CMU Informedia at TRECVID 2021: Activity Detection with Argus++

Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G. Hauptmann

Language Technologies Institute, Carnegie Mellon University
lijun@cmu.edu, yijunqia@andrew.cmu.edu, {wenhel, alex}@cs.cmu.edu

Abstract

In TRECVID 2021 ActEV task, we tackle multi-scale multi-instance activity detection in extended videos with the CMU *Argus++* [28] video understanding framework. The proposed method was the first to generate *overlapping spatio-temporal cube proposals* to ensure the coverage of activities in untrimmed video streams, instead of conventional non-overlapping cube or tube proposals. The well-designed four-stage framework achieves an ideal trade-off between computation cost and performance, achieving state-of-the-art performance within real-time on consumer-level hardware. The proposed system achieved the best performance on the TRECVID 2021 challenge live leaderboard [1], where efficiency measurement and execution verification were missing but would be desired. We further evaluate our method on the ActEV SDL benchmark series with fully-sequestered data and ready-to-run system submission. The outstanding performance of *Argus++* again emphasizes its robustness and superiority across a wide range of benchmarks where it has been leading for years.

1 Introduction

In the past decade, the computer vision community has witnessed a booming development of activity detection algorithms. In recent years, action detection in extended videos [4, 14] has drawn a widely attention and brought challenges to the area. Many of the extended videos are *unconstrained* videos captured by surveillance cameras in varies different indoor and outdoor scenarios. Activity detection in extended videos can be much difficult than others, because unconstrained videos usually are recorded in large field-of-views and contains large number of objects and activities simultaneously and continuously from time to time.

There are previous works achieving impressive performance on conventional activity detection [20, 6, 21, 10, 7]. However, most of them cannot work on extended videos. Some of these methods are only suitable to trimmed videos, i.e., the videos are pre-trimmed to several clips; Some other methods are designed for object-centered videos, which can only analyze one object at a time. Moreover, usually, conventional activity detection algorithms are only specified for certain scenarios, such as person sporting activity, etc. As a result, the performance of such algorithms would significantly downgrade when being applied to unconstrained videos.

Recent works [19, 27, 13] consider activity detection on unconstrained video as a two-stage algorithm: First, object detection and/or tracking algorithms are applied to the videos to locate the candidate objects. Second, the trajectories of the objects are straightforwardly used as tube/tubelete proposals for temporal activity localization. Simply utilizing trajectories, or so called tubelete proposals as candidate activity proposals would lose important information of activities, especially the motion

[1] Snapshot: https://web.archive.org/web/20211115022601/https://actev.nist.gov/trecvid21#tab_leaderboard.

related activities, such as 'vehicle turning right'. Because these tubelete proposals are focused on objects rather than motion of the objects, they always keep the bounding boxes around the objects. This would fail to capture important activity information, e.g., the trace of the objects. Another drawback of tubelete proposals is the object distortion in proposal. After the object detection and tracking, the video frames are cropped by multi-sized object-centred detection bounding boxes. When convert it to proposals, these cropped images must be resized into one size. Therefore, the objects in such proposals will suffer from the distortion problem because bounding boxes shifting and changing across frames. It will later harm the validity of the proposals in the activity analysis stage. After the generation of tubelete proposals, most of the previous works still rely on temporal activity localization to determine the start and end of the activities. Some works [19] generate non-overlapping proposals by straightforwardly cutting the proposals to several fixed length clips, which obviously would break the completeness of the activities.

We overcome the aforementioned problems on tubelet proposal activity detection by proposing a novel cube proposal activity detection system. The proposed system is a four-stage framework: Proposal Generation, Proposal Filtering, Activity Recognition and Activity Deduplication. In the system, we propose cube proposal instead of tubelet proposal naively generated from detection results. In cube proposal generation, we review the object trajectories and merge and crop the area of detected objects across the frames, such that context activity information, such as object traces, will be kept. Moreover, we propose a over-sampling method to generate overlapping proposals from the full video, and apply activity deduplication after recognition to keep both of the completeness and validity of activities. The proposed system outperforms the tubelet activity detection systems in the TRECVID ActEV 2021 challenge. The contributions of the proposed system are as follows:

1. We introduce overlapping spatio-temporal cubes as the activity proposals. By over-sampling, the generation of cube proposals ensures the coverage and completeness of activities in multi-scale multi-instance videos.
2. We proposed action recognition and deduplication algorithms to optimize the performance of action detection on cube proposals, which guarantees the validity of the activities.
3. The proposed system has achieved outstanding performance in TRECVID ActEV 2021.

2 Related Work

Object Detection and Tracking Object detection and tracking are fundamental computer vision tasks that aims to detect and track objects from images or videos. Image-based object detection algorithms, such as Faster R-CNN [18] and R-FCN [5], have demonstrated convincing performance but are often expensive to apply on every frame. Video-based object detection algorithms [30, 17] use optical flow guided feature aggregation to leverage motion information and reduce computation. With the deep features extracted from the backbone convolutional network, multi-object tracking algorithms [23, 22] associates objects across frames based on feature similarity and location proximity.

Activity Detection In recent years, there emerged some systems designed for spatio-temporal activity detection on unconstrained videos [19, 27, 13, 3, 25, 29]. Generally, theses systems first generates activity proposals and then feeds them to classification models. Since there have been a variety of video classification networks [20, 11, 6], the major focus is on the paradigm of proposals and the generation algorithm. In [13, 3], a detection and tracking framework is employed to extract whole object tracklets as tubelets, where temporal localization is required. In [19], an encoder-decoder network is used to generate localization masks on fixed-length clips for tubelet proposal extraction, which has varied spatial locations in different frames.

3 Method

3.1 Activity Detection Task

In this paper, we tackle the activity detection task in unconstrained videos which are untrimmed and with large field-of-views. Given an untrimmed video stream $\mathcal{V}$, the system $\mathcal{S}$ should identify a set of activity instances $\mathcal{S}(\mathcal{V}) = \{A_i\}$. Each activity instance is defined by a three-tuple $A_i = (T_i, L_i, C_i)$, referring to an activity of type C_i occurs at temporal window T_i with spatial location L_i. L_i contains

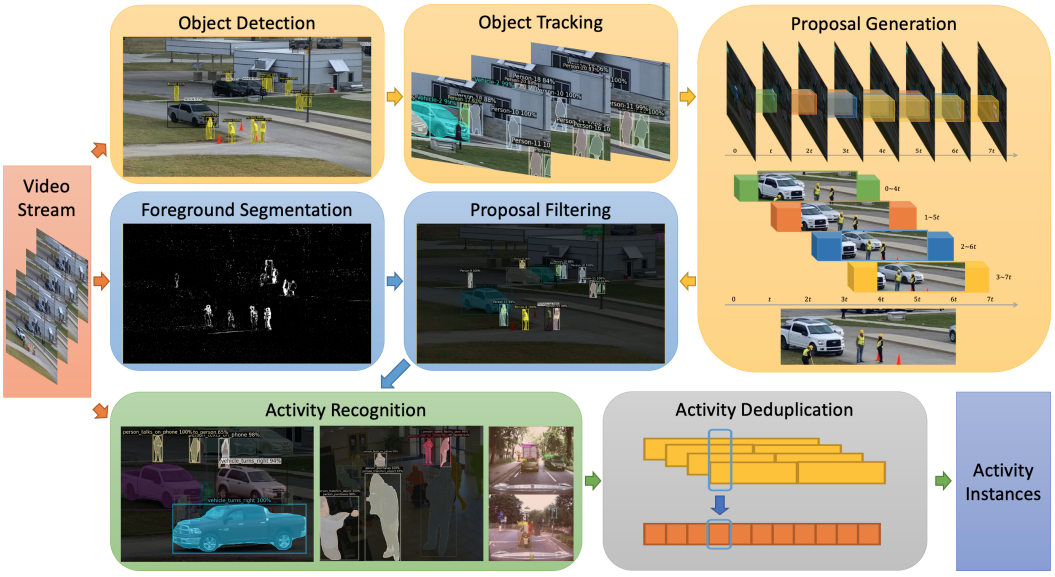

Figure 1: Architecture of *Argus++* [28]. A video stream is processed frame-by-frame through object detection and tracking to generate overlapping cube proposals. With frame-level foreground segmentation, stable proposals are filtered out. Activity recognition models determine the classification scores for each proposal. These over-sampled cubes are deduplicated to produce the final activity instances.

the precise location of A_i in each frame, forming a tube in the timeline. As such, activity detection can often be decomposed into three aspects, i.e., temporal localization (T_i), spatial localization (L_i), and action classification (C_i).

Each of the three aspects poses unique challenges to the video understanding system. Due to its multi-dimensional nature, it remains hard to define and build a useful activity detection system under the strict setting. Therefore, we also evaluates with some loosened requirements. Activity types are assumed to be either atomic activities within a temporal window (e.g. standing up) or continuous repetitive activities that can be cut into multiple identifiable windows (e.g. walking). The evaluation metric allows multiple non-overlapping predictions to be matched with one ground truth.

3.2 Argus++ System

The architecture of the introduced *Argus++* [28] system is shown in Figure 1. To tackle the task of activity detection, we adopt an intermediate concept of *spatio-temporal cube proposal* with a much simpler definition than an activity instance:

$$p_i = (x_0^i, x_1^i, y_0^i, y_1^i, t_0^i, t_1^i) \tag{1}$$

This six-tuple design relieves the localization precision and caters modern action classification models which works on fixed-length clips with fixed spatial window.

For an input video stream, the system first generates candidate proposals with frame-wise information such as detected objects, which will be covered in Section 3.3. These proposals are filtered with a background subtraction model as detailed in Section 3.4. Then, action recognition models described in Section 3.5 are applied on the proposals to predict per-class confidence scores. Finally, Section 3.6 introduces the post-processing stage to merge and filter the proposals with scores and generate final activity instances.

3.3 Proposal Generation

Starting this section, we introduce each of the components of *Argus++*. The system begins by generating a set of cube proposals. They are generated based on information from frame-level object

detection with multiple object tracking methods. Cubes are sampled densely in the timeline with refined spatial locations.

Detection and Tracking To conduct activity recognition, we first locate the candidate objects (in most cases, person and vehicle) in the video. For each selected frame F_i, we apply an object detection model to get objects $O_i = \{o_{i,j} \mid j = 1, \cdots, n_i\}$ with object types $c_{i,j}$ and bounding boxes $(x_0, x_1, y_0, y_1)_{i,j}$. Objects are detected in a stride of every S_{det} frames. A multiple object tracking algorithm is applied on the detected objects to assign track ids to each of them as $tr_{i,j}$.

Proposal Sampling To sample proposals on untrimmed videos without breaking the completeness of any activity instances, we propose a dense overlapping proposals sampling algorithm. As illustrated in Figure 2, this method ensures coverage of activities occurring at any time, with no hard boundaries. Two parameters, duration D_{prop} and stride S_{prop}, controls the sampling process. Each proposal contains a temporal window of D_{prop} frames. New proposals are generated every $S_{prop} \leq D_{prop}$ frames, possibly with overlaps. Generally, non-overlapping proposal system can be treated as a degraded case when $S_{prop} = D_{prop}$.

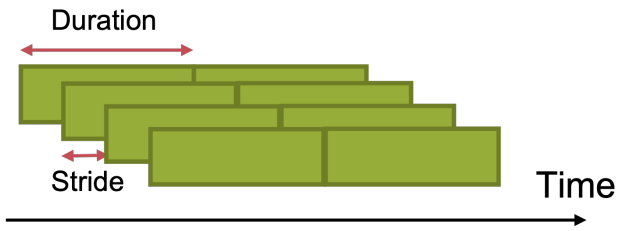

Figure 2: Dense Overlapping Proposals

Proposal Refinement To generate proposals in a temporal window from t_0 to $t_1 = t_0 + D_{prop}$, we select seed track ids Tr_{t_c} from the central frame $t_c = \lfloor \frac{t_0+t_1}{2} \rfloor$. Their bounding boxes are enlarged as the union across the temporal window

$$(x_0, x_1, y_0, y_1)_k = \bigcup_{k=1,\cdots,n_{t_c}} (\{(x_0, x_1, y_0, y_1)_{i,j} \mid t_0 \leq i \leq t_1, tr_{i,j} = tr_{t_c,k}\}) \tag{2}$$

This algorithm is robust through identity switch in the tracking algorithm as it uses the stable seeds from the central frame. It also ensures the coverage of moving objects by enlarging the bounding box when it's successfully tracked. This design is helpful for efficiency optimization by allowing a large detection stride S_{det}. When later applied for activity recognition, the bounding box can be further enlarged for a fixed rate R_{enl} to include spatial context and compensate for missed tracks.

3.4 Proposal Filtering

For now, the proposal generation pipeline applies a frame-wise object detection with slight aid of tracking information. The motion information of video is not yet explored. To produce high quality proposals, we apply a proposal filtering algorithm to eliminate the proposals that are unlikely to contain activities.

Foreground Segmentation For each proposal, a foreground segmentation algorithm is implemented to generate a binary mask for every S_{bg} frames for each video clip. We average the value of pixel masks in its cube to get its foreground score f_i. For proposals generated by object type c, those proposals with $f_i \leq F_c$ will be filtered out. The threshold F_c is determined by allowing up to P_{pos} true proposals to be filtered out.

Label Assignment To determine the above threshold and to train the activity recognition module, we need to assign labels for each generated proposal according to the ground truth activity instances. We first convert the annotation of activity instances into the cube format, denoted as ground truth cubes, by performing dense sampling of duration D_{prop} and stride S_{prop} within each instance. For

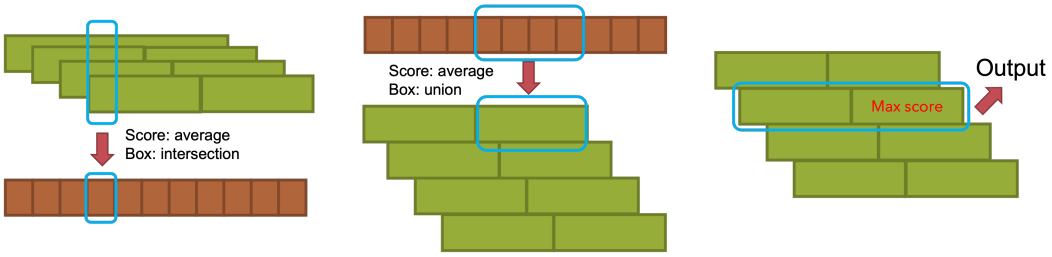

Figure 3: Deduplication Algorithm for Overlapping Proposals

each proposal, we estimate the spatial intersection-over-union (IoU) between it and ground truth cubes in the same temporal window. Then we follow Faster R-CNN [18] in the assignment process:

- For each ground truth cube, assign it to the proposal with the highest score above T_{low}.
- For each proposal, assign it with each ground truth cube with score above T_{high}.
- For each proposal, assign it as negative if all scores are below T_{low}.

T_{high} and T_{low} are the high and low thresholds. Through this algorithm, each proposal may be assigned one or more positive labels, a negative label, or nothing. Those assigned nothing are redundant detections which will not be used in classifier training.

Proposal Evaluation To measure the quality of proposals before and after the filtering, we need a method for proposal evaluation. This can be achieved by assuming a perfect classifier in the activity recognition part, so the final metrics reflects the upper bound performance with current proposals. To do this, we simply use the assigned labels as the classification outputs and pass through the deduplication algorithm covered later. To further measure other properties of the generated proposals, we can only pass through a subset of them, such as only those with spatial IoU against ground truth above 0.5.

3.5 Activity Recognition

In this section, we will elaborately introduce our action recognition modules. Given the input proposal of an activity instance p_i, our action recognition model $\mathbb{V}$ will give out the confidence vector c_i:

$$\mathbb{V}(p_i) = c_i = \{c_i^1, c_i^2, ...c_i^n\} \tag{3}$$

Where n represents the number of target actions, and $c_i \in \mathbb{R}^n$. Limited by GPU memory size and temporal length settings of pretrained weights, we need to select t frames out of $t_1^i - t_0^i$ samples from the activity instance. To do this, we strictly followed the sparse-sampling strategy mentioned in [21] for both training and inference stage. To be specific, the video is evenly separated into t segments. From each segment, 1 frame will be randomly selected to generate the sampled clip.

To transform the action recognition modules from previous multi-class task to the realm of multi-label recognition, we modified the loss function for optimization. Instead of traditional cross entropy loss (XE), we implemented a weighted binary cross entropy loss (wBCE). In which, two weight parameters are adopted, the activity-wise weight $\mathrm{W}_a = \{w_a^1, w_a^2, ..., w_a^n\}$ and the positive-negative weight $\mathrm{W}_p = \{w_p^1, w_p^2, ..., w_p^n\}$. W_a balances the training samples of different activities and W_p balances the positive and negative samples of a specific activity. With the aligned label sequence of i^{th} instance represented as $Y_i = \{y_i^1, y_i^2, ..., y_i^n\} \in \mathbb{R}^n$. The calculation of w_a^c is derived as:

$$\hat{w}_a^c = \frac{1}{\sum_{i \in [I]} y_i^c} \tag{4}$$

$$w_a^c = n \times \frac{\hat{w}_a^c}{\sum_{c \in [n]} \hat{w}_a^c} \tag{5}$$

And the derivation of w_p^c is:

$$w_p^c = \frac{\sum_{i \in [I]} \mathbf{1}_{y_i^c = 0}}{\sum_{i \in [I]} y_i^c} \tag{6}$$

In which, $[I]$ represents all input instances, and $[n]$ represent all target activities. Compared with vanilla BCE loss, we found wBCE loss can significantly improve the final performance on internal validation set.

Furthermore, we tried multiple action recognition modules and made late fusion action-wisely according to the results on the validation set. We found each classifier does show superiority on certain actions. Through the feedback from the online leaderboard, such fusion strategy can improve the final performance with noticeable margins.

3.6 Activity Deduplication

As the system generates overlapping proposals, it could have duplicate predictions for some of the proposals. This would result in a large amount of false alarms unless we deduplicate them. Figure 3 is a diagram for our deduplication algorithm which applies to each activity type with all proposals:

1. Split the overlapping cubes of duration D_{prop} and stride S_{prop} into non-overlapping cubes of duration S_{prop}. An output cube relies on all original cubes in the temporal window, with an averaged score and an intersected bounding box.
2. Merge the non-overlapping cubes of duration S_{prop} back into $\lfloor \frac{D_{prop}}{S_{prop}} \rfloor$ groups of non-overlapping cubes of duration D_{prop}. An output cube is merged from $\lfloor \frac{D_{prop}}{S_{prop}} \rfloor$ cubes with an averaged score and the union of bounding boxes.
3. Select the group where the maximum score resides.

The deduplication algorithm performs an interpolation upon the overlapping cubes. Each group in step 3 contains information from every classification results, maximizing the information utilization.

4 Experiments

4.1 Implementation Details

In the application of *Argus++* [28] at TRECVID 2021 [1], we apply Mask R-CNN [8] with a ResNet-101 [9] backbone from Detectron2 [24] pre-trained on the Microsoft COCO dataset [12] as the object detector, with $S_{det} = 8$. Only person, vehicle, and traffic light classes are selected. For the tracking algorithm, we apply the work in [22] and reuse the region-of-interest from the ResNet backbone as in [26, 15].

The proposals are generated with $D_{prop} = 64$ and $S_{prop} = 16$. The labels are assigned with $T_{high} = 0.5$ and $T_{low} = 0$. The proposal filter is set with a tolerance of $P_{pos} = 0.05$.

For activity classifiers, we adopted multiple state-of-the-art models including R(2+1)D [20], X3D [6], and Temporal Relocation Module (TRM) [16]. During training procedure, frames are cropped with jittering [21] and enlarged with $R_{enl} = 0.13$. For X3D and TRM, input frames are firstly resized to 256×256 then randomly cropped to 224×224. Backbone networks are initialized with weights pre-trained on Kinetics [10]. For R(2+1)D modules, input frames are firstly resized to 128×171 then randomly cropped to 112×112. The backbone is initialized with weights pre-trained on IG65M [7]. During validation and testing procedure, for X3D and TRM, input frames are firstly resized to 256×256 then center cropped to 224×224. For R(2+1)D, input frames are firstly resized to 128×171 then randomly cropped to 112×112. For TRECVID 2021, we trained our system only on VIRAT dataset.

4.2 Evaluation Protocols

To measure the performance, efficiency, and generalizability of *Argus++*, we evaluate it across a series of public benchmarks. *Argus++* is applied to NIST Activities in Extended Videos (ActEV) evaluations on MEVA [4] Unknown Facility , MEVA Known Facility, . For TRECVID 2021, *Argus++* is applied to VIRAT [14] settings for surveillance activity detection.

In the NIST evaluations, the metrics [2] are designed in a loosened setting, where short-duration outputs are allowed and spatial alignment is ignored. The idea was that, after processed by the system,

there will still be human reviewers to inspect the activity instances with the highest confidence scores for further usages. The performance is thus measured by the probability of miss detection (P_{miss}) of activity instances within a time limit of all positive frames plus T_{fa} of negative frames, where T_{fa} is referred to as time-based false alarm rate. The major metric, $nAUDC@0.2T_{fa}$, is an integration of P_{miss} on $T_{fa} \in [0, 0.2]$.

For metrics in the following tables, $\downarrow$ means lower is better and $\uparrow$ means higher is better. For each metric, the best value is bolded and the second best is underscored. For ongoing public evaluations, the result snapshot at 11/01/2021 is presented.

4.3 NIST TRECVID 2021 ActEV

NIST TRECVID 2021 ActEV evaluations are where only results are submitted and test data is accessible. However, following this procedure, the execution time of the whole system can't be measured. In real-world applications, real-time processing is almost a hard requirement. What's more, the reproducibility and robustness of submissions can't be verified under current protocol. Thus, we strongly suggest next year's evaluation procedure can follow the steps of ActEV'21 Sequestered Data Leaderboard.

For all our TRECVID submissions, we use the official splits of VIRAT for training and validation and no extra data is used for training. For our best submission, we fused confidence scores from different activity recognition models action-wisely. The performance of each system and the final fusion submission are shown in Figure 4. Table 1shows the leaderboard upon our submission. As is shown in the table, our system holds the first place with noticeable margins on both $nAUDC@0.2T_{fa}$ and Mean $P_{miss}@0.15T_{fa}$ metrics and ranks second on $wP_{miss}@0.15R_{fa}$ metric.

Table 1: NIST TRECVID 2021 ActEV Evaluation [1][2]

System/Team	$nAUDC@0.2T_{fa} \downarrow$	Mean $P_{miss}@0.15T_{fa} \downarrow$	Mean $wP_{miss}@0.15R_{fa} \downarrow$
Argus++ (Ours)	**0.39607**	**0.30622**	<u>0.81080</u>
BUPT	<u>0.40853</u>	<u>0.32489</u>	**0.79798**
UCF	0.43059	0.34080	0.86431
M4D	0.84658	0.79410	0.88521
TokyoTech_AIST	0.85159	0.81970	0.94897
Team UEC	0.96405	0.95035	0.95670

4.4 NIST ActEV'21 SDL Leaderboard

ActEV Sequestered Data Leaderboards (SDL) are platforms where a system is submitted to run on NIST's evaluation servers. This submission format prevents access to the test data and measures the processing time with unified hardware platform[3]. For these evaluations, *Argus++* was trained on MEVA, a large-scale surveillance video dataset with activity annotations of 37 types. We used 1946 videos in its training release drop 11 as the training set and 257 videos in its KF1 release as validation set. The optimization target is reaching better performance within 1x real-time.

Table 2 shows the published results from CVPR 2021 ActivityNet Challenge ActEV SDL Unknown Facility evaluation, where *Argus++* demonstrated around 20% advantage in $nAUDC@0.2T_{fa}$ over runner-up system.

The test set of unknown facility is captured with a different setting from MEVA, which challenges the generalization of action detection models. Table 4 shows the ongoing NIST ActEV'21 SDL Known Facility leaderboard, where *Argus++* shows over 40% advantage in $nAUDC@0.2T_{fa}$.

The test set of known facility shares a similar distribution with MEVA, where our system learns well and is getting nearer for real-world usages. Table 3 shows the ongoing NIST ActEV'21 SDL

[2]Snapshot: `https://web.archive.org/web/20211115022601/https://actev.nist.gov/trecvid21#tab_leaderboard`

[3]`https://actev.nist.gov/pub/Phase3_ActEV_2021_SDL_EvaluationPlan_20210803.pdf`

Table 2: CVPR 2021 ActivityNet Challenge[4] ActEV SDL Unknown Facility Evaluation

System/Team	$nAUDC@0.2T_{fa} \downarrow$	Mean$P_{miss}@0.02T_{fa} \downarrow$	Relative Processing Time
Argus++ (Ours)	**0.3535**	**0.5747**	0.576
UMD_JHU	<u>0.4232</u>	0.6250	0.345
IBM-Purdue	0.4238	0.6286	0.530
UCF	0.4487	<u>0.5858</u>	0.615
Visym Labs	0.4906	0.6775	0.770
MINDS_JHU	0.6343	0.7791	0.898

Table 3: NIST ActEV'21 SDL Unknown Facility Evaluation

System/Team	$nAUDC@0.2T_{fa} \downarrow$	Mean$P_{miss}@0.02T_{fa} \downarrow$	Relative Processing Time
Argus++ (Ours)	**0.3330**	<u>0.5438</u>	0.776
UCF	<u>0.3518</u>	**0.5372**	0.684
IBM-Purdue	0.3533	0.5531	0.575
Visym Labs	0.3762	0.5559	1.027
UMD	0.3898	0.5938	0.515
UMD-Columbia	0.4002	0.5975	0.520
UMCMU	0.4922	0.6861	0.614
Purdue	0.4942	0.7294	0.239
MINDS_JHU	0.6343	0.7791	0.898

Unknown Facility leaderboard continued from ActivityNet, where *Argus++* still holds the leading position with over 5% advantage in $nAUDC@0.2T_{fa}$.

4.5 Ablation Study

Coverage of Proposal Formats We analyze the coverage of dense spatio-temporal proposals and determines the best hyper-parameters for the proposal format. By directly use ground truth cubes as proposals, we estimate the upper bound performance of both overlapping and non-overlapping proposal formats on VIRAT validation set. The results are shown in Table 6, where non-overlapping proposals shows at least 6.7% systematic errors while overlapping proposals with duration 64 and stride 16 only has 1.3%.

Performance of Proposal Filtering We examine the quality of the proposals with and without the filter, as shown in Table 7 and 5. With the proposal evaluation procedure introduced in Section 3.4, the proposals are further filtered by IoU with reference and coverage of reference at levels from 0, 0.1, to 0.9 to calculate partial results.

Table 4: NIST ActEV'21 SDL[5] Known Facility Evaluation

System/Team	$nAUDC@0.2T_{fa} \downarrow$	Mean$P_{miss}@0.02T_{fa} \downarrow$	Relative Processing Time
Argus++ (Ours)	**0.1635**	**0.3424**	0.413
UCF	<u>0.2325</u>	<u>0.3793</u>	0.751
UMD	0.2628	0.4544	0.380
IBM-Purdue	0.2817	0.4942	0.631
Visym Labs	0.2835	0.4620	0.721
UMD-Columbia	0.3055	0.4716	0.516
UMCMU	0.3236	0.5297	0.464
Purdue	0.3327	0.5853	0.131
MINDS_JHU	0.4834	0.6649	0.967
BUPT-MCPRL	0.7985	0.9281	0.123

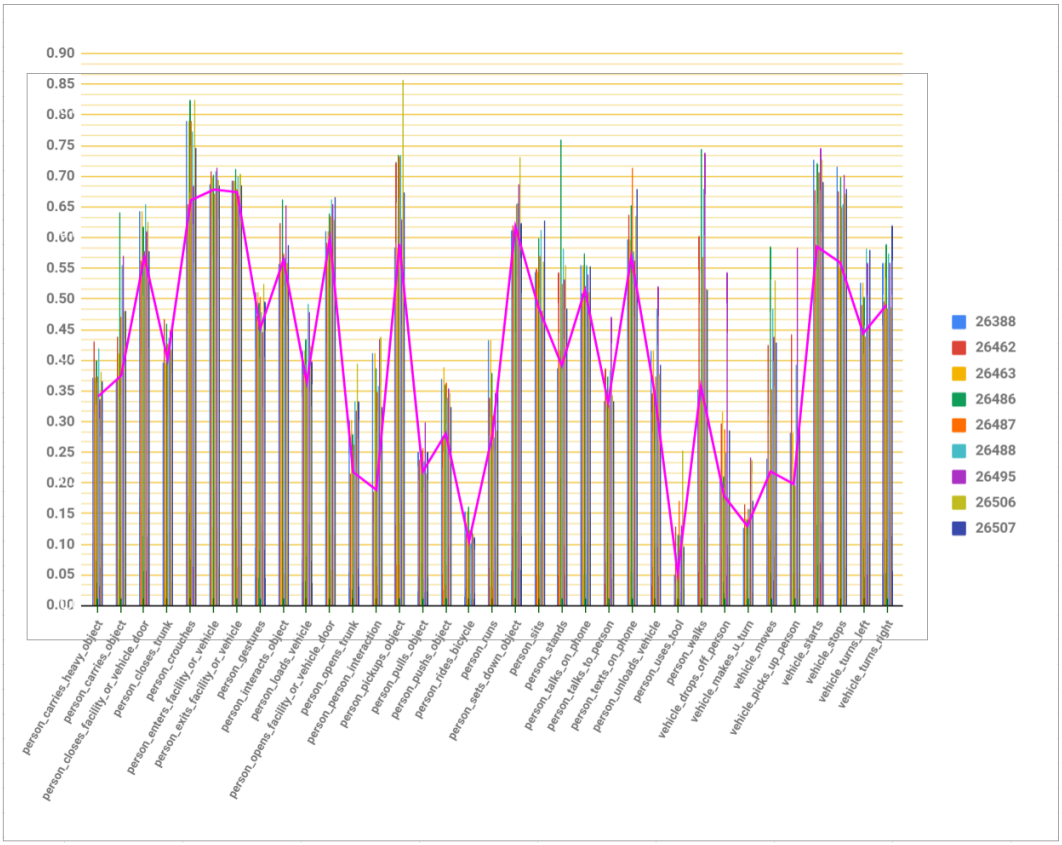

Figure 4: Action-wise nAUDC@0.2tfa of our systems. Our best submission (26562) is an action-wise fusion system (shown as the pink curve) and ranks first on the ActEV TRECVID21 Leaderboard.

Table 5: Proposal Quality Metrics on VIRAT Validation Set

$nAUDC@0.2T_{fa}$	IoU			Reference Coverage		
Threshold	Average	≥ 0	≥ 0.5	Average	≥ 0.5	≥ 0.9
Unfiltered Proposals	0.2358	0.0772	0.1518	0.1562	0.1125	0.4211
Filtered Proposals	0.2352	0.0772	0.1469	0.1563	0.1099	0.4280

Table 6: Lower Bounds of $nAUDC@0.2T_{fa}$ on VIRAT Validation Set with different proposal formats. Italic values are non-overlapping proposals while the others are overlapping proposals. Duration and stride are in the unit of frames.

Duration / Stride	16	32	64	96
32	0.0705	*0.1208*	-	-
64	**0.0127**	0.0621	*0.0673*	-
96	0.0275	0.0504	-	*0.0688*

With the dense cube proposals, the best $nAUDC@0.2T_{fa}$ we can achieve with a ideal classifier is 0.08, as indicated in the IoU ≥ 0 column. The IoU and reference coverage bounded scores are used to measure the spatial matching quality of proposals, as the $nAUDC@0.2T_{fa}$ does not consider spatial alignments. We can see that even with a condition of IoU ≥ 0.5, our proposal can achieve up to 0.15, which indicates the spatial preciseness. The proposal filter is also proved effective, which removed 70% of original proposals without dropping the recall level.

The effect of the proposal filter is also evaluate on the SDL, as shown in Table 8. It not only reduces processing time from 0.925 to 0.582, but also improves $nAUDC@0.2T_{fa}$ due to reduced false alarms.

Table 7: Statistics of Proposals on VIRAT Validation Set

Name	Unfiltered	Filtered
Number of Proposals	211271	62831
Positive rate	0.1704	**0.5204**
Rate of unique label	0.4558	0.4415
Rate of two labels	0.4127	0.4252
Rate of three labels	0.1017	0.1060

Table 8: Proposal Filter on NIST ActEV'21 SDL Unknown Facility Micro Set

Proposal Filter	$nAUDC@0.2T_{fa} \downarrow$	Processing Time
Enabled	**0.4822**	0.582
Disabled	0.5176	0.925

5 Conclusion

In this work, we introduced a latest application of the CMU *Argus++* [28] video understanding framework that achieved and kept state-of-the-art performance for years. In the system, we generated and processed a novel *overlapping spatio-temporal cube proposal* instead of tubelete proposal used by previous works. By over-sampling, we generated cube proposal while ensured the coverage and completeness of activities. Then, we applied a proposal filtering to select the most important candidate proposals. After that, we applied activity recognition and activity deduplication to classify the target activities in the proposals. The proposed system is able to process streaming videos in real-time for varies scenarios in large field-of-views and achieved the best performance on the TRECVID 2021 challenge up to now.

There are already impressive works extended from this system to other area, including UAV video and road video analysis and archived the outperformed results. Future works could be done on widely real-world applications, such as first-person human view activity understanding, vision based self-driving, etc.

6 Acknowledgements

This research is supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00340. This research is supported in part through the financial assistance award 60NANB17D156 from U.S. Department of Commerce, National Institute of Standards and Technology. This project is funded in part by Carnegie Mellon University's Mobility21 National University Transportation Center, which is sponsored by the US Department of Transportation.

References

[1] George Awad, Asad A. Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas Diduch, Jeffrey Liu, Yvette Graham, Gareth J. F. Jones, , and Georges Quénot. Evaluating multiple video understanding and retrieval tasks at trecvid 2021. In *Proceedings of TRECVID 2021*. NIST, USA, 2021.

[2] George Awad, Asad A. Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth J. F. Jones, Wessel Kraaij, and Georges Quenot. TRECVID 2020: A comprehensive campaign for evaluating video retrieval tasks across multiple application domains. *arXiv:2104.13473 [cs]*, Apr. 2021.

[3] Xiaojun Chang, Wenhe Liu, Po-Yao Huang, Changlin Li, Fengda Zhu, Mingfei Han, Mingjie Li, Mengyuan Ma, Siyi Hu, and Guoliang Kang. MMVG-INF-Etrol@ TRECVID 2019: Activities in Extended Video. In *TREC Video Retrieval Evaluation, TRECVID*, 2019.

[4] Kellie Corona, Katie Osterdahl, Roderic Collins, and Anthony Hoogs. MEVA: A Large-Scale Multiview, Multimodal Video Dataset for Activity Detection. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1059–1067, Waikoloa, HI, USA, Jan. 2021. IEEE.

[5] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[6] Christoph Feichtenhofer. X3D: Expanding Architectures for Efficient Video Recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 200–210, Seattle, WA, USA, June 2020. IEEE.

[7] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-Scale Weakly-Supervised Pre-Training for Video Action Recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12038–12047, Long Beach, CA, USA, June 2019. IEEE.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):386–397, Feb. 2020.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE.

[10] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics Human Action Video Dataset. *arXiv:1705.06950 [cs]*, May 2017.

[11] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal Shift Module for Efficient Video Understanding. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7082–7092, Seoul, Korea (South), Oct. 2019. IEEE.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[13] Wenhe Liu, Guoliang Kang, Po-Yao Huang, Xiaojun Chang, Lijun Yu, Yijun Qian, Junwei Liang, Liangke Gui, Jing Wen, Peng Chen, and Alexander G. Hauptmann. Argus: Efficient Activity Detection System for Extended Video Analysis. In *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 126–133, Snowmass Village, CO, USA, Mar. 2020. IEEE.

[14] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, J. K. Aggarwal, Hyungtae Lee, Larry Davis, Eran Swears, Xioyang Wang, Qiang Ji, Kishore Reddy, Mubarak Shah, Carl Vondrick, Hamed Pirsiavash, Deva Ramanan, Jenny Yuen, Antonio Torralba, Bi Song, Anesco Fong, Amit Roy-Chowdhury, and Mita Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*, pages 3153–3160, June 2011.

[15] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann. ELECTRICITY: An Efficient Multi-camera Vehicle Tracking System for Intelligent City. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2511–2519, Seattle, WA, USA, June 2020. IEEE.

[16] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann. Trm: Temporal relocation module for video recognition. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, 2022.

[17] Yijun Qian, Lijun Yu, Wenhe Liu, Guoliang Kang, and Alexander G. Hauptmann. Adaptive Feature Aggregation for Video Object Detection. In *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 143–147, Snowmass Village, CO, USA, Mar. 2020. IEEE.

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[19] Mamshad Nayeem Rizve, Ugur Demir, Praveen Tirupattur, Aayush Jung Rana, Kevin Duarte, Ishan R Dave, Yogesh S Rawat, and Mubarak Shah. Gabriella: An Online System for Real-Time Activity Detection in Untrimmed Security Videos. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4237–4244, Jan. 2021.

[20] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, Salt Lake City, UT, June 2018. IEEE.

[21] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 20–36, Cham, 2016. Springer International Publishing.

[22] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards Real-Time Multi-Object Tracking. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12356, pages 107–122. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science.

[23] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[24] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[25] Lijun Yu, Peng Chen, Wenhe Liu, Guoliang Kang, and Alexander G. Hauptmann. Training-free Monocular 3D Event Detection System for Traffic Surveillance. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 3838–3843, Dec. 2019.

[26] Lijun Yu, Qianyu Feng, Yijun Qian, Wenhe Liu, and Alexander G. Hauptmann. Zero-VIRUS * : Zero-shot Vehicle Route Understanding System for Intelligent Transportation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2534–2543, Seattle, WA, USA, June 2020. IEEE.

[27] Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G Hauptmann. CMU Informedia at TRECVID 2020: Activity Detection with Dense Spatio-temporal Proposals. In *TREC Video Retrieval Evaluation, TRECVID*, page 9, 2020.

[28] Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G. Hauptmann. Argus++: Robust real-time activity detection for unconstrained video streams with overlapping cube proposals. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, 2022.

[29] Lijun Yu, Dawei Zhang, Xiangqun Chen, and Alexander Hauptmann. Traffic Danger Recognition With Surveillance Cameras Without Training Data. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, Nov. 2018.

[30] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-Guided Feature Aggregation for Video Object Detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 408–417, Venice, Oct. 2017. IEEE.