Tokyo Tech at TRECVID 2021: Multi-Stage Framework for Video Action Detection

Ronaldo Prata Amorim
ronaldo@ks.c.titech.ac.jp

Nakamasa Inoue
inoue@ks.c.titech.ac.jp

Koichi Shinoda
shinoda@ks.c.titech.ac.jp

Tokyo Institute of Technology

Abstract

We utilize an action detection system for detecting human and vehicle actions in long untrimmed videos, submitted for the TRECVID Activities in Extended Video (ActEV) 2021 challenge [1]. It separates the task into an object detection and tracking stage to divide the initial video into object tracks for all possible actors, which are then divided into several short clips which are input into an action recognition model, that classifies each clip with relation to all relevant action classes.

Besides the VIRAT dataset utilized for the challenge, we utilize networks pretrained on the ImageNet and Kinetics-700 datasets. Summaries of the different submitted runs are as follows:

- 26508 - Action recognition model using cubed clips - clips with bounding boxes that do not move over time.
- 26509 - Action recognition model using sparse sampling of clips for training.
- 26510 - Model combining the elements of the Cube and Sparse models.

From the run results, we can see that neither variant improves the model's performance in any significant way and in fact the performances for the Cube and Sparse models are inverted when compared to their performance with the validation subset. This seems to be the result of poor implementation, as this model is otherwise very similar to the model submitted by another group in a previous edition of ActEV, with few missing parameters.

1 Introduction

Over the last few years, research in the area of image and video analysis has steadily advanced into more and more complex tasks, especially in regards to the spread of video tasks and the added challenge of tackling previous tasks, such as image understanding and object detection, with the additional element of time. While such tasks have had their breakthroughs and steadily advanced, their performance and reliability are still far behind their lower dimensional equivalents, due to the many difficulties inherently present in video analysis, such as the larger degree of object variability and the larger computing power required to incorporate the third temporal dimension into analysis. As such, in spite of the advent of ever bigger and denser video datasets annotated for such tasks, processing such videos in their entirety taking into account both spatial and temporal features simultaneously is still out of our grasp.

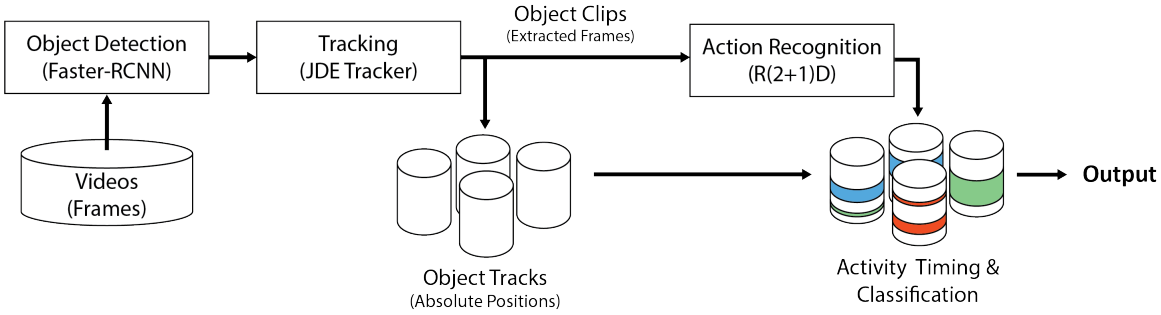

Figure 1: System overview

In view of this, we choose a method which attempts to isolate the spatial and temporal characteristics of a video, first spatially identifying actors which are expected to perform identifiable actions, and then cutting each actor's track into short clips, inputting each clip into an action recognition model to find what actions occur within them, and finally reassembling from these the full spatio-temporal actions.

2 System

Our system is based on a framework with two stages, a spatial stage and a temporal stage, as shown in Figure 1. First, an object detection network is run for in one out of every k frames of the input video, localizing all possible objects present in the video. These detections are then concatenated into object tracks utilizing a kalman filter-based object tracking system. These object tracks are then used to define object clips, short video tracks which represent part of an object's trajectory throughout the video, which are used as input for an action recognition network which predicts the actions performed by each object. These detected actions are fused with the previously found spatial information to identify the full spatial-temporal actions.

Comparing this system to the one we submitted for last year's ActEV task, while the initial spatial stage is nearly identical, with only a slight change in the framework used for object tracking, the most prominent difference was with the change from an action localization network, where entire object tracks were input in order to simultaneously classify and temporally localize possible actions, into an action recognition network, where object tracks are first split into several smaller clips, where the temporal localization is implicitly done through the fluctuation of classification results over the different clips.

2.1 Object Detection and Tracking

Given a video V composed of T frames, described as $V = \{I_t \in \mathbb{R}^{W \times H \times 3}\}_{t=1}^{T}$ with I_k being the k_{th} frame of the video with width W height H, we run an object detection network, e.g. Faster R-CNN [6], for every k_{th} frame, spatially localizing and classifying all objects on each frame which might perform the actions we wish to detect, which we call actors. In the case of the VIRAT dataset [5] utilized in this challenge, these actors are composed exclusively of persons and vehicles, so we focus on these two object classes for object detection.

This frame-wise actor detection information is then concatenated into object tracks, utilizing a Kalman Filter [9] based object tracking system. We initialize a different tracking system for each class of actors, and these systems receive as inputs the bounding boxes of every actor detected in the previous stage, as well as the visual features for each bounding box extracted by the detection model, and outputs for every actor a an object track $A_a = \{i_t \in \mathbb{R}^{w \times h \times 3}\}_{t=t_0}^{t_f}$. Here, i_k is the k_{th} frame of the video trimmed and centered around actor a, of width w and height h, t_0 and t_f are the first and last frames where actor A is present in the video respectively.

Table 1: Frame-wise object detection results on the VIRAT validation subset

person	vehicle	bike	parking meter	door	tree	dumpster	prop	push/pull object	mAP
31.7	69.6	2.1	73.2	57.0	94.0	89.4	5.0	17.8	48.9

2.2 Action Recognition

With object tracks for each actor A_a, we proceed to split each into several short overlapping clips C_a^c for input to the action recognition model, each clip having the same duration D_{clip} and the stride between clips S_{clip}, with $S_{clip} \leq D_{clip}$.

With several object clips for each actor in a video, we then input them to an action recognition network, e.g., the R(2+1)D network [7], which aims to temporally localize and classify all actions present in a given track. The input for this stage is the previously described object clips C_a^i, with the output being a confidence score for each action class. This is done separately for every clip of every object track of a given input video.

Combining this information with the previously described spatial information for each respective object track, we produce full spatio-temporal action detection results. Two major modifications performed on top of the action recognition model were cube-based clips and sparse clip sampling during training, which can be applied individually or at the same time.

2.3 Cube-based Clips

While a normal object clip C_a^i utilizes frames centered around actor a, whose position and aspect ration can change for each frame the object is present in the video, these variances can negatively impact the quality of the visual features utilized by the action recognition model. For this reason, we test the utilization of cube-based clips, in which a single central bounding box is utilized for the entire duration of the clip, ensuring that both the size and aspect ratio of the clip remain consistent, while also ensuring that the actor of which the clip is taken is present throughout its entirety.

2.4 Sparse Clip Loading

for this modification, we adapt the sparse-sampling strategy proposed in [8] during training, where the sampling of frames for generating training clips is performed by dividing the target clip into N segments, and from each segment we randomly sample a frame, concatenating these forming an N-length clip during training, while testing and validation is performed with the standard center cropping.

3 Experiments

Experiments were conducted for each stage separately, testing the performance for object detection and action recognition, as well as for the entire video action detection task, training with the VIRAT dataset's training subset and testing on its validation subset.

For object detection we utilize a Faster-RCNN model [6] with a ResNet-50 [3] backbone, pretrained on ImageNet [2] and refined on VIRAT's own training subset for frame-wise object detection. Object detection is realized every 5 video frames for the 13 most common object classes in the VIRAT dataset, although only person and vehicle detection results are utilized for the rest of the framework. The results of these tests can be seen on Table 1.

For action recognition we utilize the R(2+1)D network [7], pretrained on Kinetics [4] and trained on the canon tracks for each actor provided by VIRAT's training subset, aiming to classify all actions in which that actor participates. We experiment with four settings for model modifications, one baseline with no modifications, one with cube-based clips, one with sparse sampling, and one with both modifications. The results of these tests can be seen on Table 2.

Table 2: Action recognition results on the VIRAT validation subset

	Partial AUDC	Mean p-miss
Baseline	0.68634	0.59901
Cube	0.55858	0.45350
Sparse	0.54207	0.40858
Cube+Sparse	0.57489	0.46326

The results of the submitted runs on the VIRAT testing set can be seen on Table 3, along with the top-3 team results from this year and our results from previous the previous year. From it we can see that although results on the validation subset were promising, none of the submitted models had a reasonable performance, being very far from the best scoring models this year, and even performing worse than the models our group submitted for last year's ActEV task. Interestingly, although results on the validation subset seemed to indicate a slightly advantage of the Sparse model compared to Cube, that situation is inverted in the testing subset, indicating a possible failure in the integration of the two halves of the framework.

Table 3: Full spatio-temporal activity detection results on the VIRAT testing subset

	Partial AUDC	Mean p-miss
INF	0.39607	0.30622
BUPT-MCPRL	0.40853	0.32489
UCF	0.43059	0.34080
2020 Team Best	0.79753	0.75502
Cube	0.85158	0.81969
Sparse	0.90834	0.88472
Cube+Sparse	0.86028	0.83755

4 Conclusion

We presented our framework for video action detection in the context of the ActEv challenge, and our related experiments in modifications of the action recognition model in hopes of tackling the challenges inherent in this task. We showed the results of our experiments in both the open VIRAT validation subset for individual framework stages, as well as the closed VIRAT testing subset for the full framework, where good performance in validation did not translate to better performance in the testing subset compared to previous years. This difference can be possibly attributed to errors in implementation and integration of the framework used, as it is based on another that performed considerably better in a previous edition of the same task.

References

[1] G. Awad, A. A. Butt, K. Curtis, J. Fiscus, A. Godil, Y. Lee, A. Delgado, J. Zhang, E. Godard, B. Chocot, L. Diduch, J. Liu, Y. Graham, G. J. F. Jones, , and G. Quénot. Evaluating multiple video understanding and retrieval tasks at trecvid 2021. In *Proceedings of TRECVID 2021*. NIST, USA, 2021.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[4] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[5] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*, pages 3153–3160, 2011.

[6] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[7] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. *CoRR*, abs/1711.11248, 2017.

[8] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal segment networks: Towards good practices for deep action recognition. *CoRR*, abs/1608.00859, 2016.

[9] G. Welch, G. Bishop, et al. An introduction to the kalman filter, 1995.