An Examination of Proposal-based Approaches to Fine-grained Activity Detection in Untrimmed Surveillance Videos

Sathyanarayanan Aakur, Daniel Sawyer, Michal Balazia, and Sudeep Sarkar
Department of Computer Science and Engineering
University of South Florida, Tampa
{saakur, danielsawyer, balazia, sarkar}@mail.usf.edu

Abstract

Spatiotemporal localization of activities in untrimmed surveillance videos is a hard task, especially when given the occurrence of simultaneous activities across different temporal and spatial scales. We tackle this problem using a cascaded region proposal and detection (CRPAD) framework implementing frame-level simultaneous activity detection, followed by tracking. We propose the use of a frame-level spatial detection model based on advances in object detection and a temporal linking algorithm that models the temporal dynamics of the detected activities. We also evaluate a proposal-based approach to the multi-activity, multi-label problem through cascaded modules of detection, tracking and recognition. A combination of handcrafted rules and deep learning methods show encouraging results to the activity detection problem. We show results on the VIRAT dataset through our participation at the recent 2018 TRECVID ActEV Challenge.

1. Introduction

We aim to address the problem of spatiotemporal activity detection. Given an untrimmed video sequence with multiple activities, our goal is to detect, classify and track every activity and their constituent actors at frame-level. Compared with object detection and recognition, the task of activity detection and tracking presents an even stiffer challenge due to arbitrary temporal duration of activities, presence of simultaneous activities and the reality of large intra-class variation of activities. These issues are exposed by the TRECVID Activities in Extended Video (ActEV) Challenge[1] [1, 2], in which we participated as the USF Bulls team. The task was to detect 19 activities of interest in surveillance videos of the VIRAT dataset[2] [3].

Advances in object detection and recognition have inspired a variety of approaches to activity detection based on deep learning [4, 5]. A common approach has been to detect activities in individual frames or short frame snippets and then to temporally link such spatiotemporal regions (called action tubes) to detect activity segments. Such methods process and fuse motion- and appearance-based features separately. There have been several approaches to temporal detection of activities in untrimmed videos [6, 7, 8, 9, 10] and spatial detection in trimmed videos [11, 12]. There have, however, been fewer of those to tackle the problem of spatiotemporal localization of activities [13, 14, 15]. Some approaches to temporal segmentation have the underlying assumption that there are no simultaneous activities occurring in the same temporal segment. However, surveillance videos such as those in the VIRAT dataset pose a different set of problems: (1) there can be multiple simultaneous activities, (2) an actor or object can have multiple activity labels and (3) there are large intra-class variations and small inter-class variations. The latter poses a significantly different challenge to traditional approaches to object detection, where there are less intra-class variations and more inter-class variations.

We build a cascaded region proposal and detection framework involving frame-level simultaneous activity detection and tracking. The proposed approach has two major parts: (1) a spatial activity proposal network based on YOLO [5] and (2) a probabilistic temporal linking model that takes frame-wise spatial detections and outputs action tubes such as those in [13]. This approach has been proposed in [16] and for completeness we describe it again in Section 2.

2. Cascaded Region Proposal and Detection Framework

As shown in Figure 1, a feature extraction network and a frame-wise activity detection network are combined to generate frame-level activity proposals. These frame-wise detection outputs are linked temporally to generate spatiotem-

[1] https://actev.nist.gov/
[2] http://www.viratdata.org/

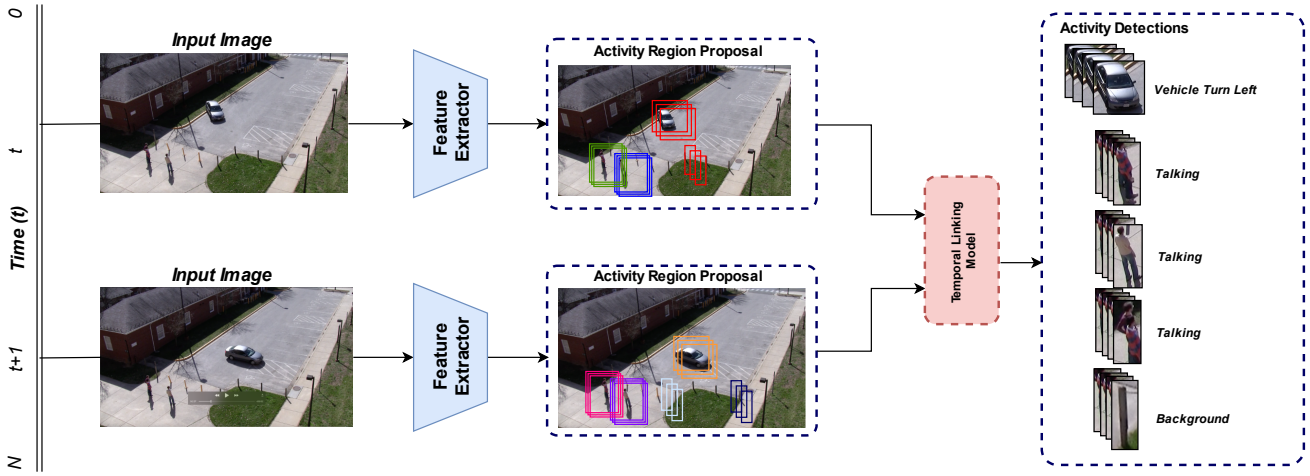

Figure 1. **Overall architecture**: The proposed approach is shown here. There are three basic components to the approach: a feature extraction network, a frame-wise activity detection network and a temporal linking model.

poral activity detections. In this section, we discuss model architecture and training procedures of the proposed Cascaded Region Proposal and Detection (CRPAD) framework.

2.1. Frame-wise Activity Region Proposal

We use a spatial activity detection network, similar to that proposed by Redmon *et al.* [5]. We then extract spatial features using the deep CNN, Darknet-53, a variation of Darknet-19 [17]. The activity proposal network is built on top of the final convolutional layer of the Darknet-53 network. An *action-ness* map is constructed by convolving across the final activation map of the CNN feature extraction model and constructing an $S \times S$ grid for the image and class probabilities for each of the grids. We then fix the grid size to be 13×13 and predict the probability of each class occurring in each grid. Finally, we use this grid to regress the bounding box similar to what is proposed in [5].

Training Details: We first pre-train the feature extraction network, Draknet-53 on the ImageNet dataset [18]. We choose Draknet-53 after experimenting with different, shallower architectures such as VGG16 (CITE). We find that the shallower networks are not as conducive to fine-grained detection as Draknet-53. We then train the final detection layer on the target dataset, VIRAT. The network was trained end-to-end at the frame-level, with the minibatch size set to be 128. We ensured that the minibatch was balanced across activities by selection of frame images across sequences.

2.2. Temporal Linking Model

We employ a probabilistic temporal linking, based on the Viterbi algorithm [19]. The bounding box region proposals ($d \in D$) from the activity proposal network defined in Section 2.1 are used as input to the temporal linking model. We denote the i-th region proposal from time t by d_t^i. The temporal affinity score between two regions d_t^i and d_{t+1}^j from consecutive frames is given by

$$S_c(d_t^i, d_{t+1}^j) = (1-\beta)E_c(d_t^i) + \beta E_c(d_{t+1}^j) + \psi_{d_t^i, d_{t+1}^j} \quad (1)$$

where $\psi_{d_t^i, d_{t+1}^j}$ is the distance between the center of the bounding boxes d_t^i, d_{t+1}^j and $E_c(\cdot)$ is the class confidence score of the given region proposal and β is a temporal memory factor. The linking score is high for region proposals that share class confidence scores and overlap highly in consecutive frames. The region proposals with maximum total temporal affinity are combined to form action tube proposals. We also experimented with a greedy linking algorithm for combining high confidence bounding boxes in successive frames, but found that the Viterbi algorithm performs better.

2.3. Evaluation

We evaluate the efficacy of the proposed approach at each stage of the pipeline. For evaluating the spatial localization capacity of the region proposal network, we set a threshold of 50% Intersection over Union (IOU) and compute the accuracy of detected bounding boxes regardless of the predicted class. We evaluate the overall framework by the probability of missed detection as defined in the 2018 TRECVID ActEV Challenge [1, 2].

From our experiments, we found the region proposal network to have an accuracy of 57.23% across all activities on the validation set. This corresponds to the detection of 38.64% activities on the validation set, with a threshold of 10% temporal overlap between the ground truth and frame-wise bounding boxes predictions. Overall, the proposed approach achieves a p-miss of 85% and 93.4% at the rate of 0.15 frames per second on the validation and test sets respectively and 68.12% p-miss at the rate of 1.0 false alarms per minute on the validation set.

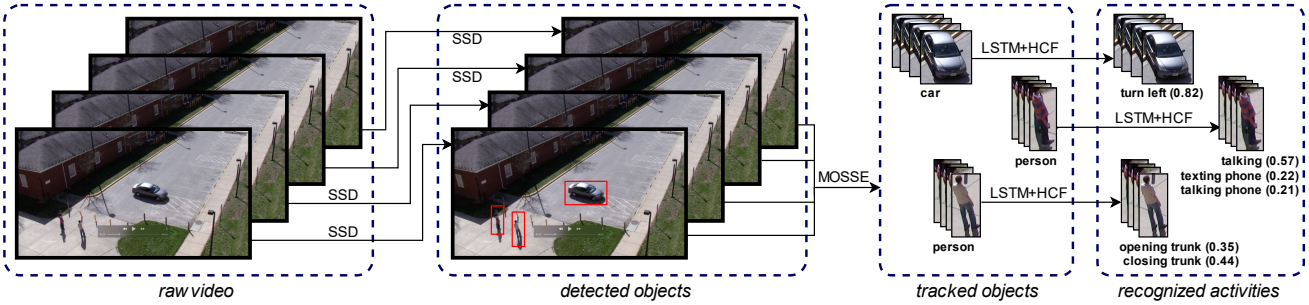

Figure 2. Architecture of the method consists of three steps: detection by SSD, tracking by MOSSE and recognition by LSTM+HCF.

3. Detection-Tracking-Recognition Approach

In this section, we describe an alternative approach to activity detection. We use a cascaded framework of detection, tracking and recognition for multi-activity detection in surveillance videos. The overall architecture is shown in Figure 2. We describe each module in detail below.

3.1. Detection

The first step is to detect objects independently in frames of input videos. We adopt the Single-Shot Detector (SSD) [20] for multiple object classes that predicts class scores and bounding boxes for a fixed set of default bounding boxes using small convolutional filters applied to feature maps extracted by VGG16.

SSD discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to handle objects of various sizes. The overall objective loss function is a weighted sum of localization loss L_{loc} between predicted box (l) and ground truth box (g) parameters and confidence loss L_{conf} over multiple class confidences (c)

$$L\left(x,c,l,g\right)=\frac{1}{N}\left(L_{conf}\left(x,c\right)+\alpha L_{loc}\left(x,l,g\right)\right) \quad (2)$$

where x are indicators for matching default boxes to ground truth boxes, N is the number of matched default boxes and α is set to 1 by cross validation.

We run an implementation[3] of SSD on every 5-th frame of each video adjusted to 512×512. Resulting detections of confidence below 0.25 and object class not in $\{person, car, bus\}$ are filtered out.

[3]Code obtained from `https://github.com/elranu/ssd_pi/blob/master/ssd_predictor.py` and pre-trained weights from `https://github.com/elranu/ssd_pi/blob/master/trained_weights/VGG_coco_SSD_512x512.h5`.

3.2. Tracking

The second step is to track the detected objects by stitching their bounding boxes as their position and appearance changes in time during observation. We use a tracker based on training the Minimum Output Sum of Squared Error (MOSSE) [21] correlation filter initialized using a single frame (input) and trained on the subsequent frame (output) to predict location of each tracked object with maximum correlation. MOSSE is robust to variations in lighting, scale, pose, and non-rigid deformations.

Training is conducted in the Fourier domain to take advantage of a simple element-wise relationship between input and output. Let F_i, G_i and H_i be the Fourier transform of the input images, output images and filters, respectively. Then

$$G_i = F_i \odot H_i^* \quad (3)$$

where $\odot$ is element-wise multiplication and $*$ indicates the complex conjugate. MOSSE finds a filter H that minimizes the sum of squared error between the actual output of the convolution and the desired output of the convolution by

$$\min_{H^*} \sum_i \left|F_i \odot H^* - G_i\right|^2. \quad (4)$$

Fed with the SSD output from the first step, the used implementation[4] processes each frame of the input video by initializing new trackers on detected bounding boxes that are not assigned to any trackers and updating all existing trackers from previous frame. After the video is processed entirely, MOSSE outputs the object tubes adjusted to 320×240 and temporally segmented to the average length of individual activities in the ground truth.

3.3. Recognition

The final step is recognition of the object tubes into classes that represent 19 described activities of interest. We modify the multi-class VGG16-based Long Short-Term Memory (LSTM) [22] algorithm into a regressor that for

[4]Code obtained from `https://github.com/opencv/opencv/blob/master/samples/python/mosse.py`.

a given track outputs the confidence in a specific activity. Therefore, 19 LSTM regressors are learned to produce confidence scores for every activity.

The LSTM network is trained on an equal number of instances of a given activity as positive samples of other activities of the same subject type as negative samples. That is, the LSTM regressor for activity carrying is trained on about 200 instances of carrying and 200 instances of non-carrying activities a person can perform such as closing trunk or texting on phone. We use the ground truth annotations as input to train the network. We add a fully connected layer to produce class probabilities for each of the 19 classes in the dataset, based on the hidden state of the RNN at the final time step. We use the traditional cross entropy loss with softmax activation from the output of the fully connected layer. We use Stochastic Gradient Descent to the train the recognition network, with the number of time steps to the LSTM being 20. Learning rate is set 10^{-3} at the first epoch and reduced to 10^{-4} on epoch 10, then to 10^{-5} on epoch 30 and finally to 10^{-6} on epoch 100. Training is terminated on an epoch of saturated validation loss.

Additionally, a series of reasonable hand-crafted filters (HCF) is applied to the resulting confidence scores to suppress unlikely activities with a potentially high confidence caused by high visual similarity with respect to the LSTM features. These filters, for example, allow only vehicles to turn left or right, and only people to talk or ride a bike. Another filter calculates angle between directions at the beginning ($[x_b, y_b]$) and at the end ($[x_e, y_e]$) of a vehicle tube by the 2D arcus tangent

$$\arctan(x_b y_e - y_b x_e\,,\; x_b x_e + y_b y_e) \tag{5}$$

to classify the turn as left when negative or as right when positive. Finally, confidences in all activities of all static tubes are set to zero since there is no action.

3.4. Evaluation

The SSD produces bounding boxes with 50.3% misdetection rate and 92.3% false alarm rate. The bounding boxes are then temporally and spatially connected into tubes by MOSSE with similar 50.5% misdetection rate and 92.9% false alarm rate. Note that the false alarm rate is high mainly because even objects that are detected and tracked correctly are not engaged in any of the 19 activities in a large number of frames, thus they do not hit ground truth bounding boxes. Finally, on the validation set, the complete approach that consists of detection by SSD, tracking by MOSSE and recognition by LSTM trained using ground truth segmentation and HCF filters achieves a misdetection rate of 88.5% at the rate of 0.15 false alarms per minute and 76% misdetection rate at the rate of 1.0 false alarms per minute.

4. Discussion and Future Work

As can be seen from the experimental evaluation, the proposed approach provides a platform for spatial localization of activities at the frame level without explicit temporal modeling. We believe that the use of temporal modeling at both the semantic and feature levels would help in improving the performance of the proposed approach. Additionally, we find that our approach is able to identify the differences in fine-grained activity classes such as *vehicle turn left* and *vehicle turn right* with a high degree of accuracy, even without explicit temporal modeling.

References

[1] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, (New York, NY, USA), pp. 321–330, ACM Press, 2006.

[2] G. Awad, A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, D. Joy, A. Delgado, A. F. Smeaton, Y. Graham, W. Kraaij, G. Qunot, J. Magalhaes, D. Semedo, and S. Blasi, "Trecvid 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search," in *Proceedings of TRECVID 2018*, NIST, USA, 2018.

[3] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *Computer vision and pattern recognition (CVPR), 2011 IEEE conference on*, pp. 3153–3160, IEEE, 2011.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[6] S. Stein and S. J. McKenna, "Combining embedded accelerometers with computer vision for recognizing food preparation activities," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pp. 729–738, ACM, 2013.

[7] H. Kuehne, J. Gall, and T. Serre, "An end-to-end generative framework for video segmentation and recognition," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pp. 1–8, IEEE, 2016.

[8] B. L. Bhatnagar, S. Singh, C. Arora, C. Jawahar, and K. CVIT, "Unsupervised learning of deep feature representation for clustering egocentric actions," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 1447–1453, AAAI Press, 2017.

[9] J.-B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien, "Unsupervised learning from narrated

instruction videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4575–4583, 2016.

[10] L. Yao and Y. Qian, "Dt-3dresnet-lstm: An architecture for temporal activity recognition in videos," in *Pacific Rim Conference on Multimedia*, pp. 622–632, Springer, 2018.

[11] K. Soomro, H. Idrees, and M. Shah, "Action localization in videos through context walk," in *Proceedings of the IEEE international conference on computer vision*, pp. 3280–3288, 2015.

[12] L. Wang, Y. Qiao, X. Tang, and L. Van Gool, "Actionness estimation using hybrid fully convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2708–2717, 2016.

[13] G. Gkioxari and J. Malik, "Finding action tubes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 759–768, 2015.

[14] P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Learning to track for spatio-temporal action localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 3164–3172, 2015.

[15] Y. Tian, R. Sukthankar, and M. Shah, "Spatiotemporal deformable part models for action detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2642–2649, 2013.

[16] S. Aakur, D. Sawyer, and S. Sarkar, "Fine-grained action detection in untrimmed surveillance videos," in *Human Activity Detection in multi-camera, Continuous, long-duration Video (HADCV'19) Workshop, 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pp. 38–40, IEEE, 2019.

[17] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, IEEE, 2017.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, Ieee, 2009.

[19] G. D. Forney, "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *In Proc. of European Conference on Computer Vision 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 21–37, Springer International Publishing, 2016.

[21] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, June 2010.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.