughly 240 epochs. The network is fine-tuned from C3D model. The filters of deconvolution layer are initialized as bilinear interpolation filter. We augment the data by random mirroring, and random cropping along spatial dimensions for spatial jittering. The training of 2DFCN-8S adopts the same learning rate, momentum and weight decay. Batch size is set to 4. Training stops after 160000 iterations. The image stream network is fine-tuned from FCN32S [18], while the flow image stream network weights are initialized using "xavier" method.

Implementation All of the networks are implemented with caffe[12] future branch [18]. For the convolution layers with dilated kernels, we use caffe engine as convolution algorithm. For other convolution layers and deconvolution layer, we use CuDNN convolution method to save memory. The hardware we use is NVIDIA GTX TITAN X with 12 GB graphics memory. Due to the memory overhead incurred by dilated convolution operation, for 3DFCN-8s, the input video is resized to a spatial size of 280×210. Temporal dimension is unchanged.

CRFs paramters For the pairwise potential in CRFs, $w^{(1)}$ is set to 10. θ_α is set to 100. θ_β is set to 10. $w^{(2)}$ is set to default value 3, θ_γ is set to default value 3. Result is obtained after 10 iterations.

Figure 4: Comparison of different settings with 8s sampling stride.

4.3 Results

In this section, we compare the proposed methods with baselines. The results are shown in Tab. 1. Note that the localization IOU performance of [19] is not reported. At first, we evaluate different sampling stride settings. Compared with 3DFCN-32s, 3DFCN-8s improves the performance of foreground IOU and localization IOU. It is shown that 3DFCN-based methods outperform 2DFCN-based methods by a significant margin, which demonstrates the superiority of learning spatio-temporal feature jointly with 3D convolution.

With the same sampling stride length of 8, different post processing methods are evaluated. 3DFCN-8s+HMM+CRFs outperforms [19] by a margin of 7.4% on foreground IOU. Compared with base setting 3DFCN-32s, it improves by 9.1% on foreground IOU and by 3.3% on localization IOU. Fig. 4 shows a qualitative comparison of the proposed methods with sampling stride length 8. For 3DFCN-8s, it is shown that the predictions between frames are still not coherent enough, *e.g.*, it fails to mask the feet of the first frame as action foreground. With belief $P(l_{t,i}|\underline{Z}_{T,i})$ as posterior, the prediction of 3DFCN-8s+HMM is more coherent across frames, yet generates a coarse action localization. Directly using CRFs without belief as posterior gives better localization, but aggravates the temporal incoherency. Using both CRFs and belief $P(l_{t,i}|\underline{Z}_{T,i})$ provides better result.

5 Conclusion

In this paper, we propose 3D fully convolutional network to address action segmentation and localization, a challenging problem of action recognition. The proposed network is trained end-to-end. With 3D convolution and pooling, it jointly exploits spatial and temporal information. With dilated convolution kernel and 3D deconvolution, it is able to yield prediction at the full spatio-temporal resolution with different precision. In addition, this paper investigates temporal dynamics using a HMM model, and fully connected CRFs for accurate spatial localization of action. The performance of the proposed method outperforms state-of-the-art

method by a large margin.

Acknowledgement

This work was supported in part by National Science Foundation grant IIS-1217302, IIS-1619078, and the Army Research Ofice ARO W911NF-16-1-0138.

References

[1] Jake K Aggarwal and Michael S Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[3] Konstantinos G Derpanis, Mikhail Sizintsev, Kevin Cannons, and Richard P Wildes. Efficient action spotting based on a spacetime oriented structure representation. In *CVPR*, pages 1990–1997, 2010.

[4] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015.

[5] Alessandro Giusti, Dan C Cireşan, Jonathan Masci, Luca M Gambardella, and Jürgen Schmidhuber. Fast image scanning with deep max-pooling convolutional neural networks. *arXiv preprint arXiv:1302.1700*, 2013.

[6] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *CVPR*, pages 759–768, 2015.

[7] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *PAMI*, 29(12):2247–2253, 2007.

[8] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, pages 2141–2148, 2010.

[9] Manan Jain, Jan Van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees GM Snoek. Action localization with tubelets from motion. In *CVPR*, pages 740–747, 2014.

[10] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *ICCV*, pages 3192–3199, 2013.

[11] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *PAMI*, 35(1):221–231, 2013.

[12] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[13] Andrej Karpathy, George Toderici, Sachin Shetty, Tommy Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014.

[14] Yan Ke, Rahul Sukthankar, and Martial Hebert. Spatio-temporal shape and flow correlation for action recognition. In *CVPR*, pages 1–8, 2007.

[15] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, pages 275–1, 2008.

[16] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*, 2011.

[17] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008.

[18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014.

[19] Jiasen Lu, Jason J Corso, et al. Human action segmentation with hierarchical supervoxel consistency. In *CVPR*, pages 3762–3771, 2015.

[20] Shugao Ma, Jianming Zhang, Nazli Ikizler-Cinbis, and Stan Sclaroff. Action recognition and localization by hierarchical space-time segments. In *ICCV*, pages 2744–2751, 2013.

[21] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, pages 1520–1528, 2015.

[22] Dan Oneata, Jérôme Revaud, Jakob Verbeek, and Cordelia Schmid. Spatio-temporal object detection proposals. In *ECCV*, pages 737–752. 2014.

[23] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[24] Mikel D Rodriguez, Javed Ahmed, and Mubarak Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, pages 1–8, 2008.

[25] Sreemanananth Sadanand and Jason J Corso. Action bank: A high-level representation of activity in video. In *CVPR*, pages 1234–1241, 2012.

[26] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[27] Khurram Soomro and Amir R Zamir. Action recognition in realistic sports videos. In *Computer Vision in Sports*, pages 181–208. Springer, 2014.

[28] Yicong Tian, Rahul Sukthankar, and Mubarak Shah. Spatiotemporal deformable part models for action detection. In *CVPR*, pages 2642–2649, 2013.

[29] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, pages 1799–1807, 2014.

[30] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3d: generic features for video analysis. *arXiv preprint arXiv:1412.0767*, 2014.

[31] Chunyu Wang, Yizhou Wang, and Alan L Yuille. An approach to pose-based action recognition. In *CVPR*, pages 915–922, 2013.

[32] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011.

[33] Jiang Wang, Zhuoyuan Chen, and Ying Wu. Action recognition with multiscale spatio-temporal contexts. In *CVPR*, pages 3185–3192, 2011.

[34] Limin Wang, Yu Qiao, and Xiaoou Tang. Video action detection with relational dynamic-poselets. In *ECCV*, pages 565–580. Springer, 2014.

[35] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, pages 3164–3172, 2015.

[36] Chenliang Xu and Jason J Corso. Evaluation of super-voxel methods for early video processing. In *CVPR*, pages 1202–1209, 2012.

[37] Chenliang Xu, Spencer Whitt, and Jason J Corso. Flattening supervoxel hierarchies by the uniform entropy slice. In *ICCV*, pages 2240–2247, 2013.