GabriellaV2: Towards better generalization in surveillance videos for Action Detection

Ishan Dave Zacchaeus Scheffer Akash Kumar Sarah Shiraz Yogesh Singh Rawat
Mubarak Shah
Center for Research in Computer Vision, University of Central Florida
{ishandave, zaccy, akash_k, sarah.shiraz}@knights.ucf.edu, {yogesh, shah}@crcv.ucf.edu

Abstract

Activity detection has wide-reaching applications in video surveillance, sports, and behavior analysis. The existing literature in activity detection has mainly focused on benchmarks like AVA, AVA-Kinetics, UCF101-24, and JHMDB-21. However, these datasets fail to address all issues of real-world surveillance camera videos like untrimmed nature, tiny actor bounding boxes, multi-label nature of the actions, etc. In this work, we propose a real-time, online, action detection system which can generalize robustly on any unknown facility surveillance videos. Our real-time system mainly consists of tracklet generation, tracklet activity classification, and prediction refinement using the proposed post-processing algorithm. We tackle the challenging nature of action classification problem in various aspects like handling the class-imbalance training using PLM method and learning multi-label action correlations using LSEP loss. In order to improve the computational efficiency of the system, we utilize knowledge distillation. Our approach gets state-of-the-art performance on ActEV-SDL UF-full dataset and second place in TRECVID 2021 ActEV challenge. Project Webpage: www.crcv.ucf.edu/research/projects/gabriellav2/

1. Introduction

The problem of video understanding has wide-reaching applications like action recognition [1–4], action detection [5–9], temporal action localization [10, 11], and video synthesis [12, 13].

The task of spatio-temporal activity localization involves detecting the actions present in the videos, and generating a spatial bounding box that tracks the activities over time. The main two problem statements involving videos are: *Can we recognize the action in the video?* and *If so, can we say where the activity is happening?* The first problem is termed as video classification, which involves labeling single or multiple simultaneous activities present in a video. The second problem targets annotating *where* the activity is happening. This is referred as the task of spatio-temporal activity localization.

The majority of works [14–18] on action detection focus on benchmark datasets like AVA [19], AVA-Kinetics [20], UCF101-24 [21] or J-HMDB [22]. These approaches are not suitable for real-world surveillance video due to several reasons: (1) actor size of the surveillance camera is tiny compared to the actor-centric videos of the benchmarks, (2) surveillance videos are untrimmed, unlike the 3 second trimmed videos of AVA [19] and AVA-Kinetics [20], and (3) real-time and online approach is required for the video surveillance.

Prior works [6, 9, 23–30] present approaches for action detection in surveillance video. One of the best performing systems from the prior works is our prior system, Gabriella [6], which is a real-time, online, action detection approach. Gabriella adopts an end-to-end approach by first detecting the action proposal using a pixel-wise localization module which is followed by action classification and post-processing. Although this system outperforms most of the concurrent systems, it has two main limitations: (1) it merges overlapping actor bounding boxes, which results in huge regions for indoor scene and degrades performance of action classification stage, and (2) localization network does not generalize well on the unknown scene/facility camera, which results in a high probability of missing actions.

In this work, we build upon our previous system, Gabriella, to improve the system overall performance and generalization capability in unknown facility cameras. Firstly, in order to avoid merging in crowded scenes we replace the pixel-wise localization network with the object detector and tracker to get actor-centric trackelets. Secondly, we strengthen the action classification unit by utilizing state-of-the-art multi-label class-imbalance training, partial label masking (PLM), and learning class-correlation through log-sum-exp pairwise (LSEP) loss. We also utilize knowledge distillation to make the action classification component more computationally efficient. Our system achieves state-of-the-art performance on MEVA [31] ActEV-SDL UF-Full and places second in VIRAT TRECVID ActEV 2021 challenge.

2. Related Works

Spatio-Temporal Activity Localization: The task of recognizing and localizing actions across frames in videos is termed as spatio-temporal activity localization. Primitive works took inspiration from images and 2D models and extended such approaches to frames. With the introduction of 3D convolutions, most of the works shifted from 2D-CNN backbones [32–34] to 3D-CNN [35–37]. The main limitation of the prior works is that they have been trained and tested mostly on trimmed datasets such as UCF101-24 [38], JHMDB-21 [22] or AVA [19]. In the real-world, we deal with untrimmed videos. In the literature, only a few large-scale datasets have been created to tackle this problem [39–41]. ActEV UF-Full and TrecVID utilize the MEVA dataset and VIRAT [42] datasets respectively to develop more works on untrimmed videos for the spatio-temporal localization task. What makes these datasets challenging, is the average length of videos, which is 20 to 30 times that of previously proposed datasets. The mains problem solved on untrimmed datasets is to approximate where the activity is happening in the temporal dimension and detect the type of action being localized. Also, the solutions are not always real-time, which is a critical aspect for security surveillance videos. In our work, we develop a real-time spatio-temporal localization framework to detect actions in these long untrimmed videos.

Post-processing: In general, raw output of object detection algorithm can't be used as a finalized localization map. It contains a lot of false positives indicating multiple instances of a single object. These multiple instances needs to be suppressed to generate a single instance per object detected. There have been works [43–45] to tackle this issue utilizing Non-Maximum threshold in parallel to object detection approaches. T-CNN [46] imposes high confidence score based on contextual information. [43], [44] and [45] uses temporal overlap scores of bounding box across frames. This approaches are mostly limited to ImageNetVID [47] dataset. Since, most of the datasets are trimmed, the problem of false alarms have mostly been looked over spatially across frames. On the other hand, in an untrimmed video, multiple actions have an abrupt starting and ending time. Thus, we extend these approaches to spatio-temporal dimension. We target multiple detection on a frame (spatially), and, extend those detections across multiple frames (temporal) suppressing the false alarm detections. However, we use tracking ids of proposals instead of object detections per frame. We also monitor the classification score of detections over time. This procedure not only helps us to link detections efficiently, it also suppresses the contrastive fine-grained activities such as *person standing up* versus *person sitting down*.

3. Method

An overview of our system is depicted in Figure 1. Details of each component of our system is given in this section.

3.1. Tracklet Generation

For tracklet generation, we first detect actors (person and vehicle) in the frames of a clip using YOLOv5 object detector [48]. YOLOv5 is an optimized implementation of YOLO [49] single stage object detection framework using combination of universal features like Weighted-Residual-Connections (WRC), Cross-Stage-Partial-connections (CSP), Cross mini-Batch Normalization (CmBN), Self-adversarial-training (SAT) and Mish-activation with Mosaic data augmentation, DropBlock regularization, and CIoU loss. The detected regions are provided to mixture of gaussian (MoG) background subtractor to remove relatively static objects. The filtered detected actor bounding boxes are joined based on a simple IoU based criterion using SORT tracker [50]. Each tracklet coordinates are stored in the memory with an object id which is carried forward to the next trimmed clip to track an object through different clips.

3.2. Activity Classification

After getting the tracklet from the tracker, it is downsampled to a fixed size and sent to the action classifier.

3.2.1 Baseline Action Classification

Since multiple actions can be present at a time for an actor tracklet, we formulate our baseline action classification approach as multi-label classification problem with each prediction independent of each other. To train the baseline action classifier, input tubelets are extracted directly from the ground-truth annotations. Apart from action tubelets, we also provide background tubelets (i.e. no spatio-temporal overlap with the ground-truth actions) to the action classifier which results in a total of $C + 1$ classes, where C is the number of activities present in the annotations. We utilize various 3D-CNN backbone to get spatio-temporal features and apply a linear classification layer followed by sigmoid activation function. The baseline classifier is trained using the BCE loss as shown in Equation 1.

$$\mathcal{L}_{\text{BCE}}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)\right], \quad (1)$$

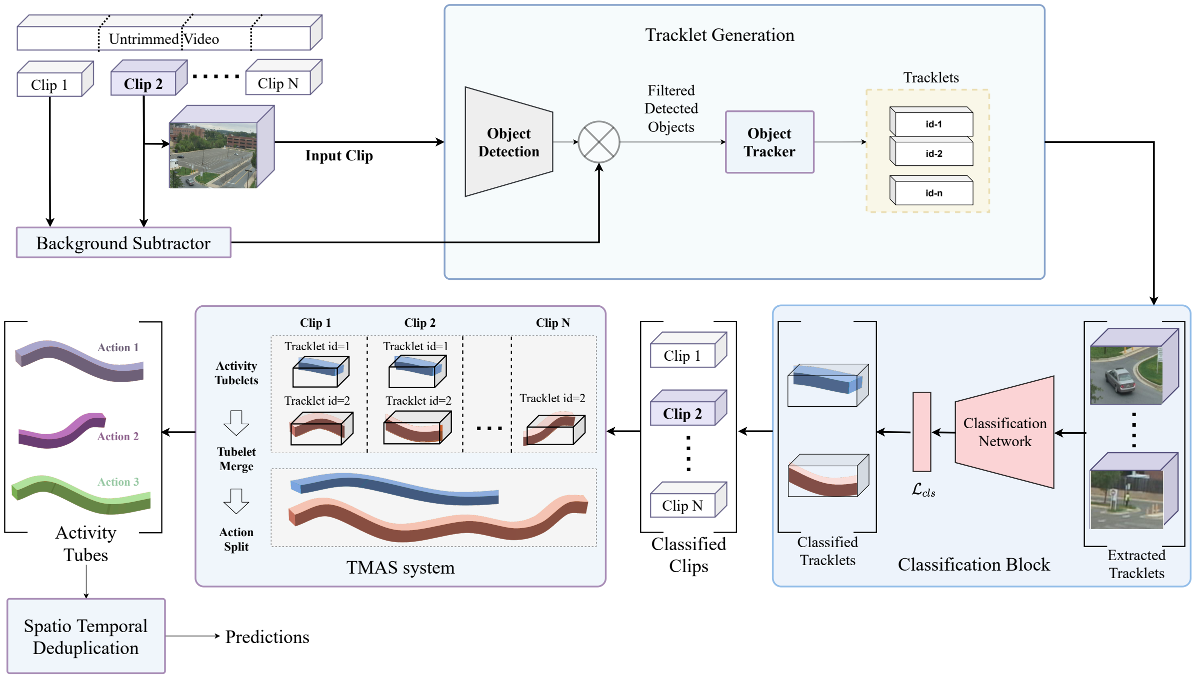

Figure 1: Schematic Diagram for UCF DIVA system: Firstly, an untrimmed video is divided into fixed temporal sized clips, which are then passed to the object detector to detect the actors frame-wise. The actor bounding boxes in different frames of the clip are then joined using a tracker to get tracklets. The action classifier predicts actions classes on each tracklet, which are then post-processed through the proposed post-processing algorithm.

where N is batchsize, $y_i \in \{0, 1\}$ is the target label , $\hat{y}_i \in [0, 1]$ is predicted output.

3.2.2 Adapting the baseline for the generated tracklets

The object detector in the inference pipeline gets a large number of actors which are not participating in any action; this results in increased number of false positives. This problem arises because the action classifier is not trained on any actor-centric background tubes from the ground-truth. With this motivation, we train the action classifier with the tracklets extracted from YOLO object detector and action labels built from the spatio-temporal overlap with the ground-truth. The schematic of the baseline adaptation for the generated tracklets is shown in Figure 2 .

3.2.3 Class balanced training

The MEVA dataset and VIRAT dataset have a large class-imbalance due to inherent nature of actions. For example, talking is more common action than stealing. The vanilla BCE loss of Equation 1 provides equal weight to each activity class regardless of the number of samples. The tail-classes have $0.001\times$ sample size of the head class, which results in low performance in the tail classes compared to the head classes. In order to handle the class-imbalance we opt of recent multi-class re-weighting scheme PLM [51]. The method balances the positive to negative ratio for each class by randomly masking the training labels in the loss computation.

3.2.4 Learning multi-label correlations

Equation 1 treats each action class independently which fails in exploiting the inherent cooccurance of the multiple action classes. For example, talking and standing heavily cooccur in the VIRAT dataset. In order to exploit the class correlations we use Log Sum Exp Pairwise (LSEP) loss [52], which is a ranking type of loss introduced as a baseline solution to learn the multi-label actions dependencies in Multi-Moments in Time dataset [53]. The LSEP loss is modified in a way to ignore loss computation for the activity instances having ambiguous spatio-temporal overlap with the ground-truth annotations. The original implementation of the LSEP loss is based on the BCE loss, in our use case we implemented the LSEP loss on the foundation of PLM loss to handle the class-imbalance problem as well.

3.2.5 Knowledge Distillation

Hinton et al. [54] proposed a technique to pass over the “dark knowledge” learned by a neural network to another network of different capacity. This Knowledge Distillation method can be used for model compression and learning

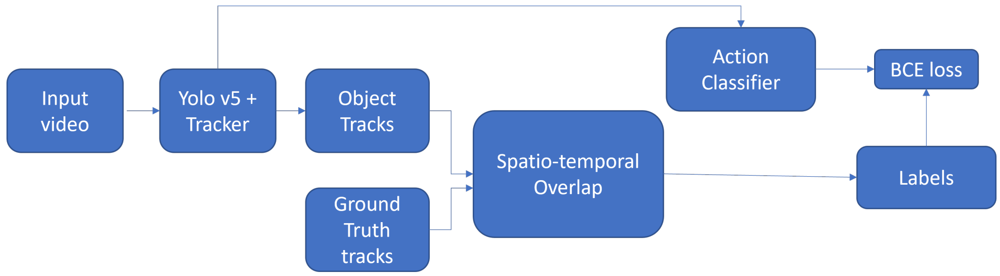

Figure 2: Action classifier training from tracklets obtained using object detector and tracker

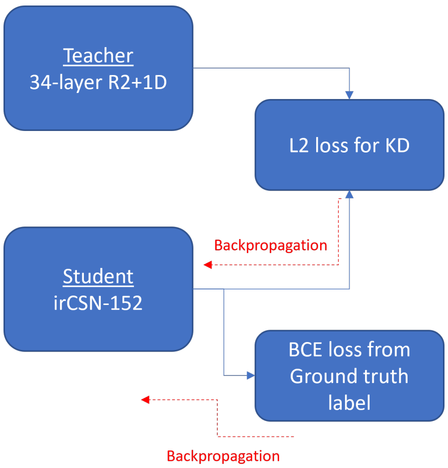

Figure 3: Distillation from heavy backbone

multi-label action dependencies. We use knowledge distillation to improve the compute efficiency of the system in 2 ways: (1) higher capacity, compute intensive networks are distilled into lower capacity, lower compute networks (Figure 3) , and (2) ensembles of distilled networks require fewer models to achieve similar performance. For the knowledge distillation training setup we use L2-loss as the distillation loss from distance between teacher and students predictions. More details on model selection and computation savings is described in Section 5.4.

3.3. Post Processing

The first part of our post-processing algorithm is to use a Tracklet Merge, Action Split (TMAS) algorithm, which turns class-wise tracklet predictions into action tubes. This is followed by Non-Maximum Suppression (NMS).

3.3.1 Tracklet Merge

The first task in post processing is to merge consecutive tracklets with the same object id as determined by the SORT tracker. These merged tracklets form "actor tracks", so termed because the object detected by the YOLO model is a physical object, while we are concerned with the activity in which the actor engages. Because the tracklets are generated using a sliding window, the score for a given tracklet is given to the first half of the frames covered by the corresponding sliding window.

3.3.2 Action Split

Once actor tracks have been obtained, we traverse each actor track, applying a sliding-window average to each class. Then, for each frame, if the hard_negative class exceeds the 0.8 background threshold, we discard all predictions of the actor track at that time. Otherwise, we create "action tubes" from each class that exceeds the 0.05 foreground threshold at a given time. Two class-wise scores of the same class, A, on the same actor track (but at different times) are contained in the same action tube if and only if all consecutive frames between them obtain a class score for A over the foreground threshold, and a hard_negative score below the background threshold. If two predictions are not on the same actor track, or not of the same class, they will never be on the same action tube. A diagram of the full TMAS algorithm is given in Figure 1.

3.3.3 NMS Deduplication

The object detection system faces an issues of overlapping actor tracks as shown in Figure 4, in addition to multi-actor actions. Both of these issues can cause multiple actor tracks to include the same action.

To solve this problem, we perform class-wise Non-Maximum Suppression (NMS) using an IoU threshold to remove many of the duplicates. This is done for each frame and each class. To perform NMS for a given frame and given class, we first make a list of all action-tube bounding boxes in that frame of the given class. Then, we remove from that list, the bounding box with the highest class confidence, and additionally remove all bounding boxes with sufficient IoU overlap. The bounding boxes removed because of IoU overlap are removed from their corresponding tubes entirely. This is repeated until the original list is empty. If a frame in the middle of an action tube is removed, all frames after the removed frame are moved to a new ac-

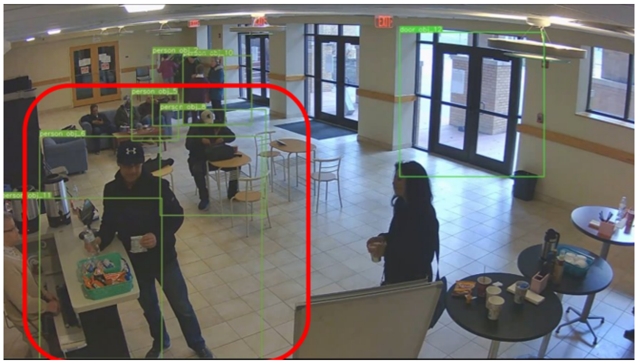

Figure 4: Example of duplicate instance in the predictions. Green boxes show the square form of the detected objects. In the red box we have a bigger tube (left) covering *purchasing* and *reading* activity, however, the overlapping tube on the right outputs *reading* activity at the same time, which creates a false alarm for *reading* activity.

tion track. Otherwise, the frame is at the beginning or end, and it is removed with no extra steps.

4. Experiments

4.1. Implementation Details

Dataset: The videos we use are taken at 30fps, and we consider only every other frame by using a skip rate of 2 (so 16 fps effective) everywhere except as noted in training. **ActEV SDL21** contains the `UF116hr-R13` subset of MEVA videos. **TRECVID-2021 ActEV** data contains VIRAT videos with split provided on https://actev.nist.gov/trecvid21#tab_data

Tracklet generation: We generate bounding boxes every 8^{th} frame and use a YOLOv5x model pretrained on MS-COCO [55] dataset. SORT tracker is used with a memory of 1 detection instance i.e. 8 frames with an IoU threshold of 0.25.

Action Classification: The cropped tracklet is linearly down-sampled to a $16\times112\times112$ fixed size as an input the action classifier. All action classifiers are pretrained on Kinetics-400 [56] dataset and finetuned using a base learning rate of 1e-4 with Adam optimizer. A cosine annealing learning rate is used with a linear warm-up upto 5 epochs.

4.2. System Evaluation

Firstly, we explain the performance measurement for the evaluation protocols and then we show results on ActEV-SDL 21 [57] and TRECVID-21 ActEV evaluation protocols.

4.2.1 Performance Metrics

In the evaluation protocol, we consider the relative processing time, Pmiss@Xtfa, and nAUDC@Xtfa. The relative processing time is computed as the time required to process a video on four NVIDIA 1080Ti's divided by the video's running time. Pmiss is the ratio of activities where the system did not detect the activity for at least one second. TFA (tfa) refers to the time-based false alarm rate, i.e. the portion of the time that the system detected an activity when, in fact, there was none. A detection is determined as being "present" or not based on a confidence threshold, so Pmiss and tfa are functions of this confidence threshold, c.

$$Pmiss = Pmiss(c) \tag{2}$$

$$tfa = tfa(c) \tag{3}$$

To obtain a Pmiss@0.02tfa score, we calculate the Pmiss and tfa at multiple confidences until a 0.02 tfa is obtained, and take the corresponding Pmiss score. Notice that tfa and Pmiss are both monotone in the confidence threshold. Therefore Pmiss as a function of tfa is well defined, except for possible vertical jumps (in which case, we use lower Pmiss).

$$Pmiss@0.02tfa = Pmiss(tfa^{-1}(0.02)) \tag{4}$$

The above process of checking various confidence thresholds gives a relationship between Pmiss and tfa. We can compute the nAUDC@Xtfa as

$$nAUDC@Xtfa = \int_0^X Pmiss(tfa^{-1}(f))df \tag{5}$$

where we are integrating over the false alarm rate.

4.3. Comparison with other teams

We evaluate our system on 2 protocols: ActEV-SDL Unknown Facility Full set and TRECVID 2021 ActEV protocol. As shown in Table 1 our system is the best performing system among other teams in terms of mean pmiss and second best system in nAUDC@0.2tfa metric. We use pmiss@0.02tfa for ranking in Table 1, as it is the primary performance measurement for DIVA program. Table 2 shows our system gets second best performance among all participants of TRECVID 2021.

Trade-off between processing time and performance for different teams is shown in Fig. 6. Our system achieves the best trade-off among other teams.

Generalization from Known to Unknown Facility camera is shown in Fig. 5. We report the difference between the best systems of known and unknown facility for each team to evaluate the generalization capability. Our system gets consistently lower performance drop across activities, which shows superiority of our system in terms of generalization compared to other teams.

Rank	Team Name	sub_id	mean p_miss@0.01tfa	mean p_miss@0.02tfa	mean nAUDC@0.2tfa	relative_ processing_time
1	UCF	25908	0.62	0.5372	0.3518	0.6840
2	CMU-DIVA	26095	0.65	0.5438	0.3330	0.7760
3	IBM-Purdue	26113	0.65	0.5531	0.3533	0.5750
4	UMD	26619	0.68	0.5938	0.3898	0.5150
5	UMD-Columbia	25031	0.68	0.5975	0.4002	0.5200
6	UMCMU	25576	0.75	0.6861	0.4922	0.6140
7	Purdue	25782	0.80	0.7294	0.4942	0.2390
8	MINDS_JHU	24666	0.84	0.7791	0.6343	0.8980

Table 1: ActEV-SDL Unknown Facility Full-set leaderboard. Best and second best scores are highlighted.

Rank	team_name	team_abbrev	nAUDC@tfa0.2	p_miss@tfa0.15
1	BUPT-MCPRL	BUPT-MC_26542	0.4085	0.3249
2	UCF	UCF_26546	0.4306	0.3408
3	INF	INF_26532	0.4444	0.3508
4	M4D_2021	M4D_202_26467	0.8466	0.7941
5	TokyoTech_AIST	TOKYOTE_26508	0.8516	0.8197
6	Team UEC	TEAMUE_26530	0.9640	0.9503

Table 2: Official results for TRECVID 2021 ActEV challenge. Best and second best scores are highlighted.

4.4. Progress over the time

The overall progress of our system over time is summarized in Table 3. Our system's final performance on ActEV's Sequestered Data Leader board (SDL) Unknown Facility micro set is summarized in Table 1. Overall, we have improved over **11%** in nAUDC and **8%** in Pmiss compare to GabriellaV1 system [6]. Qualitative result videos can be found on our project webpage. [1]

5. Ablations

5.1. Activity recall

An ideal action localization model is expected to have 100% recall. To evaluate the performance of actor tracklets, we use activity recall at different spatio-temporal overlap of the ground-truth annotation as shown in Figure 7. From bounding box visualization of the output, we observe that 80% spatio-temporal overlap with the ground-truth annotation is the best point for recall measurement. GabriellaV1 system using I3D based localization model [6] gets an average recall of 0.65 at 80% overlap whereas our proposed method gets recall of 0.87, which is **22%** higher than that of [6].

[1] www.crcv.ucf.edu/research/projects/gabriellav2/

5.2. Action classification: Architectures

We use various 3D-CNN architectures as backbone of action classifier task. The performance of each architecture with the baseline training scheme is shown in Table 4. To evaluate the performance of action classification task, we use macro average metrics like macro-mAP, macro-recall, and background-precision on the validation set. We also report number of parameter and inference cost for a single batch to measure the compute efficiency. We observe that R2+1D-34 layer architecture performs the best in all metrics, however, at high inference computation cost.

5.3. Action classification: Training losses

We train the action classifier using different training schemes using different losses like BCE, Softmax+CrossEntropy, PLM [51], LSEP [52], LSEP based on PLM, Multilabel Margin Loss and BCE loss using label smoothing. All of the experiments are performed on R2+1D-34 layer model, shown in Table 5. Our first observation is that training with Softmax activation with a high temperature leads to lower recall(-10%) and higher background precision(+20%), which provides us a very different model than the baseline and provides diversity in ensemble. Secondly, LSEP based on PLM loss works best among all training losses, which shows importance of resolving class-imbalance and learning class-correlations in activity detection problem. Thirdly, introducing label smoothing greatly improves recall(+5%) of the baseline.

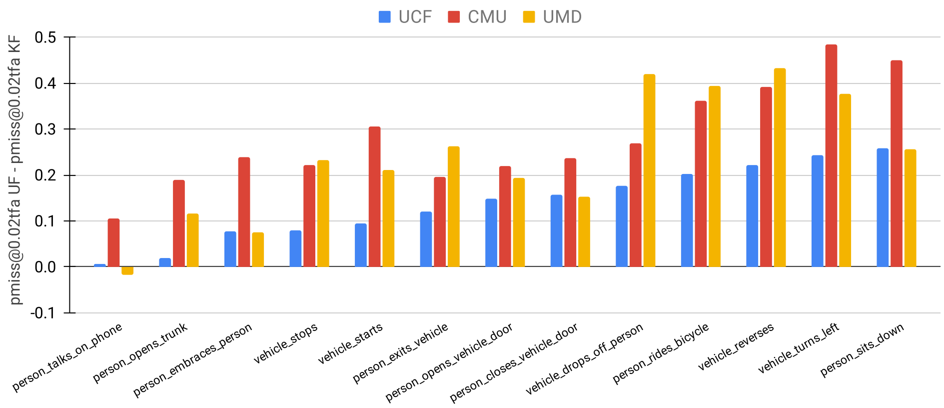

Figure 5: **Activity-wise generalization from Known Facility (KF) to Unknown Facility (UF) camera** of MEVA SDL test set. Lower value indicates better generalization. Our system gets the minimum drop in performance while generalizing from known to unknown facility camera in comparison of the other teams.

Date	System Details	nAUDC@0.2tfa	PMiss@0.02tfa
2021-04-30	GabriellaV1 system [6]	0.497	0.647
2021-05-03	New pipeline(GabriellaV2)	0.5198 $\downarrow$ 2.28%	0.6693 $\downarrow$ 2.25%
2021-05-11	Ensemble (Yolo+ ResNet18 and 34)	0.4941 $\uparrow$ 0.29%	0.6411 $\uparrow$ 0.59%
2021-05-13	Memory Optimization (Mixed Precision) Bigger Classifier Ensemble	0.4775 $\uparrow$ 1.95%	0.6396 $\uparrow$ 0.74%
2021-05-14	Ensemble (include PIP dataset)	0.4684 $\uparrow$ 2.86%	0.6438 $\uparrow$ 0.32%
2021-05-16	Spatio-Temporal Deduplication	0.4458 $\uparrow$ 5.12%	0.6010 $\uparrow$ 4.60%
2021-06-26	Post-Processing tuning: Sigmoid threshold and Deduplication threshold	0.3973 $\uparrow$ 9.97%	0.6041 $\uparrow$ 4.29%
2021-07-23	MEVA drop11 annotations	0.3791 $\uparrow$ 11.79%	0.5760 $\uparrow$ 7.10%
2021-07-31	Knowledge Distillation	0.3869 $\uparrow$ 11.01%	0.5634 $\uparrow$ 8.36%

Table 3: Progress of our system over time on SDL-UF micro leaderboard $\uparrow$% or $\downarrow$% indicates absolute change with respect to GabriellaV1 baseline

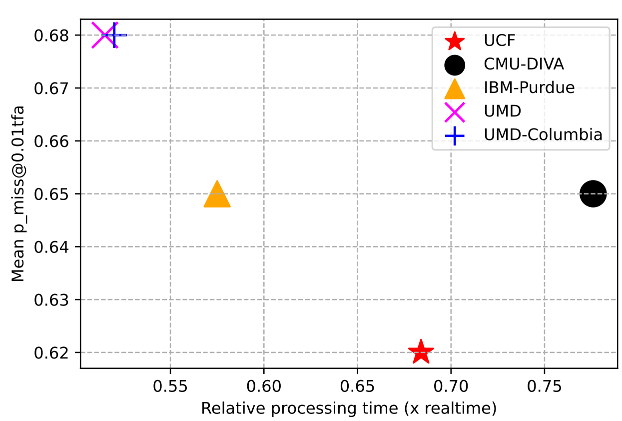

Figure 6: Trade-off between pmiss@0.01tfa vs Relative processing time for UF-Full leaderboard.

5.4. Knowledge Distillation

Distillation from heavier backbone teacher As seen from Table 4, we observe that the R2+1D-34 performs best however with the higher computation cost, whereas irCSN-152 model performs 8% worse at 2$\times$ lower computation cost. The first goal of the knowledge distillation is to distill the knowledge of higher capacity R2+1D-34 layer model to the lighter irCSN-152 model. The results of such knowledge distillation scheme is shown in Table 7. The distilled student model not only performs at lower computation cost but also able to outperform the teacher performance. Our conjecture is that, this is due to the fact of distillation loss (L2-loss) helps in learning multi-label class correlations.

Distillation from multiple teachers Our system can fit 5 action classifiers which were differently trained eg. training on BCE loss, PLM loss, LSEP loss, or training on different set of annotations like Kitware-UMD or PIP dataset. We want to distill knowledge of all these 5 classifiers into a single classifier to reduce ensemble cost, and this saved computation budget can be used to accommodate include more classifiers. The results of learning from multiple teachers are shown in Table 7. The student model improves **4%**

Architecture	mAP	Recall	BG-Prec	Params(M)	Inf Cost (MB)
R2+1D-18 [58]	46.2	0.826	0.601	31.5	809.7
R2+1D-34 [59]	51.1	0.879	0.701	63.5	1319.6
SlowOnly- R3D50 [60]	48.9	0.801	0.695	32.5	756.6
ir-CSN152 [61]	44.4	0.841	0.602	29.0	613.7
Wide-ResNet-50 [1]	44.1	0.759	0.417	157.5	567.5

Table 4: Action classification performance for various 3D-CNN backbones

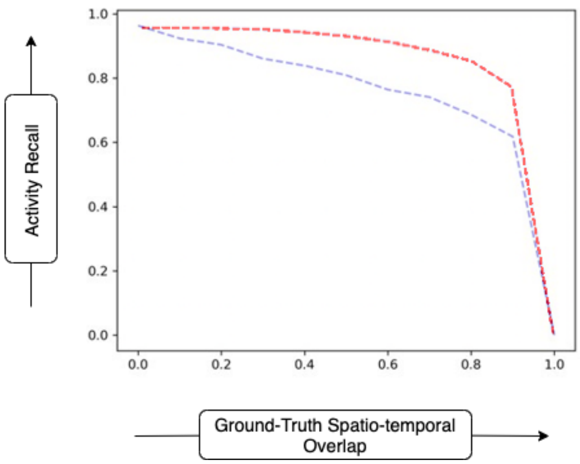

Figure 7: Activity Recall at different %overlap of the groundtruth. Red curve shows proposed method whereas the blue curve shows the performance of localization model of GabriellaV1 system [6]

Training Loss	mAP	Recall	BG-Prec
Binary Cross Entropy	51.1	0.879	0.701
Softmax + Cross Entropy	50.9	0.762	0.901
PLM + LSEP	52.0	0.911	0.673
LSEP	51.4	0.884	0.679
PLM	51.2	0.852	0.725
Multilabel Margin	49.7	0.853	0.601
Label Smoothing	51.7	0.922	0.751

Table 5: Action classification performance for various training losses

Model	mAP	Recall	BG-Prec
Teacher R2+1D-34	51.1	0.879	0.701
Student CSN-152	44.4	0.841	0.602
Student CSN-152 (distilled)	51.6	0.883	0.760

Table 6: Results of knowledge distillation from higher to lower capacity (lighter) model

from its baseline, however, it gets slightly worse (1%) performance compared to ensemble of 5 classifiers. We believe it can be improved by changing the weightage of L2-loss and BCE loss, which is equal in the existing setup.

Model	mAP	Recall	BG-Prec
Teacher Ensemble	56.4	0.943	0.861
Student	51.1	0.879	0.701
Student (distilled)	55.3 (+4.2%)	0.929	0.887

Table 7: Results of knowledge distillation from multiple teachers

6. Conclusion

In this paper, we propose GabriellaV2, a real-time system to detect activities from untrimmed surveillance videos. Our system is based on tracklet generation using state-of-the-art object detector with tracker, which is followed by tracklet action classification and post-processing units. We solve various aspects of the challenging action classification problem such as multi-label class-imbalance training using PLM method and learning multi-label action correlations using LSEP loss. We also demonstrate the importance of knowledge distillation in improving the computation efficiency of our system. We show state-of-the-art performance on ActEV-SDL UF-full dataset and second place in TRECVID 2021 ActEV challenge.

7. Acknowledgement

Authors would like to acknowledge support from the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. D17PC00345. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Authors would also like to thank Jonathan Fiscus (NIST) for providing useful tools and data for system evaluation and comparison.

References

[1] K. Hara, H. Kataoka, and Y. Satoh. Towards good practice for action recognition with spatiotemporal 3d convolutions. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2516–2521, 2018.

[2] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020.

[3] Ugur Demir, Yogesh S Rawat, and Mubarak Shah. Tinyvirat: low-resolution video action recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7387–7394. IEEE, 2021.

[4] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. *arXiv preprint arXiv:2101.07974*, 2021.

[5] Kevin Duarte, Yogesh Rawat, and Mubarak Shah. Videocapsulenet: A simplified network for action detection. In *Advances in Neural Information Processing Systems*, pages 7610–7619, 2018.

[6] Mamshad Nayeem Rizve, Ugur Demir, Praveen Tirupattur, Aayush Jung Rana, Kevin Duarte, Ishan R. Dave, Yogesh Singh Rawat, and Mubarak Shah. Gabriella: An online system for real-time activity detection in untrimmed surveillance videos. *CoRR*, abs/2004.11475, 2020.

[7] Aayush J. Rana and Yogesh S. Rawat. We don't need thousand proposals: Single shot actor-action detection in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2960–2969, January 2021.

[8] Ishan Dave, Zacchaeus Scheffer, Praveen Tirupattur, Yogesh Rawat, and Mubarak Shah. Ucf-system: Activity detection in untrimmed videos. 2020.

[9] Wenhe Liu, Guoliang Kang, Po-Yao Huang, Xiaojun Chang, Yijun Qian, Junwei Liang, Liangke Gui, Jing Wen, and Peng Chen. Argus: Efficient activity detection system for extended video analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, March 2020.

[10] Praveen Tirupattur, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. Modeling multi-label action dependencies for temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1460–1470, June 2021.

[11] Sirnam Swetha, Hilde Kuehne, Yogesh S Rawat, and Mubarak Shah. Unsupervised discriminative embedding for sub-action learning in complex activities. *arXiv preprint arXiv:2105.00067*, 2021.

[12] Naman Biyani, Aayush J Rana, Shruti Vyas, and Yogesh S Rawat. Larnet: Latent action representation for human action synthesis, 2021.

[13] Sarah Shiraz, Krishna Regmi, Shruti Vyas, Yogesh S. Rawat, and Mubarak Shah. Novel view video prediction using a dual representation, 2021.

[14] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 464–474, 2021.

[15] Shoufa Chen, Peize Sun, Enze Xie, Chongjian Ge, Jiannan Wu, Lan Ma, Jiajun Shen, and Ping Luo. Watch only once: An end-to-end video action detection framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8178–8187, 2021.

[16] Yuanzhong Liu, Zhigang Tu, Liyu Lin, Xing Xie, and Qianqing Qin. Real-time spatio-temporal action localization via learning motion representation. In *ACCV Workshops*, pages 184–198, 2020.

[17] Bo Chen and Klara Nahrstedt. Escalation: a framework for efficient and scalable spatio-temporal action localization. In *Proceedings of the 12th ACM Multimedia Systems Conference*, pages 146–158, 2021.

[18] Okan Köpüklü, Xiangyu Wei, and Gerhard Rigoll. You only watch once: A unified cnn architecture for real-time spatiotemporal action localization. *arXiv preprint arXiv:1911.06644*, 2019.

[19] Chunhui Gu, Chen Sun, Sudheendra Vijayanarasimhan, Caroline Pantofaru, David A. Ross, George Toderici, Yeqing Li, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. *CoRR*, abs/1705.08421, 2017.

[20] Ang Li, Meghana Thotakuri, David A Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *arXiv preprint arXiv:2005.00214*, 2020.

[21] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017.

[22] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *International Conf. on Computer Vision (ICCV)*, pages 3192–3199, December 2013.

[23] Ya Li, Guanyu Chen, Xiangqian Cheng, Chong Chen, Shaoqiang Xu, Xinyu Li, Xuanlu Xiang, Yanyun Zhao, Zhicheng Zhao, and Fei Su. Bupt-mcprl at trecvid 2019: Actev and ins. In *TRECVID*, 2019.

[24] Takashi Hosono, Kiyohito Sawada, Yongqing Sun, Kazuya Hayase, and Jun Shimamura. Activity normalization for activity detection in surveillance videos. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1386–1390. IEEE, 2020.

[25] Zhijian Hou, Yingwei Pan, Ting Yao, and Chong-Wah Ngo. Vireojd-mm@ trecvid 2019: Activities in extended video (actev). In *TRECVID*, 2019.

[26] Yongqing Sun, Xu Chen, Chaoyu Li, Kiyohito Sawada, Takashi Hosono, Jun Zhu, Chengjuan Xie, Sixiang Huang,

Lan Wang, Kai Hu, et al. Ntt_cqupt@ trecvid2019 actev: Activities in extended video. In *TRECVID*, 2019.

[27] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5725–5734, 2019.

[28] Wenhe Liu, Guoliang Kang, Po-Yao Huang, Xiaojun Chang, Yijun Qian, Junwei Liang, Liangke Gui, Jing Wen, and Peng Chen. Argus: Efficient activity detection system for extended video analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 126–133, 2020.

[29] Konstantinos Gkountakos, Despoina Touska, Konstantinos Ioannidis, Theodora Tsikrika, Stefanos Vrochidis, and Ioannis Kompatsiaris. Spatio-temporal activity detection and recognition in untrimmed surveillance videos. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pages 451–455, 2021.

[30] Shuo Chen, Pascal Mettes, Tao Hu, and Cees GM Snoek. Interactivity proposals for surveillance videos. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pages 108–116, 2020.

[31] Kellie Corona, Katie Osterdahl, Roderic Collins, and Anthony Hoogs. Meva: A large-scale multiview, multimodal video dataset for activity detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1060–1068, January 2021.

[32] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. In *arXiv preprint arXiv:2001.04608*, 2020.

[33] Lin Song, Shiwei Zhang, Gang Yu, and Hongbin Sun. Tacnet: Transition-aware context network for spatio-temporal action detection. *CoRR*, abs/1905.13417, 2019.

[34] Jiaojiao Zhao and Cees G. M. Snoek. Dance with flow: Two-in-one stream action detection. *CoRR*, abs/1904.00696, 2019.

[35] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 464–474, 2021.

[36] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. Step: Spatio-temporal progressive learning for video action detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[37] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (T-CNN) for action detection in videos. *CoRR*, abs/1703.10664, 2017.

[38] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[39] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[40] Gunnar A. Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Charades-ego: A large-scale dataset of paired third and first person videos. *CoRR*, abs/1804.09626, 2018.

[41] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *CoRR*, abs/1507.05738, 2015.

[42] 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, 2011.

[43] Hatem Belhassen, Heng Zhang, Virginie Fresse, and El-Bay Bourennane. Improving video object detection by seq-bbox matching. In *VISIGRAPP*, 2019.

[44] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Humphrey Shi, Jianan Li, Shuicheng Yan, and Thomas S. Huang. Seq-nms for video object detection. *ArXiv*, abs/1602.08465, 2016.

[45] Alberto Sabater, Luis Montesano, and Ana C. Murillo. Robust and efficient post-processing for video object detection. *CoRR*, abs/2009.11050, 2020.

[46] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, and Wanli Ouyang. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Trans. Cir. and Sys. for Video Technol.*, 28(10):2896–2907, October 2018.

[47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[48] Glenn Jocher. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements. `https://github.com/ultralytics/yolov5`, October 2020.

[49] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2015. cite arxiv:1506.02640.

[50] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. *CoRR*, abs/1602.00763, 2016.

[51] Kevin Duarte, Yogesh Singh Rawat, and Mubarak Shah. Plm: Partial label masking for imbalanced multi-label classification. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2733–2742, 2021.

[52] Yuncheng Li, Yale Song, and Jiebo Luo. Improving pairwise ranking for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3617–3625, 2017.

[53] Mathew Monfort, Bowen Pan, Kandan Ramakrishnan, Alex Andonian, Barry A McNamara, Alex Lascelles, Quanfu Fan, Dan Gutfreund, Rogerio Feris, and Aude Oliva. Multi-moments in time: Learning and interpreting models for multi-action video understanding. *arXiv preprint arXiv:1911.00232*, 2019.

[54] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[55] 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.

[56] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[57] Yooyoung Lee, Jonathan Fiscus, Andrew Delgado, Lukas Diduch, Eliot Godard, Baptiste Chocot, Jesse Zhang, Jim Golden, Afzal Godil, and Diane Ridgeway. Actev 2021 sequestered data leaderboard (sdl) evaluation plan.

[58] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[59] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12046–12055, 2019.

[60] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 6202–6211, 2019.

[61] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5552–5561, 2019.