University of Applied Sciences Mittweida at TRECVID 2023

Rico Thomanek[1], Max Schlosser[1], Dominik Breck[1], Benny Platte[1], Christian Roschke[1], Marc Ritter[1], and Maximilian Eibl[2]

[1]University of Applied Sciences Mittweida, D-09648 Mittweida, Germany
[2]Chemnitz University of Technology, D-09107 Chemnitz, Germany

Abstract. Analyzing CCTV video data to identify people and objects and to recognize their complex activities remains a challenging scientific task. Currently, various (semi-)automated systems are being used to address these challenges. The use of state-of-the-art Convolutional Neural Networks (CNNs) is continuously improving the accuracy rates for object detection and tracking. In our contribution to the detection of activities involving people and objects, we present a heterogeneous system that improves performance by fusing data from different detection systems. This is achieved by heuristically combining several state-of-the-art systems for object detection, location classification, and activity recognition.

We integrate advanced neural networks for object tracking and activity analysis in our contribution to *Activity of Extended Video* (ActEV), a task that focuses on detecting more complex activities of people or objects. These systems enable the extraction of bounding boxes for regions of interest or objects that can be used for further processing steps. Based on the extracted tracking results, skeleton-based and spatio-temporal activity determination methods are applied.

1 Introduction to our appearance at ActEV Self-Reported Leaderboard Challenge

In recent years, the worldwide use of surveillance cameras has increased significantly. Closed Circuit Television (CCTV) cameras continuously capture an exponentially growing amount of visual data. Typically, this data is analyzed after the fact for evidence of relevant activity. There is a growing need for intelligent and resource-efficient analysis of these videos, especially in the area of traffic safety, such as intersection surveillance or other highly sensitive environments. In the context of predictive policing, the focus is on detecting the movement of objects, including people. Continuous observation of movements over several hours is an extremely monotonous task that not only requires an exceptional level of concentration, but also leads to rapid fatigue. Another phenomenon is ”unaware blindness”: intense concentration on a particular object or activity can cause unanticipated objects to go unnoticed-even when these actions occur in the central field of view (Simons and Chabris, 1999). Intellectual evaluation of these data is limited primarily by human resources and is further limited by the high human error rate due to monotonous work.

This paper addresses our approach to the TRECVid task ’Activity in Extended Video’ (ActEV, (NIST-ActEV-Team, 2023)). Specifically, it deals with automatic detection of object activity in surveillance areas. ActEv aims to develop robust algorithms for automatic detection of activity in multi-camera streaming video environments. This task is a subset of TRECVid and is an extension of the Surveillance Event Detection Tasks (SED). It aims to detect events in real time. The analyzed dataset for this study is MEVA (Corona et al., 2021a), which consists of about 9300 hours of unprocessed continuous video from the video surveillance domain. Here, ActEV includes the challenges of ”activity detection” (AD) and ”activity and object detection” (AOD). AD is about detecting activity and determining the areas of the image where that activity is taking place. AOD extends this task by additionally requiring the identification of the actors involved and their localization in space.

In the area of activity recognition, a number of demanding challenges are encountered in connection with the available video material. These difficulties occur due to different perspectives of the cameras, varying distances to relevant objects, and highly variable recording qualities. In this research work, we present an innovative method for detecting and capturing activities in video data. Our main goal is to develop a highly complex and heterogeneous system that improves the

Correspondence to: Rico Thomanek
rico.thomanek@hs-mittweida.de

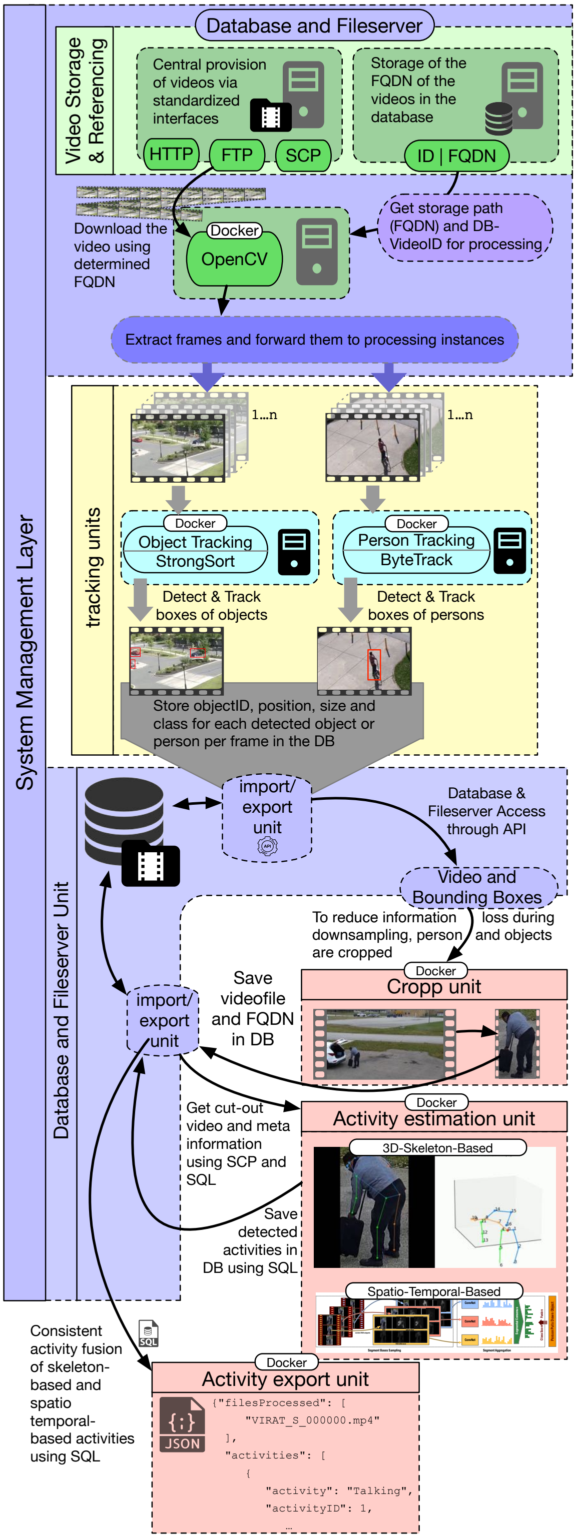
results of standalone detectors through information sharing in the form of data fusion. As a result, more accurate activity detection results will be achieved. In addition, system performance will be significantly enhanced through scalable parallel data processing and inter-process communication via standardized interfaces.

2 System Architecture

Our system integrates multiple client units that can perform recognition tasks in parallel either on a single hardware platform or distributed across multiple physical machines, depending on the available hardware in the form of Docker containers. For permanent storage and centralized organization of all raw data and results, we use a dedicated database server. MEVA raw video data is accessed via a distributed file system that supports various access protocols such as HTTP, FTP and SCP. This implementation eliminates the need for manual provisioning of the raw data on the processing clusters. Instead, the required source material is obtained directly during data analysis using an appropriate application protocol.

To minimize protocol overhead during data transport, the full videos are transmitted to the processing units and broken down into individual frames on site. The processing algorithms are then applied based on these single frames, and the detected results are immediately persisted in the central database. To access these detection results, we have set up a self-developed web service that provides standardized APIs in various exchange formats.

Our session management layer allows parallel processing to be easily launched in the form of Docker containers to efficiently process the total of 201 videos. This layer handles the task of distributing and scaling the data processing tasks and allocates resources to the processing nodes and services. This ensures efficient parallelization of the processes. Automated data processing continues until all resources are successfully completed. In case of occurring errors, integrated error correction mechanisms initiate a reprocessing of the data. After three unsuccessful attempts, processing is aborted, the processing block with errors is marked as such in the database, and the next processing block is made available. All intermediate and final results generated during the entire working process are carefully archived in the database. Via the corresponding API, each node instance is enabled to directly access these resources and use them for further analysis and evaluation.


System Management Layer
Video Storage & Referencing
Database and Fileserver
Central provision of videos via standardized interfaces
Storage of the FQDN of the videos in the database
HTTP
FTP
SCP
ID | FQDN
Download the video using determined FQDN
Docker
OpenCV
Get storage path (FQDN) and DB-VideoID for processing
Extract frames and forward them to processing instances
tracking units
1...n
1...n
Docker
Object Tracking StrongSort
Docker
Person Tracking ByteTrack
Detect & Track boxes of objects
Detect & Track boxes of persons
Store objectID, position, size and class for each detected object or person per frame in the DB
Database and Fileserver Unit
import/ export unit
Database & Fileserver Access through API
Video and Bounding Boxes
To reduce information loss during downsampling, person and objects are cropped
Save videofile and FQDN in DB
import/ export unit
Docker
Cropp unit
Get cut-out video and meta information using SCP and SQL
Docker
Activity estimation unit
3D-Skeleton-Based
Save detected activities in DB using SQL
Spatio-Temporal-Based
Segment Bases Sampling
Segment Aggregation
ConvNet
ConvNet
ConvNet
Segmental Consensus
Class Score Fusion
Person Puts Down Object
Consistent activity fusion of skeleton-based and spatio temporal-based activities using SQL
SQL
Docker
Activity export unit
JSON

```
{"filesProcessed": [
     "VIRAT_S_000000.mp4"
  ],
  "activities": [
     {
        "activity": "Talking",
        "activityID": 1,
          …
```

Figure 1: Our holistic system workflow for ActEV.

To perform the tasks in the extended video task, open source frameworks for object identification and tracking as well as two self-trained neural networks for space-time-based and pose-based activity recognition were used. As can be seen in Figure 1, in a first processing step, the people and objects appearing in all videos to be analyzed are tracked and their frame and bounding box information is stored in the database. In order to minimize the loss of information when reducing the image resolution (downsampling) for pose extraction, individual videos of all detected persons were extracted and then subjected to pose extraction. The identified poses were also stored in the database. Based on the extracted individual videos, a space-time and pose-based activity recognition is performed. The activity recognition, which is based on the recognized objects (e.g. vehicle), is performed using simple heuristics. In order to identify specific activities, the extracted partial results are then merged using individual SQL queries.

In order to fulfill the requirements of the ActEV task, all results received are transferred from the management system to the processing or scoring unit. This unit contains the necessary business logic for evaluation and generates a result object. This object is then transferred to an XML or JSON container by the export unit. The visualization unit uses these containers directly to display the tracking and activity recognition results. This makes it possible to create an interface for the intelligent annotation of the data as part of the ActEV task. The quality of the results can thus be intellectually assessed after the competition period to support follow-up work or more in-depth analysis.

2.1 Frameworks

We recognise people and objects separately and integrate all results into feature vectors. The identification of activities is done by using SQL queries that relationally link all results via the frame ID, which is realised by foreign keys. We use the following algorithms/frameworks for the detailed extraction of features: ByteTrack, Strongsort, mmpose and mmaction2.

We use two different algorithms to effectively recognise and track people and objects. To ensure decentralised processing of the videos, we have integrated these algorithms into our existing infrastructure.

ByteTrack

We use the ByteTrack framework (Zhang et al., 2022) to track people. The ByteTrack framework is implemented in Python and uses PyTorch for the development of spatiotemporal tracking algorithms. We use ByteTrack as a Docker container and perform parallel processing of the test dataset with multiple Docker containers, which leads to a significant reduction in processing time.

StrongSort

We use the yolo_tracking framework (Broström, 2023) to track objects. It enables the use of various tracking algorithms. For tracking objects, we use the StrongSort algorithm (Du et al., 2023).

MMPose (MMPose-Contributors, 2020) is an open source pose estimation framework based on PyTorch, which is part of the OpenMMLab (OpenMMLab-Team) project. MMPose supports numerous state of the art pose estimation algorithms. In order to reduce the loss of information for small individuals during downsampling, all detected and tracked humans are extracted from the source material and individually passed to the pose detector. We use MotionBERT (Zhu et al., 2023), a framework for generating 3D poses from corrupted 2D skeleton sequences, for our pose estimation. First, 2D skeletons are extracted from the individual videos of the persons. These potentially corrupted 2D skeletons are then converted into 3D skeletons using a motion encoder. The data provided by MMPose are normalized values with the hip as the zero point. Based on this data, activity recognition is then performed using the MMAction framework.

MMAction2 (MMAction2-Contributors, 2020) is a video understanding framework that integrates state-of-the-art algorithms and datasets and improves the recognition of skeletal actions with different motion modalities. It introduces the Inferencer tool, which enables model inference in just one line of code and greatly simplifies the process.

2.2 Data handling and interface

For the permanent storage and provision of data we use the architecture described under (Thomanek et al., 2018).

As described, we use a chain of docker instances to process all data analysis tasks. Docker is open source software for isolating applications with container virtualization. Docker simplifies the deployment of applications by simplifying the transport and installation of containers with all necessary packages as files.

For video activity detection, it is necessary to analyze successive image data. This results in large amounts of image data that must be efficiently stored and distributed to the processing frameworks. To access this data, we have implemented a session management layer, which distributes the data to be processed to the frameworks working in parallel. For this purpose, each framework must log on to the session management and submit a processing request. Session management manages and monitors the processing process in the background and assigns unprocessed data to the free framework instances. The individual sessions are monitored using a decentralized heartbeat function, in which the framework instances must signal their availability at regular intervals. If an instance is no longer available due to errors, unprocessed data is then assigned to another free instance. The execution of multiple processing instances on physical hardware is determined by evaluating their CPU, GPU, and memory utilization. Depending on the resources required, the physical hardware can thus be optimally utilized and the processing time significantly reduced.

3 Workflow of Our Method

Our approach consists of recognizing the identified objects (vehicles and people) in the video material and using their position and bounding boxes for activity recognition. The algorithms *ByteTrack* and *Strongsort* were used. To derive activities from the objects detected by *ByteTrack* and *StrongSort*, their unique object ID and bounding box information were stored in the database. The center of the bounding boxes and the pattern information of all objects were used, and their change in position with respect to the previous and subsequent frame was determined.

The algorithm described in (Thomanek et al., 2019) for detecting the activities “vehicle_turns_left” and “vehicle_turns_right” was adapted based on the definitions for activities and the associated objects described in (ActEV Team, 2020).

To determine the activities “person_enters_vehicle”, “person_exits_vehicle”, “person_exits_scene_through_structure”, “person_enters_scene_through_structure”, “vehicle_starts” and “vehicle_stops”, the algorithm described in (Thomanek et al., 2019) was adapted based on the definitions for activities and the associated objects described in (ActEV Team, 2020).

For the remaining activities “person_closes_vehicle_door”, “person_reads_document”,“person_sits_down”, “person_stands_up”, “person_talks_to_person”, “person_texts_on_phone”,“person_interacts_with_laptop”, “person_transfers_object”, “person_opens_facility_door”, “person_opens_vehicle_door”,“person_picks_up_object” and “person_puts_down_object” we trained two custom activity classifiers that predict the most likely activity based on the time-varying pose keypoints and the spatiotemporal change in RGB space. All recognized activities were stored in the database and fused using SQL queries. For example, the activity “person_closes_vehicle_door” was only detected if there was a vehicle in the position range of the person.

3.1 Retrieval of Object Data for Tracking

We use the ByteTrack (Zhang et al., 2021) algorithm to recognise and track people. In the MOT17 and MOT20 benchmarks, ByteTrack has demonstrated high metrics such as MOTA, ID switches, Mostly Tracked Objects and Mostly Lost Objects. Especially for smaller displayed persons, the confidence value can often fall below a threshold value. ByteTrack uses a motion model that manages a queue called tracklets to store recognised objects and use them for tracking and matching between bounding boxes. Unassigned boxes are also compared with the low confidence bounding boxes in a second matching process, allowing even distant and small people to be tracked. ByteTrack uses the YOLOX detector for object recognition.

Since the networks we use in the ByteTrack algorithm are only trained on people, we have integrated the StrongSort (Du et al., 2023) algorithm with a YOLOv8 detector to recognise and track objects. StrongSort also shows outstanding results in the MOT17 and MOT20 benchmarks for the metrics mentioned. To predict object motion, StrongSort uses the NSA Kalman algorithm, which is based on a nonlinear state space model and enables adaptive calculation of the noise covariance. As a result, it provides more accurate estimates of object positions and velocities, leading to improved overall object tracking performance. The results of our tracking methods are stored in the central database for further processing steps.

3.2 Activity detection using STGCN based activity classifier

For the activity analysis, an approach was developed that can combine two methods for activity classification. This approach only targets person-related activities in the data set. Activities associated with cars, for example, are not considered. The basis for both components of the approach was MMAction2 (MMAction2-Contributors, 2020), which is available as open source software under the OpenMMLab (OpenMMLab-Team) computer vision libraries for numerous video understanding tasks and, in particular, activity classification. To create both approaches, a separate model was trained in each case using the training data set from the MEVA AWS Video Data Bucket from the “drops-123-r13” (Corona et al., 2021b) directory. Complete data series of individual activities were used in each case.

The first approach works purely pose-based by passing pose information of several contiguous frames to the model in a static keypoint format. In order to minimize the loss of information when reducing the image resolution (downsampling) during pose extraction, all persons detected and tracked by the tracker were extracted with regard to their detected bounding box size and stored as individual videos in the central file server and referenced in the database. This reduces the loss of information, especially in the case of people who are far away and therefore small. As a result, pose information can also be obtained for these people. The extraction of the bone points takes place in three-dimensional space, whereby the bone points are delivered normalized in relation to the hip as a zero point. Based on these extracted bone points, the model estimates what activity the human character performs in the frames. As window size we use a batch of 30 skeletal values, which corresponds to a length of 1s for the given test data set. The model is based on a Spatial Temporal Graph Convolutional Network (STGCN). To train the pose-based model, the individuals' bone points were first extracted from the video clips using MMPose (MMPose-Contributors, 2020) in COCO format. The individual data series, which contain the bone points of all frames of an activity, were then used with their respective labels for training.

3.3 Activity detection using TSN based activity classifier

The second approach, on the other hand, is based on RGB data using the Temporal Segment Network (TSN) shown in Figure 2. The cropped person videos were also used for this in order to minimize the loss of information through downsampling and to eliminate unnecessary information in the rest of the image. As shown in Figure 2, each person video is

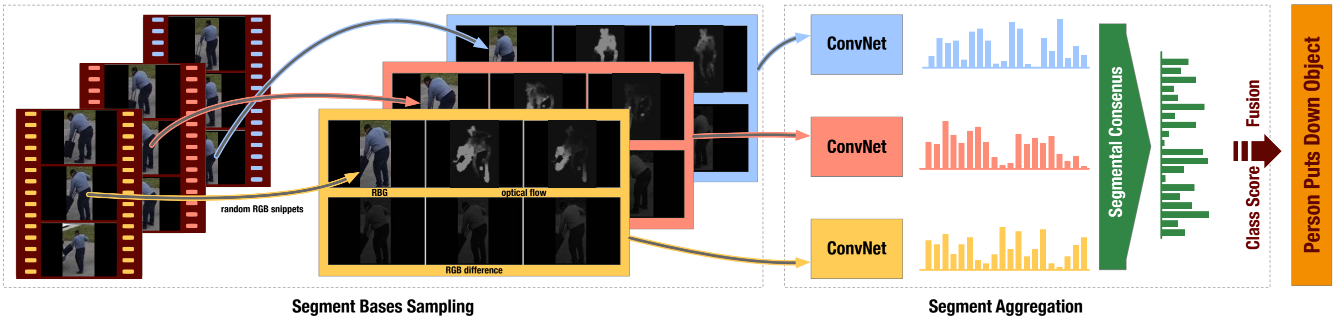


Figure 2: Temporal segement network. (according to (Wang et al., 2019))

divided into K segments. From each of these segments, a random RGB frame, the RGB difference and the optical flow are transferred to a ConvNet network. The resulting class values are fused by a segmental consensus function to obtain an activity prediction. This approach also divides a batch of 30 contiguous frames into K segments. The results of both methods allow the creation of single or combined views. In the combined views, only results in which both methods have identified the same activity are included.

3.4 Activity detection using simple heuristics

Some activities are closely related to other objects. The interaction of people with certain objects such as cars and bags can be considered an activity over a period of time. To make this possible, precise detection and tracking of the objects in each individual frame is required. The bounding boxes of all objects must then be examined for possible overlaps. For this purpose, an overlap value is calculated directly in the database using a geometric function in Postgres SQL. To ensure that it is not just a matter of brief touches, all common frames between the interacting objects are determined and interpolated if necessary. This results in a common start and end frame as well as an overlap value for each interaction.

The individual objects are identified by evaluating the tracking results. In addition, the direction vectors of the tracked objects are also determined and included in the evaluation. This makes it possible to check whether the objects are moving away from each other, towards each other or in the same direction over time. However, if the tracking fails or provides incorrect results, this can have a negative impact on the recognition of activities. For example, the same object may disappear briefly within a video and reappear elsewhere with a different ID. This leads to incomplete activity recognition results. To overcome this problem in the best possible way, we have tried to use clustering techniques.

4 Results and Future Work in Activity Event Detection

In activity event recognition, different approaches were used to evaluate the performance. The first approach was based on an activity classifier based on Space-Time Graph Convolutional Networks (STGCN). The results obtained show that the average value for the probability of omission (PMiss) at 0.1 relative false alarms (rfa) was 0.9841. The average Normalized Mode (nMODE) and Normalized Area Under the Detection Curve (nAUDC) values at 0.1rfa were 0.1349 and 0.9856, respectively. In terms of Activity Detection (AD), the corresponding values were 0.9641 and 0.9669. The second approach used an activity classifier based on Temporal Segment Networks (TSN). Here, an average PMiss value of 0.9843 at 0.1rfa was achieved. The average values for nMODE and nAUDC at 0.1rfa were 0.1382 and 0.9863. The values for activity detection (AD) were 0.9632 for PMiss and 0.9673 for nAUDC. The third approach used simple heuristics for activity detection and obtained the following results: Average PMiss at 0.1rfa of 0.9887, average nMODE of 0.1939 and average nAUDC of 0.9857. In the context of activity detection, the values were 0.9778 for PMiss and 0.9706 for nAUDC. It was concluded that the best results were achieved by combining the pose and RGB approaches. Second place was taken by the RGB approach, while third place was based solely on the pose approach. For future work, we plan to increase the training data set. Furthermore, fusion at decision level with the inclusion of plausibility checks is proposed as a promising approach for the further development of this activity recognition system.

Acknowledgments

The European Union and the European Social Fund for Germany partially funded this research. This work was also partially funded by the German Federal Ministry of Education and Research in the program of Entrepreneurial Regions InnoProfileTransfer in the project group localizeIT (funding code 03IPT608X). Program material in sections 1–3 is copyrighted by BBC. We want to thank all the organizers of these tasks, especially George Awad and Afzal Godil, for the hard work they put into the annotation, evaluation and organization of these challenges.

References

ActEV Team, N.: Performers: DIVA Annotation Definitions for MEVA Data, 2020.

Broström, M.: BoxMOT: pluggable SOTA tracking modules for object detection, segmentation and pose estimation models, doi:https://zenodo.org/record/7629840, https://github.com/mikel-brostrom/yolo_tracking, 2023.

Corona, K., Osterdahl, K., Collins, R., and Hoogs, A.: MEVA: A Large-Scale Multiview, Multimodal Video Dataset for Activity Detection, in: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1059–1067, doi:10.1109/WACV48630.2021.00110, 2021a.

Corona, K., Osterdahl, K., Collins, R., and Hoogs, A.: 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), pp. 1060–1068, 2021b.

Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., and Meng, H.: StrongSORT: Make DeepSORT Great Again, IEEE Transactions on Multimedia, pp. 1–14, doi:10.1109/TMM.2023.3240881, 2023.

MMAction2-Contributors: OpenMMLab's Next Generation Video Understanding Toolbox and Benchmark, https://github.com/open-mmlab/mmaction2, 2020.

MMPose-Contributors: OpenMMLab Pose Estimation Toolbox and Benchmark, https://github.com/open-mmlab/mmpose, 2020.

NIST-ActEV-Team: ActEV Self-Reported Leaderboard (SRL) Chalenge - Draft Evaluation Plan, https://actev.nist.gov/uassets/Draft_ActEV_SRL_Eval_Plan_May10.pdf, 2023.

OpenMMLab-Team: OpenMMLab, https://openmmlab.com/, [Accessed 07-11-2023].

Simons, D. J. and Chabris, C. F.: Gorillas in Our Midst: Sustained Inattentional Blindness for Dynamic Events, Perception, 28, 1059–1074, doi:10.1068/p281059, 1999.

Thomanek, R., Roschke, C., Manthey, R., Platte, B., Rolletschke, T., Heinzig, M., Vodel, M., Kowerko, D., Kahl, S., Zimmer, F., Eibl, M., and Ritter, M.: University of Applied Sciences Mittweida and Chemnitz University of Technology at TRECVID 2018, Gaithersburg, Maryland, USA, 2018.

Thomanek, R., Roschke, C., Platte, B., Manthey, R., Rolletschke, T., Heinzig, M., Vodel, M., Zimmer, F., and Eibl, M.: A scalable system architecture for activity detection with simple heuristics, in: Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision Workshops, WACVW 2019, doi:10.1109/WACVW.2019.00012, 2019.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., and Van Gool, L.: Temporal Segment Networks for Action Recognition in Videos, IEEE Transactions on Pattern Analysis and Machine Intelligence, 41, 2740–2755, doi:10.1109/TPAMI.2018.2868668, 2019.

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., and Wang, X.: ByteTrack: Multi-Object Tracking by Associating Every Detection Box, CoRR, abs/2110.0, https://arxiv.org/abs/2110.06864, 2021.

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., and Wang, X.: ByteTrack: Multi-Object Tracking by Associating Every Detection Box, 2022.

Zhu, W., Ma, X., Liu, Z., Liu, L., Wu, W., and Wang, Y.: MotionBERT: A Unified Perspective on Learning Human Motion Representations, 2023.