ROAD: The ROad event Awareness Dataset for Autonomous Driving

Gurkirt Singh[3] Stephen Akrigg[1] Manuele Di Maio[5] Valentina Fontana[2]
Reza Javanmard Alitappeh[4] Suman Saha[3] Kossar Jeddisaravi[4] Farzad Yousefi[4]
Jacob Culley[1] Tom Nicholson[1] Jordan Omokeowa[1] Salman Khan[1]
Stanislao Grazioso[2] Andrew Bradley[1] Giuseppe Di Gironimo[2] Fabio Cuzzolin[1]

Abstract

Humans approach driving in a holistic fashion which entails, in particular, understanding road events and their evolution. Injecting these capabilities in an autonomous vehicle has thus the potential to take situational awareness and decision making closer to human-level performance. To this purpose, we[1] introduce the ROad event Awareness Dataset (ROAD) for Autonomous Driving, to our knowledge the first of its kind. ROAD is designed to test an autonomous vehicle's ability to detect road events, defined as triplets composed by a moving agent, the action(s) it performs and the corresponding scene locations. ROAD comprises 22 videos, originally from the Oxford RobotCar Dataset, annotated with bounding boxes showing the location in the image plane of each road event. We also provide as baseline a new incremental algorithm for online road event awareness, based on inflating RetinaNet along time, which achieves a mean average precision of 16.8% and 6.1% for frame-level and video-level event detection, respectively, at 50% overlap. Though promising, these figures highlight the challenges faced by situation awareness in autonomous driving. Finally, ROAD allows scholars to investigate exciting tasks such as complex (road) activity detection, future road event anticipation and the modelling of sentient road agents in terms of mental states. Dataset can be obtained from `https://github.com/gurkirt/road-dataset` *and baseline code from* `https://github.com/gurkirt/3D-RetinaNet`.

1. Introduction

In recent years, *autonomous driving* (or *robot-assisted driving*) has emerged as a fast-growing research area. The race towards fully autonomous vehicles pushed many large companies, such as Google, Toyota and Ford, to develop their own concept of *robot-car* [100, 50, 65]. While self-driving cars are widely considered to be a major development and testing ground for the real-world application of artificial intelligence, major reasons for concern remain in terms of safety, ethics, cost, and reliability [59]. From a safety standpoint, in particular, smart cars need to robustly interpret the behaviour of the humans (drivers, pedestrians or cyclists) they share the environment with, in order to cope with their decisions. *Situation awareness* and the ability to understand the behaviour of other road users are thus crucial for the safe deployment of autonomous vehicles (AVs).

The latest generation of robot–cars is equipped with a range of different sensors (i.e., laser rangefinders, radar, cameras, GPS) to provide data on what is happening on the road [6]. The information so extracted is then fused to suggest how the vehicle should move [2, 28, 95, 14]. Some authors, however, maintain that vision is a sufficient sense for AVs to navigate their environment, supported by humans' ability to do just so. Without enlisting ourselves as supporters of the latter point of view, in this paper we consider the context of *vision-based* autonomous driving [4] from video sequences captured by cameras mounted on the vehicle in a streaming, online fashion.

While detector networks [76] are routinely trained to facilitate object and actor recognition in road scenes, this simply allows the vehicle to 'see' what is around it, without any real understanding of the scene context. Our position is that robust self-driving capabilities require a deeper, more human-like understanding of the road environment and of the evolving behaviour of other road users over time. Behavioural inference has been proposed as an option [25], as the historical behaviour of road users can be used to predict possible future events in accordance with a discrete set of policies, defined at programming time. Ontology-based approaches [27] have been considered to allow decisions about how to act in certain situations according to a set of

[1]VAIL, Oxford Brookes University, UK, [2] University of Naples Federico II, Italy, [3] CVL, ETH Zurich, [4] University of Science and Technology of Mazandaran, Behshahr, Iran, [5]Siemens SpA, Bologna, Italy. This collaborative work took place at Oxford Brookes University when V. Fontana and M. Di Maio were visiting students, and S. Akrigg and G. Singh were students there. E-mail: gurkirt.singh@vision.ee.ethz.ch, fabio.cuzzolin@brookes.ac.uk

rules (e.g. 'always give way at a junction'). Such an approach, however, is way too rigid to allow an AV to flexibly adapt to what happens in the environment, and does not at all contemplate the learning element.

On the opposite side of the spectrum lies end-to-end reinforcement learning. There, the behaviour of a human driver in response to road situations is used to train, in an imitation learning setting [17], an autonomous car to respond in a more 'human-like' manner to road scenarios. This, however, requires an astonishing amount of data from a myriad of road situations. For highway driving only, a relatively simple task when compared to city driving, Fridman et al. in [26] had to use a whole fleet of vehicles to collect 45 million frames. Perhaps more importantly, in this approach the network learns a mapping from the scene to control inputs without any real understanding of why the driver may be behaving in a certain manner. As discussed in [20], many authors [75, 78] have recently highlighted the insufficiency of models which directly map observations to actions, specifically in the self-driving cars scenario. On the specific aspect of learning human goals (or preferences, or reward functions) just from examples, in a brute-force approach, Armstrong and Mindermann [1] showed that, if no assumptions are made about human rationality, then a human's preferences cannot be recovered from behaviour only, no matter how much data the system can learn from.

1.1. ROAD: a multi-label, multi-task dataset

An alternative route, and one more inspired by human behaviour, seeks to exploit sophisticated situation awareness to extract from road scenes semantically meaningful concepts, as a stepping stone for intention prediction and automated decision making. One advantage of this approach is that it allows the autonomous vehicle to focus on a much smaller amount of relevant information when learning how to make its decisions, in a way arguably closer to how decision making takes place in humans.

To allow the research community to thoroughly investigate situation awareness for autonomous driving, this paper introduces ROAD, the first *ROad event Awareness in Autonomous Driving Dataset*, built upon (a fraction of) the Oxford RobotCar Dataset [57]. ROAD is the result of annotating 22 carefully selected, relatively long-duration (ca 8 minutes each) videos from the RobotCar dataset in terms of what we call *road events* (REs), as seen from the point of the view of the autonomous vehicle capturing the video. REs are defined as triplets $E = (Ag, Ac, Loc)$ composed by a moving agent Ag, the action Ac it performs, and the location Loc in which this takes place. Agent, action and location are all classes in a finite list compiled by surveying the content of the 22 videos. Road events are represented as 'tubes', i.e., time series of frame-wise bounding box detections.

ROAD is a dataset of significant size, as $122K$ video frames are labelled for a total of $560K$ detection bounding boxes in turn associated with $1.7M$ unique individual labels, broken down into $560K$ agent labels, $640K$ action labels and $499K$ location labels. The dataset was designed according to the following principles.

- A *multi-label* benchmark: each road event is composed by the label of the (moving) agent responsible, the label(s) of the type of action(s) being performed, and labels describing where the action is located.
- Each event can be assigned *multiple instances* of the same label type whenever relevant (e.g., an RE can be an instance of both *moving away* and *turning left*).
- The labelling is done *from the point of view of the AV*: the final goal is for the autonomous vehicle to use this information to make the appropriate decisions.
- The meta-data is intended to contain all the information required to fully describe a road scenario: an illustration of this concept is given in Figure 1. After closing one's eyes, the set of labels associated with the current video frame should be sufficient to recreate the road situation in one's head (or, equivalently, sufficient for the AV to be able to make a decision).

To allow scientists to assess the performance of their method of choice in terms of road event detection, a robust baseline is proposed that combines state-of-the-art single-stage object detector technology with an online tube construction method [88] with the aim of linking detections over time to create *event tubes* [81, 35].

The second objective of this work is to contribute to taking action detection into the real world (in particular, that of autonomous driving). ROAD moves away from human body actions almost entirely, to consider (besides pedestrian behaviour) actions performed by humans as drivers of various types of vehicles, shifting the paradigm from *actions performed by human bodies* to *events caused by agents*. As shown in our experiments, ROAD is much more challenging than current action detection benchmarks due to the complexity of road events happening in real, non-choreographed driving conditions, the number of active agents present and the variety of weather conditions encompassed. Importantly, it is naturally geared towards *online, incremental detection* because of autonomous driving requirements.

As described in Sec. 3.4, ROAD allows one to validate manifold tasks associated with situation awareness for self-driving, each associated with a label type (agent, action, location) or combination thereof: *spatiotemporal* (i) *agent detection*, (ii) *action detection*, (iii) *location detection*, (iv)

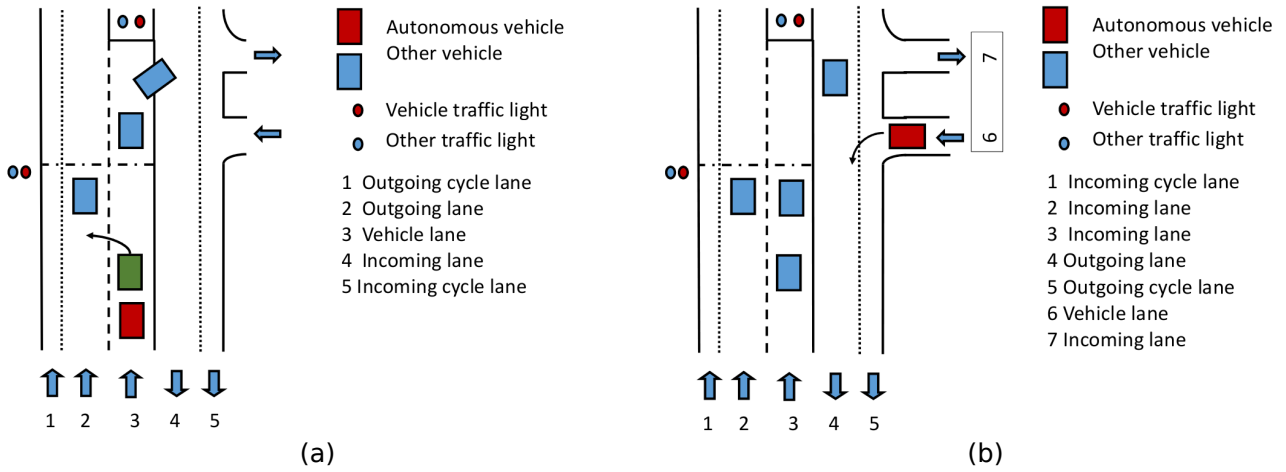

Figure 1. Use of labels in ROAD to describe typical road scenarios. (a) A green car is in front of the AV while changing lanes, as depicted by the arrow symbol. The associated event will then carry the following labels: *in vehicle lane* (location), *moving left* (action). Once the event is completed, the location label will change to: *in outgoing lane*. (b) Autonomous vehicle turning left from lane 6 into lane 4: lane 4 will be the *outgoing lane* as the traffic is moving in the same direction as the AV. However, if the AV turns right from lane 6 into lane 4 (a wrong turn), then lane 4 will become the *incoming lane* as the vehicle will be moving into the incoming traffic. The overall philosophy of ROAD is to use suitable combinations of multiple label types to fully describe a road situation, and allow a machine learning algorithm to learn from this information.

agent-action detection, (v) *road event detection*, as well as the (vi) *temporal segmentation of AV actions*. For each task one can assess both *frame-level* detection, which outputs independently for each video frame the bounding box(es) (BBs) of the instances there present and the relevant class labels, and *video-level* detection, which consists in regressing the whole series of temporally-linked bounding boxes (i.e., in current terminology, a 'tube') associated with an intance, together with the relevant class label. In this paper we conduct tests on both. All Tasks come with both the necessary annotation and a shared baseline, which is described in Section 4.

1.2. Contributions

This paper puts forward the following major contributions.

- A formal definition of the notion of road event, as a triplet composed by a road agent, the action(s) it performs and the location(s) of the event, all from the point of view of the AV.
- A push towards the adoption of event detection techniques in perception for autonomous driving, in a shift away from the simple detection of objects in the scene towards an understanding of the semantics of what takes place around the autonomous vehicle.
- A new ROad event Awareness Dataset for Autonomous Driving (ROAD), the first of its kind, designed to allow the testing of a whole range of tasks related to situation awareness for autonomous driving: agent and/or action detection, event detection, ego-action classification. Within the computer vision community, ROAD significantly pushes the boundaries of current action detection benchmarks, in terms of size, context, and specific challenges associated with the road scenario.
- A new method for online action/agent/event detection in autonomous driving, evolved from a previous work of ours proposing the first online action detector algorithm [88]. We adopt the former as a baseline for the ROAD tasks and report and critically assess its quantitative results for others to build upon.

Further, near-future extensions to additional tasks such as future event anticipation, decision making and machine theory of mind [20] are discussed in Section 6. We are confident that this work will lay the foundations upon which much further research in this area can be built.

1.3. Outline

The remainder of the paper is organised as follows. Section 2 reviews related work concerning existing datasets, both for autonomous driving (Sec. 2.1) and action detection (Sec. 2.2), as well as action detection methods (Sec. 2.3). Section 3 presents our ROAD dataset in full detail, including: its multi-label nature (Sec. 3.1), data collection (Sec. 3.2), annotation (Sec. 3.3), the tasks it is designed to validate (Sec. 3.4), and a quantitative summary (Sec. 3.5). Section 4 presents an overview on the novel online action

detection approach chosen as a baseline. Experiments are described in Section 5, where results are reported and critically analysed in detail. Section 6 outlines additional exciting tasks the dataset can be used as a benchmark for in the near future. Conclusions and future work are outlined in Section 7.

2. RELATED WORK

2.1. Autonomous driving datasets

In recent years a multitude of AV datasets have been released, mostly focusing on object detection and scene segmentation. We can categorise them into two main bins: (1) RGB without range data (single modality) and (2) RGB with range data (multimodal).

Single-modality datasets. Collecting and annotating RGB data only is relatively less time-consuming and expensive than building multimodal datasets including range data from LiDAR or radar. Most single-modality datasets [7, 18, 62, 106, 96, 13] provide 2D bounding box and scene segmentation labels for RGB images. Examples include Cityscapes [18], Mapillary Vistas [62], BDD100k [106] and Apolloscape [96]. To allow the studying of how vision algorithms generalise to different unseen data, [62, 13, 106] collect RGB images under different illumination and weather conditions. Other datasets only provide pedestrian detection annotation [21, 24, 101, 23, 107, 63, 19]. Recently, MIT and Toyota have released DriveSeg, which comes with pixel-level semantic labelling for 12 agent classes [22].

Multimodal datasets. KITTI [32] was the first-ever multimodal dataset. It provides depth labels from front-facing stereo images and dense point clouds from LiDAR alongside GPS/IMU (inertial) data. It also provides bounding-box annotations to facilitate improvements in 3D object detection. H3D [67] and KAIST [16] are two more examples of multimodal datasets. H3D provides 3D box annotations, using real-world LiDAR-generated 3D coordinates, in crowded scenes. Unlike KITTI, H3D comes with object detection annotations in a full 360-degree view. KAIST provides thermal camera data alongside RGB, stereo, GPS/IMU and LiDAR-based range data. Among other notable multimodal datasets [5, 56] only consist of raw data without semantic labels, whereas [43] and [15] provide labels for location category and driving behaviour, respectively. The most recent multimodal large-scale AV datasets [12, 46, 93, 70, 33] are significantly larger in terms of both data (also captured under varying weather conditions, e.g. by night or in the rain) and annotations (RGB, LiDAR/radar, 3D boxes). For instance, Argovers [12] doubles the number of sensors in comparison to KITTI [32] and nuScenes [9], providing 3D bounding boxes with tracking information for 15 objects of interest. Similarly, Lyft [46] provides 3D bounding boxes for cars and location annotation including lane segments, pedestrian crosswalks, stop signs, parking zones, speed bumps, and speed humps. Both 3D bounding boxes based on LiDAR data and 2D annotation on camera data for 4 objects classes are provided in Waymo [93]. In [70], using similar 3D annotation for 5 objects classes, the authors provide a more challenging dataset by adding more night-time scenarios using a faster-moving car. Amongst large-scale multimodal datasets, nuScenes [9], Lyft L5 [46], Waymo Open [93] and A*3D [70] are the most dominant ones in terms of number of instances, the use of high-quality sensors with different types of data (e.g., point clouds or 360° RGB videos), and richness of the annotation providing both semantic information and 3D bounding boxes. Furthermore, nuScenes [9], Argoverse [12] and Lyft L5 [46] provide contextual knowledge through human-annotated rich semantic maps, an important prior for scene understanding.

Trajectory prediction. Another line of work considers the problem of pedestrian trajectory prediction in the autonomous driving setting, and rests on several influential RGB-based datasets. To compile these datasets, RGB data were captured using either stationary surveillance cameras [51, 68, 64] or drone-mounted ones [77] for aerial view. [105, 11] use RGB images capturing an egocentric view from a moving car for future trajectory forecasting. Recently, the multimodal 3D point cloud-based datasets [32, 67, 46, 9, 93, 12], initially introduced for the benchmarking of 3D object detection and tracking, have been taken up for trajectory prediction as well. A host of interesting recent papers [74, 60, 73, 58] do propose datasets to study the intentions and actions of agents using cameras mounted on vehicles. However, they encompass a limited set of action labels (e.g. walking, standing, looking or crossing), wholly insufficient for a thorough study of road agent behaviour. Among them, TITAN [58] is arguably the most promising.

Our ROAD dataset is similar to TITAN in the sense that both consider actions performed by humans present in the road scene and provide spatiotemporal localisation for each person using multiple action labels. However, TITAN is a collection of much shorter videos which only last 10-20 seconds; it does not not contemplate agent location (a crucial source of information); the size of its vocabulary in terms of number of agents and actions is much smaller; finally, action labels are restricted to humans (pedestrians), rather then human-operated vehicles.

Our proposed ROAD dataset departs substantially from all previous efforts, as: (1) it builds upon the multimodal Oxford RobotCar dataset to formally define the notion of *road event* as a combination of three semantically-meaningful labels such as agent, action and location; (2) it provides both bounding-box-level *and* tube-level annotation (to validate methods that exploit the dynamics of mo-

tion patterns) on long-duration videos (thus laying the foundations for future work on event anticipation and continual learning); (3) it provides temporally dense annotation; (4) it labels the actions not only of physical humans but also of other relevant road agents such as vehicles of different kinds.

2.2. Action detection datasets

Providing annotation for action detection datasets is a painstaking process. Specifically, the requirement to track actors through the temporal domain makes the manual labelling of a dataset an extremely time consuming exercise, requiring frame-by-frame annotation. As a result, action detection benchmarks are fewer and smaller than, say, image classification, action recognition or object detection datasets.

Action recognition research can aim for robustness thanks to the availability of truly large scale datasets such as Kinetics [45], Moments[61] and others, which are the de-facto benchmarks in this area. The recent 'something-something' video database focuses on more complex actions performed by humans using everyday objects [36], exploring a fine-grained list of 174 actions. More recently, temporal activity detection datasets like ActivityNet [8] and Charades [83] have come to the fore. Whereas the latter still do not address the spatiotemporal nature of the action detection problem, however, datasets such as J-HMDB-21 [41], UCF24 [92], LIRIS-HARL [102], DALY [99] or the more recent AVA [38] have been designed to provide spatial and temporal annotations for human action detection.
In fact, most action detection papers are validated on the rather dated and small LIRIS-HARL[102], J-HMDB-21 [41], and UCF24 [92], whose level of challenge in terms of presence of different source domains and nuisance factors is quite limited. Although recent additions such as DALY [99] and AVA [38] have somewhat improved the situation in terms of variability and number of instances labelled, the realistic validation of action detection methods is still an outstanding issue. AVA is currently the biggest action detection dataset with $1.6M$ label instances, but it is annotated rather sparsely (at a rate of one frame per second).

Overall, the main objective of these datasets is to validate the localisation of human actions in short, untrimmed videos. ROAD, in opposition, goes beyond the detection of actions performed by physical humans to extend the notion of other forms of intelligent agents (e.g., human- or AI-driven vehicles on the road). Furthermore, in contrast with the short clips considered in, e.g., J-HMDB-21 and UCF24, our new dataset is composed of 22 very long videos (around 8 minutes each), thus stressing the dynamical aspect of events and the relationship between distinct but correlated events. Crucially, it is geared towards online detection rather than traditional offline detection, as these videos are streamed in using a vehicle-mounted camera.

2.3. Online action detection

We believe advances in the field of human action recognition[10, 97, 30, 29] can be useful when devising a general approach to the situation awareness problem. We are particularly interested in the *action detection* problem [35, 38, 34, 103], in particular *online* action detection [88], given the incremental processing needs of an autonomous vehicle. Recent work in this area [90, 88, 3, 44, 52, 104] demonstrates very competitive performance compared to (generally more accurate) offline action detection methods [38, 97, 81, 79, 69, 108, 87, 86] on UCF-101-24 [92]. As mentioned, UCF-101-24 is the main benchmark for online action detection research, as it provides annotation in the form of action tubes and every single frame of the untrimmed videos in it is annotated (unlike AVA [38], in which videos are only annotated at one frame per second).

A short review of the state-of-the-art in online action detection is in place. Singh *et al*. [88]'s method was the first one to propose an online, real-time solution to action detection in untrimmed videos, validated on UCF-101-24, and based on an innovative incremental tube construction method. Since then, many other papers [44, 52, 80, 87] have made use of the online tube-construction method in [88]. A common trait of many recent online action detection methods is the reliance on 'tubelet' [79, 44, 52] predictions from a stack of frames. This, however, leads to processing delays proportional to the number of frames in the stack, making these methods not quite applicable in pure online settings. In the case of [79, 44, 52] the frame stack is usually 6-8 frames long, leading to a latency of more than half a second.

For these reasons, inspired by the frame-wise (2D) nature of [88] and the success of the latest single-stage object detectors (such as RetinaNet [55]), we propose a simple extension of [88] towards a '3D-RetinaNet'. The latter is completely online when using a 2D backbone network. One, however, can also insert a 3D backbone to make it even more accurate, while keeping the prediction heads online. We will benchmark our proposed 3D-RetinaNet architecture against the above-mentioned online and offline action detection methods on the UCF-101-24 dataset to show its effectiveness, twinned with its simplicity and efficiency.

3. The Dataset

3.1. A multi-label benchmark

The ROAD dataset is specially designed from the perspective of self-driving cars, and thus includes actions performed not just by humans but by all road agents in specific locations, to form *road events* (REs). REs are annotated by drawing a bounding box around each active road

agent present in the scene, and linking these bounding boxes over time to form 'tubes'. As explained, to this purpose three different types of labels are introduced, namely: (i) the category of *road agent* involved (e.g. *Pedestrian*, *Car*, *Bus*, *Cyclist*); (ii) the *type of action* being performed by the agent (e.g. *Moving away*, *Moving towards*, *Crossing* and so on), and (iii) the *location* of the road user relative the autonomous vehicle perceiving the scene (e.g. *In vehicle lane*, *On right pavement*, *In incoming lane*). In addition, ROAD labels the actions performed by the vehicle itself. Multiple agents might be present at any given time, and each of them may perform multiple actions simultaneously (e.g. a *Car* may be *Indicating right* while *Turning right*). Each agent is always associated with at least one action label.

Table 1. ROAD active agent classes with description.

Label name	Description
Autonomous-vehicle	The autonomous vehicle itself
Car	A car up to the size of a multi-purpose vehicle
Medium vehicle	Vehicle larger than a car, such as van
Large vehicle	Vehicle larger than a van, such as a lorry
Bus	A single or double-decker bus or coach
Motorbike	Motorbike, dirt bike, scooter with 2/3 wheels
Emergency vehicle	Ambulance, police car, fire engine, etc.
Pedestrian	A person including children
Cyclist	A person is riding a push/electric bicycle
Vehicle traffic light	Traffic light related to the AV lane
Other traffic light	Traffic light not related to the AV lane

Agent labels. Within a road scene, the objects or people able to perform actions which can influence the decision made by the autonomous vehicle are termed *agents*. We only annotate *active* agents (i.e., a parked vehicle or bike or a person visible to the AV but located away from the road are not considered to be 'active' agents). Three types of agent are considered to be of interest, in the sense defined above, to the autonomous vehicle: people, vehicles and traffic lights. For simplicity, the AV itself is considered just another agent: this is done by labelling the vehicle's bonnet. People are further subdivided into two sub-classes: pedestrians and cyclists. The vehicle category is subdivided into six sub–classes, listed in Table 1. Finally, the 'traffic lights' category is divided into two sub–classes: *Vehicle traffic light* (if they apply to the AV) and *Other traffic light* (if they apply to other road users). Only one agent label can be assigned to each active agent present in the scene at any given time.

Table 1 summarises the list of ROAD agent labels.

Action labels. Each agent can perform one or more *actions* at any given time instant. For example, a traffic light can only carry out a single action: it can be either red, amber, green or 'black'. A car, instead, can be associated with two action labels simultaneously, e.g., *Turning right* and *Indicating right*. Although some road agents are inherently multitasking, some action combinations can be suitably described by a single label: for example, pushing an object (e.g. a pushchair or a trolley-bag) while walking can be simply labelled as *Pushing object*. The latter was our preferred choice. Note that pedestrians simply walking normally on the pavement are not annotated, as their action is often irrelevant to the autonomous car's decision. All ROAD action labels are listed in Table 2.

AV own actions. Each video frame is also labelled with the action label associated with what the AV is doing. To this end, a bounding box is drawn on the bonnet of the AV. The AV can be assigned one of the following seven action labels: *AV-move*, *AV-stop*, *AV-turn-left*, *AV-turn-right*, *AV-overtake*, *AV-move-left* and *AV-move-right* (see Table 4). Note that these are separate classes only applicable to the AV, with a different semantics than the similar-sounding classes in Table 2. For instance, the regular *Moving* action label means 'moving in the perpendicular direction to the AV', whereas *AV-move* means that the AV is on the move along its normal direction of travel. These labels mirror those used for the autonomous vehicle in the Honda Research Institute Driving Dataset (HDD) [72], hence our choice.

Location labels. Agent *location* is crucial for deciding what action the AV should take next. As the final objective is to assist autonomous decision making, we propose to label the location of each agent from the perspective of the autonomous vehicle. For example, a pedestrian can be found on the right or the left pavement, in the vehicle's own lane, while crossing or at a bus stop. The same applies to other agents and vehicles as well. There is no location label for the traffic lights as they are not movable objects, but agents of a static nature and well-defined location. To understand this concept, Fig. 1 illustrates two scenarios in which the location of the other vehicles sharing the road is depicted from the point of view of the AV. Table 3 shows all the possible locations an agent can be attached to.

3.2. Data collection

ROAD is composed by 22 videos from the publicly available Oxford RobotCar Dataset [56] (OxRD) released in 2017 by the Oxford Robotics Institute[2], covering diverse road scenes under various weather conditions. The OxRD dataset, collected from the narrow streets of the historic city of Oxford, was selected because it presents challenging scenarios for an autonomous vehicle due to the diversity and density of various road users and road events.

Note, however, that our labelling process (described below) is not limited to OxRD. In principle, other autonomous vehicle datasets (e.g. [106, 31]) may be labelled in the same manner to further enrich the ROAD benchmark, as long as long enough videos are provided to capture realistic road event scenarios. We plan to do exactly so in the near future.

[2] http://robotcar-dataset.robots.ox.ac.uk/

Table 2. List of ROAD action labels, with description.

Label name	Description
Moving away	Agent moving in a direction that increases the distance between Agent and AV.
Moving towards	Agent moving in a direction that decreases the distance between Agent and AV.
Moving	Agent moving perpendicular to the traffic flow or vehicle lane.
Reversing	Agent is moving backwards.
Braking	Agent is slowing down, vehicle braking lights are lit.
Stopped	Agent stationary but in ready position to move.
Indicating left	Agent indicating left by flashing left indicator light, or using a hand signal.
Indicating right	Agent indicating right by flashing right indicator light, or using a hand signal.
Hazard lights on	Hazards lights are flashing on a vehicle.
Turning left	Agent is turning in left direction.
Turning right	Agent is turning in right direction.
Moving right	Moving lanes from the current one to the right one.
Moving left	Moving lanes from the current one to the left one.
Overtaking	Agent is moving around a slow-moving user, often switching lanes to overtake.
Waiting to cross	Agent on a pavement, stationary, facing in the direction of the road.
Crossing road from left	Agent crossing road, starting from the left and moving towards the right of AV.
Crossing road from right	Agent crossing road, starting from the right pavement and moving towards the left pavement.
Crossing	Agent crossing road.
Pushing object	Agent pushing object, such as trolley or pushchair, wheelchair or bicycle.
Traffic light red	Traffic light with red light lit.
Traffic light amber	Traffic light with amber light lit.
Traffic light green	Traffic light with green light lit.
Traffic light black	Traffic light with no lights lit or covered with an out-of-order bag.

Table 3. Description of ROAD location labels.

Label name	Description
In vehicle lane	Agent in same road lane as AV.
In outgoing lane	Agent in road lane that should be flowing in the same direction as vehicle lane.
In incoming lane	Agent in road lane that should be flowing in the opposite direction as vehicle lane.
In outgoing bus lane	Agent in the bus lane that should be flowing in the same direction as AV.
In incoming bus lane	Agent in the bus lane that should be flowing in the opposite direction as AV.
In outgoing cycle lane	Agent in the cycle lane that should be flowing in the same direction as AV.
In incoming cycle lane	Agent in the cycle lane that should be flowing in the opposite direction as AV.
On left pavement	Pavement to the left side of AV.
On right pavement	Pavement to the right side of AV.
On pavement	A pavement that is perpendicular to the movement of the AV.
At junction	Road linked.
At crossing	A marked section of road for cross, such as zebra or pelican crossing.
At bus stop	A marked bus stop area on road, or a section of pavement next to a bus stop sign.
At left parking	A marked parking area on left side of the road.
At right parking	A marked parking area on right side of the road.

Video selection. Within OxRD, videos were selected with the objective of ensuring diversity in terms of weather conditions, times of the day and types of scenes recorded. Specifically, the 22 videos have been recorded both during the day (in strong sunshine, rain or overcast conditions, sometimes with snow present on the surface) and at night. Only a subset of the large number of videos available in OxRD was selected. The presence of semantically meaningful content was the main selection criterion. This was done by manually inspecting the videos in order to cover all types of labels and label classes and to avoid 'deserted' scenarios as much as possible. Each of the 22 videos is 8 minutes and 20 seconds long, bar three whose duration is 6:34, 4:10 and 1:37, respectively. In total, ROAD comprises 170

Figure 2. Sample frames and annotation. ROAD's annotated frames cover multiple agents and actions, recorded at different weather conditions (overcast, sun, rain) at different times of the day (morning, afternoon and night). Ground truth bounding boxes and labels can also be appreciated.

Table 4. AV-related action classes and number of frames labelled per class.

Class label	Description	No instances
Av-move	AV on the move	81,196
Av-stop	AV not moving	31,801
Av-turn-right	Av turning right	3,826
Av-turn-left	AV turning left	3,787
Av-overtake	AV overtaking another vehicle	599
Av-move-left	AV moving towards left	537
Av-move-right	AV moving towards right	408
Total	–	**122,154**

minutes of video content. The original list of videos will be made public along with baseline and evaluation code.

Preprocessing. Some preprocessing was conducted. First, the original sets of video frames were downloaded and demosaiced, in order to convert them to red, green, and blue (RGB) image sequences. Then, they were encoded into proper video sequences using `ffmpeg`[3] at the rate of 12 frames per second (fps). Although the original frame rate in the considered frame sequences varies from 11 fps to 16 fps, we uniformised it to keep the annotation process consistent. As we retained the original time stamps, however, the videos in ROAD can still be synchronised with the LiDAR and GPS data associated with them in the RobotCar dataset, allowing future work on multimodal approaches.

3.3. Annotation process

Annotation tool. Annotating tens of thousands of frames rich in content is a very intensive process; therefore, a tool is required which can automate this process in a fast and intuitive manner. Different annotation tools are currently available and might be used to this end, including Vatic[4] by Columbia University, VoTT[5] by Microsoft, or the Automated Driving System Toolbox[6] from MATLAB/Simulink.

For this work, we adopted VoTT. The most useful feature of this annotation tool is that it can copy annotations

[3]https://www.ffmpeg.org/

[4]`http://www.cs.columbia.edu/~vondrick/vatic/`

[5]`https://github.com/Microsoft/VoTT/`

[6]`https://www.mathworks.com/products/automated-driving.html`

(bounding boxes and their labels) from one frame to the next, while maintaining a unique identification for each box, so that boxes across frames are automatically linked together. This feature of VoTT allows us to implicitly link bounding boxes across time, creating the ground-truth tubes needed to evaluate action and event detection methods. Moreover, VoTT also allows for multiple labels, thus lending itself well to ROAD's multi-label annotation concept. Examples of annotated frames from two videos, one captured during the day and one at night, are shown in Fig. 2.

VoTT provides a user-friendly graphical interface which allows annotators to draw boxes around the agents of interest and select the labels they want to associate with them from a predefined list at the bottom. After saving the annotations, the information is stored in a `json` file having the same name as the video. The file structure contains the bounding boxes' coordinates and the associated labels per frame; a unique ID (UID) helps identify boxes belonging to different frames which are part of the same tube. This is important as it is possible to have several instances related to the same kind of action. As a result, the temporal connections between boxes can be easily extracted from this file which is, in turn, crucial to measure performance in terms of video-mAP (see Experiments). It is important to note that tubes are built for each active agent, while the action label associated with a tube can in fact change over time, allowing us to model the complexity of an agent's road behaviour as it evolves over time.

A general rule followed in this work is that only the bottom half of the scene needs to be labelled, as when driving this is our main focus. This can vary somewhat if the road is bending or the autonomous vehicle is turning; in this case, the agents on one side of the picture might be bigger than those on the other side. Therefore, if the object is too far so that it is not possible to distinguish what it is appropriately, then it does not need labelling. Only active agents are labelled (e.g. pedestrians, cyclists, moving / indicating vehicles) - parked vehicles are not considered active, and thus are not labelled.

Event label generation. Using the annotations manually generated for actions and agents in the multi-label scenario as discussed above it is possible to generate *event-level* labels about agents, e.g. *Pedestrian / Moving towards* the AV *On right pavement* or *Cyclist / Overtaking / In vehicle lane*. Any combinations of location, action and agent labels are admissible. If location labels are ignored, the resulting event labels become location-invariant.
In addition to event tubes, in this we work we do explore *agent-action* pair instances (see Sec. 5). Namely, given an agent tube and the continuous temporal sequence of action labels attached to its constituent bounding box detections, we can generate action tubes by looking for changes in the action label series associated with each agent tube. For instance, a *Car* appearing in a video might be first *Moving away* before *Turning left*. The agent tube for the car will then be formed by two contiguous agent-action tubes: a first tube with label pair *Car / Moving away* and a second one with pair *Car / Turning left*.

3.4. Tasks

ROAD is designed as a sandbox for validating the six tasks relevant to situation awareness in autonomous driving outlined in Sec. 1.1. Five of these tasks are detection tasks, while the last one is a frame-level action recognition task sometimes referred to as 'temporal action segmentation' [83].

All detection tasks are evaluated both at frame-level and at video- (tube-)level. *Frame-level detection* refers to the problem of identifying in each video frame the bounding box(es) of the instances there present, together with the relevant class labels. *Video-level detection* consists in regressing a whole series of temporally-linked bounding boxes (i.e., in current terminology, a 'tube') together with the relevant class label. In our case, the bounding boxes will mark a specific active agent in the road scene. The labels may issue (depending on the specific task) either from one of the individual label types described above (i.e., agent, action or location) or from one of the meaningful combinations described in 3.3 (i.e., either agent-action pairs or events).

Below we list all the tasks for which we currently provide a baseline, with a short description.

1. *Active agent detection* (or *agent detection*) is the task which aims at localising an active agent using a bounding box (frame-level) or a tube (video-level) and labelling it with a class label from Table 1.

2. *Action detection* is the task in which we seek to localise an active agent occupied in performing an action from the list of Table 2.

3. In *agent location detection* (or *location detection*) the label attached to the relevant bounding box or tube comes from the list of locations in Table 3.

4. In *agent-action detection* the bounding box or tube is attached a pair agent-action as explained in 3.3. We sometimes refer to this task as 'duplex detection'.

5. *Road event detection* (or *event detection*) consist in assigning to each box or tube a triplet of class labels.

6. *Autonomous vehicle temporal action segmentation* is a frame-level action classification task in which each video frame is assigned a label from the list of possible AV own actions (Table 4). We refer to this task as 'AV-action segmentation', similarly to [83].

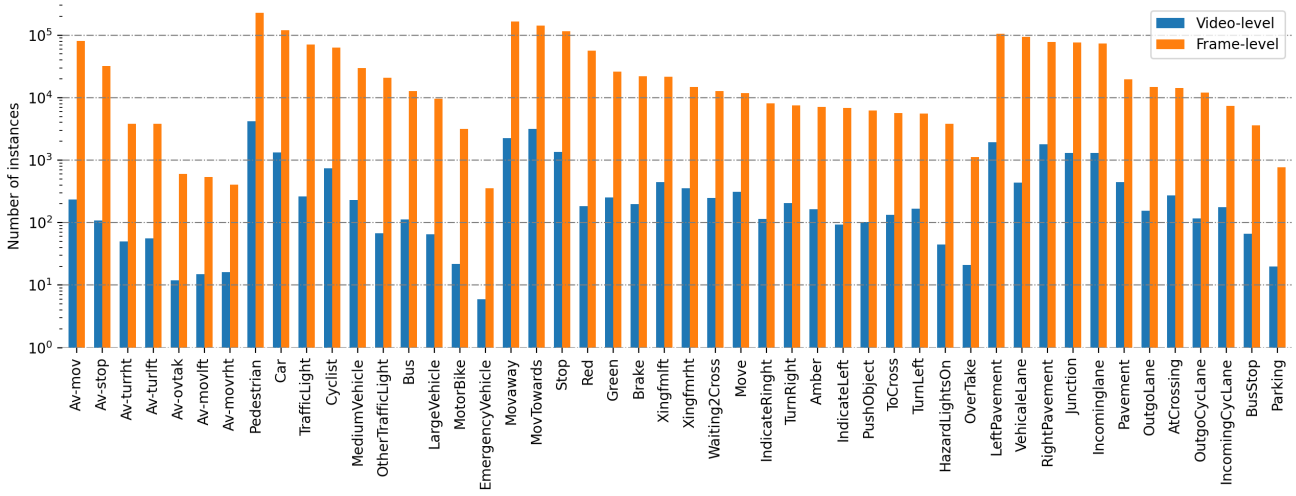

Figure 3. Number of instances of each class of individual label-types in logarithmic scale.

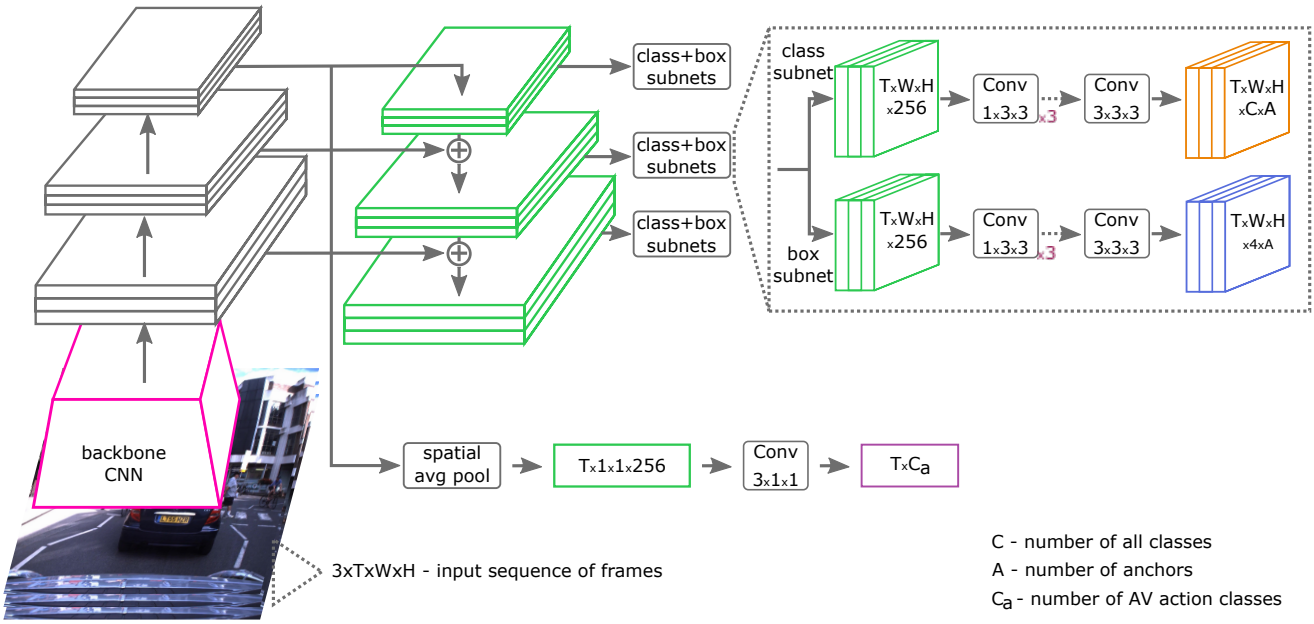

Figure 4. Proposed 3D-RetinaNet architecture for online video processing.

3.5. Quantitative summary

Overall, $122K$ frames extracted from 22 videos were labelled, in terms of both AV own actions (attached to the entire frame) and bounding boxes with attached one or more labels of each of the three types: agent, action, location. In total, ROAD includes $560K$ bounding boxes with $1.7M$ instances of individual labels. The latter figure can be broken down into $560K$ instances of agent labels, $640K$ instances of action labels, and $499K$ instances of location labels. Based on the manually assigned individual labels, we could identify $603K$ instances of duplex (agent-action) labels and $454K$ instances of triplets (event labels).

The number of instances for each individual class from the three lists is shown in Fig. 3 (frame-level, in orange). The $560K$ bounding boxes make up $7,029$, $9,815$, $8,040$, $9,335$ and $8,394$ tubes for the label types agent, action, location, agent-action and event, respectively. Figure 3 also shows the number of tube instances for each class of individual label types as number of video-level instances (in blue).

4. Tube Detection Baseline

Inspired by the success of recent 3D CNN architectures [10] for video recognition and of feature-pyramid networks (FPN) [54] with focal loss [55], we propose a simple yet effective 3D feature pyramid network (3D-FPN) with

focal loss as a baseline method for ROAD's detection tasks. We call this architecture *3D-RetinaNet*.

4.1. 3D-RetinaNet architecture

The data flow of 3D-RetinaNet is shown in Figure 4. The input is a sequence of T video frames. As in classical FPNs [54], the initial block of 3D-RetinaNet consists of a backbone network outputting a series of forward feature pyramid maps, and of lateral layers producing the final feature pyramid composed by T feature maps. The second block is composed by two sub-networks which process these features maps to produce both bounding boxes (4 coordinates) and C classification scores for each anchor location (over A possible locations). In the case of ROAD, the integer C is the sum of the numbers of agent, action, location, action-agent (duplex) and agent-action-location (event) classes, plus one reserved for an *agentness* score. The extra class agentness is used to describe the presence or absence of an active agent. As in FPN [54], we adopt ResNet50 [39] as the backbone network.

2D versus 3D backbones. In our experiments we show results obtained with both the original (2D) ResNet50 and with an inflated 3D (I3D) ResNet50, in the manner also explained in [97]. Choosing a 2D backbone makes the detector completely online [88], with a delay of a single frame. Choosing an I3D backbone network, instead, causes a 4-frame delay at detection time. Note that, as an I3D network makes use of a max-pool layer with stride 2, the initial feature pyramid in the second case contains $T/2$ feature maps. Nevertheless, in this case we can simply linearly upscale the output to T feature maps.

AV action prediction heads. In order for the method to also address the prediction of the AV's own actions (e.g. whether the AV is stopping, moving, turning left etc.), we branch out the last feature map of the pyramid (see Fig. 4) and apply spatial average pooling, followed by a temporal convolution layer. The output is a score for each of the C_a classes of AV actions, for each of the T input frames.

Loss function. As for the choice of the loss function, we adopt a binary cross-entropy-based focal loss [55]. We choose a binary cross entropy because our dataset is multi-label in nature. The choice of a focal-type loss is motivated by the expectation that it may help the network deal with long tail and class imbalance (see Figure 3).

4.2. Online tube generation via agentness score

The autonomous driving scenario requires any suitable method for agent, action or event tube generation to work in an *online* fashion, by incrementally updating the existing tubes as soon as a new video frame is captured. For this reason, this work adopts a recent algorithm proposed by Singh *et al*. [88], which incrementally builds action tubes in an online fashion and at real-time speed. To be best of our knowledge, [88] was the first online multiple action detection approach to appear in the literature, and was later adopted by almost all subsequent works [44, 52, 87] on action tube detection.

Linking of detections. We now briefly review the tube-linking method of Singh *et al*. [88], and show how it can be adapted to build agent tubes based on an 'agentness' score, rather than build a tube separately for each class as proposed in the original paper. This makes the whole detection process faster, since the total number of classes is much larger than in the original work [88]. The proposed 3D-RetinaNet is used to regress and classify detection boxes in each video frame potentially containing an active agent of interest. Subsequently, detections whose score is lower than 0.025 are removed and non-maximal suppression is applied based on the agentness score.

At video start, each detection initialises an agentness tube. From that moment on, at any time instance t the highest scoring tubes in terms of mean agentness score up to $t-1$ are linked to the detections with the highest agentness score in frame t which display an Intersection-over-Union (IoU) overlap with the latest detection in the tube above a minimum threshold λ. The chosen detection is then removed from the pool of frame-t detections. This continue until the tubes are either assigned or not assigned a detection from current frame. Remaining detections at time t are used to initiate new tubes. A tube is terminated after no suitable detection is found for n consecutive frames. As the linking process takes place, each tube carries scores for all the classes of interest for the task at hand (e.g., action detection rather than event detection), as produced by the classification subnet of 3D-RetinaNet. We can then label each agentness tube using the k classes that show the highest mean score over the duration of the tube.

Temporal trimming. Most tubelet based methods [44, 52, 84] do not perform any temporal trimming of the action tubes generated in such a way (i.e., they avoid deciding when they should start or end). Singh *et al*. [88] proposed to pose the problem in a label consistency formulation solved via dynamic programming. However, as it turns out, temporal trimming [88] does not actually improve performance, as shown in [87], except in some settings, for instance in the DALY [99] dataset.

The situation is similar for our ROAD dataset as opposed to what happens on UCF-101-24, for which temporal trimming based on solving the label consistency formulation in terms of the actionness score, rather than the class score, does help improve localisation performance. Therefore, in our experiments we only use temporal trimming on the UCF-101-24 dataset but not on ROAD.

5. Experiments

In this section we present results on the various task the ROAD dataset is designed to benchmark (see Sec. 3.4), as well as the action detection results delivered by our 3D-RetinaNet model on UCF-101-24 [42, 91].

We first present the evaluation metrics and implementation details specific to ROAD in Section 5.1. In Section 5.2 we benchmark our 3D-RetinaNet model for the action detection problem on UCF-101-24. The purpose is to show that our proposed baseline model is competitive with the current state of the art in action tube detection while only using RGB frames as input, and to provide a sense of how challenging ROAD is when compared to standard action detection benchmarks. Indeed, the complex nature of the real-world, non-choreographed road events, often involving large numbers of actors simultaneously responding to a range of scenarios in a variety of weather conditions makes ROAD a dataset which poses significant challenges when compared to other, simpler action recognition benchmarks.

In Section 5.3 we illustrate and discuss the baseline results on ROAD for the different tasks (Sec. 5.3.2) and under different training/testing splits encoding different weather conditions (Sec. 5.3.3). In particular, in Sec. 5.3.4 we show the results one can obtain when predicting composite labels as products of single-label predictions as opposed to training a specific model for them, as this can provide a crucial advantage in terms of efficiency, as well as give the system the flexibility to be extended to new composite labels without retraining. Finally, in Sec. 5.3.5 we report our baseline results on the temporal segmentation of AV actions.

5.1. Implementation details

The results are evaluated in terms of both frame-level bounding box detection and of tube detection. In the first case, the evaluation measure of choice is *frame mean average precision* (f-mAP). We set the Intersection over Union (IoU) detection threshold to 0.5 (signifying a 50% overlap between predicted and true bounding box). For the second set of results we use *video mean average precision* (video-mAP), as information on how the ground-truth BBs are temporally connected is available. These evaluation metrics are standard in action detection [82, 98, 44, 88, 53].
We also evaluate actions performed by AV, as described in 3.1. Since this is a temporal segmentation problem, we adopt the mean average precision metric computed at frame-level, as standard on the Charades [83] dataset.

We use sequences of $T = 8$ frames as input to 3D-RetinaNet. Input image size is set to 512×682. This choice of T is the result of GPU memory constraints; however, at test time, we unroll our convolutional 3D-RetinaNet for sequences of 32 frames, showing that it can be deployed in a streaming fashion. We initialise the backbone network with weights pretrained on Kinetics [45]. For training we use an SGD optimiser with step learning rate. The initial learning rate is set to 0.01 and drops by a factor of 10 after 18 and 25 epochs, up to an overall 30 epochs. For tests on the UCF-101-24 dataset the learning rate schedule is shortened to a maximum 10 epochs, and the learning rate drop steps are set to 6 and 8.

The parameters of the tube-building algorithm (Sec. 4.2) are set by cross validation. For ROAD we obtain $\lambda = 0.5$ and $k = 4$. For UCF-101-24, we get $\lambda = 0.25$ and $k = 4$. Temporal trimming is only performed on UCF-101-24.

Table 5. Comparison of the action detection performance (frame-mAP@0.5 (f-mAP) and video-mAP at different IoU thresholds) of the proposed 3D-RetinaNet baseline model with the state-of-the-art on the UCF-101-24 dataset.

Methods / δ =	f-mAP	0.2	0.5	0.75	0.5:0.9
RGB + FLOW methods					
MR-TS Peng *et al*. [69]	–	73.7	32.1	00.9	07.3
FasterRCNN Saha *et al*. [82]	–	66.6	36.4	07.9	14.4
SSD + OJLA Behl *et al*. [3]*	–	68.3	40.5	14.3	18.6
SSD Singh *et al*. [88]*	–	76.4	45.2	14.4	20.1
AMTnet Saha *et al*. [79]*	–	78.5	49.7	22.2	24.0
ACT Kalogeiton *et al*. [44]*	–	76.5	49.2	19.7	23.4
TraMNet Singh *et al*. [87]*	–	79.0	50.9	20.1	23.9
Song *et al*. [89]	72.1	77.5	52.9	21.8	24.1
Zhao *et al*. [108]	–	78.5	50.3	22.2	24.5
I3D Gu *et al*. [37]	76.3	–	**59.9**	–	–
Li *et al*. [52]*	**78.0**	**82.8**	53.8	**29.6**	**28.3**
RGB only methods					
RGB-SSD Singh *et al*. [88]*	65.0	72.1	40.6	14.1	18.5
RGB-AMTNet Saha *et al*. [80]*	–	75.8	45.3	19.9	22.0
3D-RetinaNet / 2D (ours)*	65.2	73.5	48.6	22.0	22.8
3D-RetinaNet / I3D (ours)	**75.2**	**82.4**	**58.2**	**25.5**	**27.1**

* online methods

Table 6. Splits of training, validation and test sets for the ROAD dataset with respect to weather conditions. The table shows the number of videos in each set or split. For splits, the first figure is the number of training videos, the second number that of validation videos.

Condition	sunny	overcast	snow	night
Training and validation	7	7	1	3
Split-1	7/0	4/3	1/0	3/0
Split-2	7/0	7/0	1/0	0/3
Split-3	4/3	7/0	1/0	3/0
Testing	1	1	1	1

5.2. Baseline performance on UCF-101-24

We first benchmarked our 3D-RetinaNet on UCF-101-24 [42, 91], using the corrected annotations from [88]. We evaluated both frame-mAP and video-mAP and provided a comparison with state-of-the-art approaches in Table 5. We can see that our baseline model (3D-RetinaNet) is competitive with the current state-of-the-art [52, 37], even as those methods use both RGB and optical flow as input, as opposed to ours. As shown in the bottom part of Table 5, our

3D-RetinaNet outperforms all methods that rely solely on appearance (RGB) by large margins.

3D-RetinaNet appears to be a strong baseline method, while retaining the simplicity of single-stage methods, and sporting the flexibility of being able to be reconfigured by changing the backbone architecture. As a final remark, we note that it could be further boosted using the simple optimisation technique proposed in [47].

5.3. Experimental results on ROAD

5.3.1 Three splits: modelling weather variability

For the benchmarking of the ROAD tasks, we divided the dataset into two sets. The first set contains 18 videos for training and validation purposes, while the second set contains 4 videos for testing, equally representing the four types of weather conditions encountered.

The group of training and validation videos is further subdivided into three different ways ('splits'). In each split, 15 videos are selected for training and 3 for validation. Details on the number of videos for each set and split are shown in Table 6. All 3 validation videos for Split-1 are overcast; 4 overcast videos are also present in the training set. As such, Split-1 is designed to assess the effect of different overcast conditions. Split-2 has all 3 night videos in the validation subset, and none in the training set. It is thus designed to test model robustness to day/night variations. Finally, Split-3 contains 4 training and 3 validation videos for sunny weather: it is thus designed to evaluate the effect of different sunny conditions, as camera glare can be an issue when the vehicle is turning or facing the sun directly.

Note that there is no split to simulate a bias towards snowy conditions, as the dataset only contains one video of that kind. The test set (bottom row) is more uniform, as it contains one video from each environmental condition.

5.3.2 Results on the various tasks

Results are reported for the tasks discussed in Section 3.4.

Frame-level results across the five detection tasks are summarised in Table 7 using the frame-mAP (f-mAp) metric, for a detection threshold of $\delta = 0.5$. The reported figures are averaged across the three splits described above, in order to assess the overall robustness of the detectors to domain variations. Performance within each split is evaluated on both the corresponding validation subset and test set. Each row in the Table shows the result of a particular combination of backbone network (2D or I3D) and test-time sequence length (in number of frames, 8 and 32). Frame-level results vary between 16.8% (events) and 65.4% (agentness). Clearly, for each detection task except agentnness (which amounts to object detection) the performance is quite lower than the 75.2% achieved by our 3D baseline network on UCF-101-24 (Table 5, last row). Once again, this is due to the numerous nuisance factors present in ROAD, such as significant camera motion, weather conditions, etc. For a fair comparison, we should note that there are only 11 agent classes, as opposed to e.g. 23 action classes and 15 location classes. *Video-level results* are reported in terms of video-mAP in Table 8. As for the frame-level results, tube detection performance (see Sec. 4.2) is averaged across the three splits. One can appreciate the similarities between frame- and video-level results, which follow a similar trend. Again, results are reported for different backbone networks and sequence lengths. Video-level results at detection threshold δ equal to 0.2 vary between a minimum of 17.9% (events) to a maximum of 27.1% (locations), compared to the 82.4% achieved on UCF-101-24. For a detection threshold δ equal to 0.5, the video-level results lie between 4.6% (actions) and 10.2% (agents) compared to the 58.2% achieved on UCF-101-24 for the same IoU threshold. The difference is quite dramatic, and highlights the order of magnitude of the challenge involved by perception in autonomous driving compared to a standard benchmark portraying only human actions. Furthermore, we can notice a few important facts.

Streaming deployment. Increasing test sequence length from 8 to 32 does not much impact performance. This indicates that, even though the network is trained on 8-frame clips, being fully convolutional (including the heads in the temporal direction), it can be easily unrolled to process longer sequences at test time, making it easy to deploy in a streaming fashion. Being deployable in an incremental

Table 7. Frame-level results (mAP %) averaged across the three splits of ROAD. The models considered differ in terms of backbone network (2D and I3D) and clip length (08 vs 32). Detection threshold $\delta = 0.5$. Both validation and test performance are reported for each entry.

Model	Agentness	Agents	Actions	Locations	Duplexes	Events
2D-08	51.8/63.4	30.9/39.5	15.9/22.0	23.2/30.8	18.1/25.1	10.6/12.8
2D-32	52.4/64.2	31.5/**39.8**	16.3/22.6	23.6/31.4	18.7/25.8	10.8/13.0
I3D-08	52.3/65.1	32.2/39.5	19.3/25.4	24.5/34.9	21.5/30.8	12.3/16.5
I3D-32	52.7/**65.4**	32.3/39.2	19.7/**25.9**	24.7/**35.3**	21.9/**31.0**	12.6/**16.8**

Table 8. Video-level results (mAP %) averaged across the three ROAD splits. The models considered differ in terms of backbone network (2D and I3D) and test time clip length (08 vs 32). Both validation and test performance are reported for each entry.

Model	Agents	Actions	Locations	Duplexes	Events
Detection threshold $\delta = 0.2$					
2D-08	22.2/25.1	10.3/13.9	18.2/24.8	16.1/21.9	12.8/14.7
2D-32	22.6/25.0	11.2/14.5	18.5/25.9	16.2/22.7	13.0/15.3
I3D-08	23.2/26.5	14.1/15.8	20.8/25.8	21.1/24.0	14.9/17.4
I3D-32	24.4/**26.9**	14.3/**17.5**	21.3/**27.1**	21.4/**25.5**	15.9/**17.9**
Detection threshold $\delta = 0.5$					
2D-08	8.9/7.5	2.3/3.0	5.2/6.1	6.5/6.1	5.1/5.3
2D-32	8.3/8.0	2.7/3.3	5.6/7.1	6.3/6.8	5.0/5.7
I3D-08	9.2/9.6	4.0/4.3	5.8/6.9	7.2/7.4	4.6/5.4
I3D-32	9.7/**10.2**	4.0/**4.6**	6.4/**7.7**	7.1/**8.3**	4.8/**6.1**

Table 9. Number of video- and frame-level instances for each label (individual or composite), left. Corresponding frame-/video-level results (mAP@%) for each of the three ROAD splits (right). Val-n denotes the validation set for Split n. Results produced by an I3D backbone.

	Number of instance					Frame-mAP@0.5/Video-mAP@0.2					
Train subset	#Boxes/#Tubes					Train-1		Train-2		Train-3	
Eval subset	All	Val-1	Val-2	Val-3	Test	Val-1	Test	Val-2	Test	Val-3	Test
Agent	559142/7029	60103/781	79119/761	83750/809	82465/1138	44.5/30.1	34.0/25.7	17.2/16.0	40.9/27.4	35.3/27.1	42.6/27.5
Action	639740/9815	69523/1054	89142/1065	95760/1111	94669/1548	26.2/17.0	26.6/17.4	11.7/11.4	25.3/17.3	21.2/14.6	25.7/17.9
Location	498566/8040	56594/851	67116/864	77084/914	70473/1295	34.9/28.6	35.2/26.4	13.7/12.1	33.9/26.3	25.4/23.2	36.7/28.6
Duplex	603274/9335	60000/965	85730/1032	88960/1050	89080/1471	28.2/25.3	28.7/23.4	13.6/17.3	31.4/24.8	23.9/21.6	33.0/28.4
Event	453626/8394	43569/883	65965/963	72152/967	64545/1301	17.7/18.6	15.9/15.8	6.4/11.8	16.4/18.9	13.7/17.2	18.1/18.9
	Number of instances					Frame-AP					
AV-action	122154/490	17929/67	18001/56	16700/85	20374/82	57.9	45.7	33.5	43.6	43.7	48.2

fashion is a must for autonomous driving applications; this is a quality that other tubelet-based online action detection methods [44, 87, 52] fail to exhibit, as they can only be deployed in a sliding window fashion.

Impact of the backbone. Broadly speaking, the inflated 3D version of the backbone performs, as expected, much better than the 2D version. An I3D backbone can particularly help with tasks which require the system to 'understand' movement, e.g. when detecting actions, agent-actions pairs and road events (triplets).

Level of task challenge. The overall results on event detection (last column in both Table 7 and Table 8) are encouraging, but they remain in the low 10-20s, indicating that situation awareness in road scenario is an extremely challenging task.

Comparison across tasks. From a superficial comparison of the mAPs obtained, action detection seems to perform worse than agent-action detection or even event detection. However, the headline figures are not really comparable since, as we know, the number of class per task varies. More importantly, within-class variability is often lower for composite labels. For example, the score for *Indicating right* is really low, whereas *Car / Indicating-right* has much better performance (see Supplementary material, Tables 7–9 for class-specific performance). This is because the within-class variability of the pair *Car / Indicating-right* is much lower than that of *Indicating right*, which puts together instances of differently-looking types of vehicles (e.g. buses, cars and vans) all indicating right. Interestingly, results on agents are comparable among the four baseline models (expecially for f-mAP and v-mAP at 0.2, see Tables 7 and 8).

Validation vs test results. Results on the test set are, on average, superior to those on the validation set. This is because the test set includes data from all weather/visibility conditions (see Table 6), whereas for each split the validation set only contains videos from a single weather condition. E.g., in Split 2 all validation videos are nighttime ones.

Table 10. Comparison of joint vs product of marginals approaches with I3D backbone. Number of video-/frame-level instances for each composite label ('No instances' column) and corresponding frame-/video-level results (mAP@%) averaged across all three splits, on both validation and test sets.

	No instances	Frame-mAP@0.5/Video-mAP@0.2			
Eval-method		Joint		Prod. of marginals	
Eval subset	All	Val	Test	Val	Test
Duplexes	603274/9335	21.9/21.4	31.0/25.5	21.6/21.2	30.8/24.3
Event	453626/8394	12.6/15.9	16.8/17.9	13.7/15.4	16.3/16.1

5.3.3 Results under different weather conditions

Table 9 shows, instead, the results obtained under the three different splits we created on the basis of the weather/environmental conditions of the ROAD videos, discussed in Section 5.3.1 and summarised in Table 6. Note that the total number of instances (boxes for frame-level results or tubes for video-level ones) of the five detection tasks is comparable for all the three splits.

We can see how Split-2 (for which all three validation videos are taken at night and no nighttime videos are used for training, see Table 6) has the lowest validation results, as seen in Table 9 (Train-2, Val-2). When the network trained on Split-2's training data is evaluated on the (common) test set, instead, its performance is similar to that of the networks trained on the other splits (see Test columns). Split-1 has three overcast videos in the validation set, but also four overcast videos in the training set. The resulting network has the best performance across the three validation splits. Also, under overcast conditions one does not have the typical problems with night-time vision, nor glares issues as in sunny days. Split-3 is in a similar situation to Split-1, as it has sunny videos in both train and validation sets.

These results seem to attest a certain robustness of the baseline to weather variations, for no matter the choice of the validation set used to train the network parameters (represented by the three splits), the performance on test data (as long as the latter fairly represents a spectrum of weather conditions) is rather stable.

5.3.4 Joint versus product of marginals

One of the crucial points we wanted to test is weather the manifestation of composite classes (e.g., agent-action pairs or road events) can be estimated by separately training models for the individual types of labels, to then combine the resulting scores by simple multiplication (under an implicit, naive assumption of independence). This would have the advantage of not having to train separate networks on all sort of composite labels, an obvious positive in terms of efficiency, especially if we imagine to further extend in the future the set of labels to other relevant aspects of the scene, such as attributes (e.g. vehicle speed). This would also give the system the flexibility to be extended to new composite events in the future without need for retraining.

For instance, we may want to test the hypothesis that the score for the pair *Pedestrian / Moving away* can be approximated as $P_{Ag}(\text{Pedestrian}) \times P_{Ac}(\text{Moving away})$, where P_{Ag} and P_{Ac} are the likelihood functions associated with the individual agent and action detectors[7]. This boils down to testing whether we need to explicitly learn a model for the joint distribution of the labels, or we can approximate that joint as a product of marginals. Learning-wise, the latter task involves a much smaller search space, so that marginal solutions (models) can be obtained more easily.

Table 10 compares the detection performance on composite (duplex or event) labels obtained by expressly training a detection network for those ('Joint' column) as opposed to simply multiplying the detector scores generated by the networks trained on individual labels ('Prod. of marginals'). The results clearly validate the hypothesis that it is possible to model composite labels using predictions for individual labels without having to train on the former. In most cases, the product of marginals approach achieves results similar or even better than those of joint prediction, although in some case (e.g. *Traffic light red* and *Traffic light red*, see Supplementary material again) we can observe a decrease in performance. We believe this to be valuable insight for further research.

5.3.5 Results of AV-action segmentation

Finally, Table 11 shows the results of using 3D-RetinaNet to temporally segment AV-action classes, averaged across all three splits on both validation and test set. As we can see, the results for classes *AV-move* and *AV-stop* are very good, we think because these two classes are predominately present in the dataset. The performance of the 'turning' classes is reasonable, but the results for the bottom three classes are really disappointing. We believe this is mainly due the fact that the dataset is very heavily biased (in terms of number of instances) towards the other classes. As we do intend to further expand this dataset in the future by including more and more videos, we hope the class imbalance issue can be mitigated over time. A measure of performance weighing mAP using the number of instances per class could be considered, but this is not quite standard in the action detection literature. At the same time, ROAD provides an opportunity for testing methods designed to address class imbalance.

[7]Technically the networks output scores, not probabilities, but those can be easily calibrated to probability values.

Table 11. AV-action temporal segmentation results (frame mAP%) averaged across all three splits.

	No instances	Frame-mAP@0.5			
Model		I3D		2D	
Eval subset	All	Val	Test	Val	Test
Av-move	81196/233	92.0	96.6	83.0	87.8
Av-stop	31801/108	92.2	98.5	65.3	68.4
Av-turn-right	3826/50	46.1	63.0	35.0	57.7
Av-turn-left	3787/56	69.0	59.8	55.1	42.9
Av-overtake	599/12	4.9	1.1	2.7	2.5
Av-move-left	537/15	0.5	0.8	0.5	0.5
Av-move-right	408/16	10.5	0.6	4.0	2.0
Total/Mean	122154/490	45.0	45.8	35.1	37.4

Once again the performance of the inflated (I3D) version of our model is much better than that of the 2D backbone version, showing that by simply switching backbone one can improve on performance or other desirable properties, such as speed (as in SlowFast [30] or X3D [29]), or make the 3D CNN encoding intrinsically online, as in RCN [85].

6. Future Extensions

The results of Section 5 demonstrate the challenging nature of the ROAD dataset and its usefulness for testing various aspects of the generalisation ability of a road event detector. The baseline we provide is competitive with the current state-of-the-art, providing a benchmark for others to build upon and compare against, opening up further developments in this area. By design, ROAD, however, is an open project which we expect to evolve and grow over time. From a pure data collection and curation point of view, in the near future we will work towards completing the multi-label annotation process for a larger number of frames coming from videos spanning an even wider range of road conditions. Further down the line, we will work to extend the benchmark to other cities, countries and sensor configurations, to slowly grow towards an even more robust, 'in the wild' setting. Most importantly, as we mentioned in the Introduction, ROAD is geared up to be extended to a variety of additional, challenging Tasks closely related to situation awareness in autonomous driving.

Detection of complex road activities. While the purpose of this paper is to propose ROAD as a benchmark for road event detection, the dataset's concept is designed to allow the validation of methods for the detection of *complex road activities*, composed by a variable number of distinct road

Table 12. Tentative list of complex activities with their brief description

Activity name	Description
Negotiating intersection	AV stops at intersection/junction, waits for other vehicles to pass and then resumes moving.
Negotiating pedestrian crossing	AV stops at pedestrian crossing signal, waits for pedestrian(s) and cycles to cross the road, then resumes moving.
Waiting in a queue	AV stops and waits in a queue for a signal (either pedestrian crossing or junction traffic light, as they both look similar from the AV's perspective).
Merging into vehicle lane	AV stops and waits for bus/car to merge into the vehicle lane from the opposite side of the road.
Sudden appearance	A pedestrian/cycle/vehicle suddenly appears in front of the vehicle, coming from a side street.
Walking in the middle of the road	A pedestrian walking in the middle of the road in front of the AV, causing the AV to slow down or stop.

events performed by various road agents over a certain period of time [94]. The most common driving situations can be described in those terms. For instance, we can define the class *Negotiating intersection* as made up of the following 'atomic' events: *AV-move + Vehicle traffic light / Green + AV-stop + Vehicle(s) / Stopped / At junction+ AV-move*. An initial list of complex road activities to be selected for inclusion in ROAD, compiled by thorough visual inspection of the 22 videos, is shown in Table 12. We do provide annotation for the complex activities in this list in terms of start and end time stamp. While this annotation can be used for straightforward temporal segmentation, in the near future we aim to provide a baseline for this Task which exploits a spatiotemporal graph representation of complex road activities and to propose an evaluation protocol requiring the successful detection of not just the overall activity but also of its constituent events.

Event anticipation/intent prediction. ROAD is an oven-ready playground for action and event anticipation algorithms, a topic of growing interest in the vision community [48, 49], as it allows researchers to test predictions of both future event labels and future event locations, both spatial and temporal. Anticipating the future behaviour of other road agents is crucial to empower the AV to react timely and appropriately. The output of this Task should be in the form of one or more future tubes, with the scores of the associated class labels and the future bounding box locations in the image plane [86]. Clearly, ROAD already provides the required annotation for testing action and event prediction capabilities. In the near future we plan to also propose a baseline method for this Task, but we encourage researchers in the area to start engaging with the dataset from now.

Autonomous decision making. Similarly, we will work to design and share a baseline for AV decision making from intermediate semantic representations, in accordance to our overall philosophy. The output of this Task should be the decision made by the AV in response to a road situation [40], represented as a collection of events as defined in this paper. As the action performed by the AV at any given time is part of the annotation, the necessary meta-data is already there. Although we did provide a simple temporal segmentation baseline for this task seen as a classification problem, we intend in the near future to propose a baseline from a decision making point of view, making use of the intermediate semantic representations produced by the detectors. We also plan to run a comparative analysis of decision making based on intermediate representation versus a direct reinforcement learning approach (as, for instance, the existing annotation can already be employed for imitation learning).

Machine theory of mind [71] refers to the attempt to provide machines with (limited) ability to guess the reasoning process of other intelligent agents they share the environment with. Building on our efforts in this area [20], we will work with teams of psychologists and neuroscientists to provide annotations in terms of mental states and reasoning processes for the road agents present in ROAD. Note that theory of mind models can also be validated in terms of how close the predictions of agent behaviour they are capable of generating are to their actual observed behaviour. Assuming that the output of a theory of mind model is intention (which is observable and annotated) the same baseline as for event anticipation can be employed.

Continual event detection. ROAD's conceptual setting is intrinsically incremental, one in which the autonomous vehicle keeps learning from the data it observes, in particular by updating the models used to estimate the intermediate semantic representations. The videos forming the dataset are particularly suitable, as they last 8 minutes each, providing a long string of events and data to learn from. To this end, we plan to set a protocol for the continual learning of event classifiers and detectors and propose ROAD as the first continual learning benchmark in this area [66].

7. Conclusions

This paper outlines a possible strategy and a learning framework for online road event detection in autonomous driving. As a milestone in this direction we have presented the ROad event Awareness Dataset for Autonomous Driving (ROAD), encompassing 22 videos with 122K annotated video frames, for a total of 560K detection bounding boxes associated with 1.7M individual labels. ROAD is designed to test a robot–car's situation awareness capabilities, namely its ability to detect (and, in the future, predict) 'road events', each intended as a moving agent performing a relevant action in a specific road location. ROAD has been constructed by providing extra annotation to a fraction of the Oxford RobotCar dataset [56]. The annotation process follows a multi–label approach in which road agents (including the AV), their locations as well as the action they

perform (possibly more than one) are labelled separately. Event-level labels can be generated by simply composing lower-level descriptions.

Baseline tests were conducted on ROAD using a new 3D-RetinaNet architecture inspired by the state of the art in both online action detection and object detection. Both frame–mAP and video–mAP were used as evaluation measures of the action detection strategy in urban environments. Our preliminary results indicate that the proposed 3D-RetinaNet architecture is able to detect a road event with 16.8% frame-mAP at $\delta = 0.5$ and 17.9% video-mAP at $\delta = 0.2$. These numbers are encouraging and support the need for an even broader analysis, while highlighting the significant challenges specific to the road scene setting.

Our dataset is extensible to a number of challenging tasks associated with situation awareness in autonomous driving, and we pledge to further enrich it in the near future. We also intend to propose to organise a Workshop (possibly a series of events) at the next suitable upcoming major computer vision (e.g. ICCV, CVPR) and robotics (e.g. ICRA, IROS) conferences, in order to start mobilising both communities towards the further development of the benchmark and associated array of tasks and baselines.

References

[1] S. Armstrong and S. Mindermann. Occam's razor is insufficient to infer the preferences of irrational agents. In *Advances in Neural Information Processing Systems*, volume 31, pages 5603—-5614, 2018.

[2] S. Azam, F. Munir, A. Rafique, Y. Ko, A. M. Sheri, and M. Jeon. Object modeling from 3d point cloud data for self-driving vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 409–414, June 2018.

[3] Harkirat S Behl, Michael Sapienza, Gurkirt Singh, Suman Saha, Fabio Cuzzolin, and Philip HS Torr. Incremental tube construction for human action detection. *arXiv preprint arXiv:1704.01358*, 2017.

[4] Massimo Bertozzi, Alberto Broggi, and Alessandra Fascioli. Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, 32(1):1 – 16, 2000.

[5] José-Luis Blanco-Claraco, Francisco-Ángel Moreno-Dueñas, and Javier González-Jiménez. The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. *The International Journal of Robotics Research*, 33(2):207–214, 2014.

[6] Alex Broggi, Alberto et al. Christian Laugier. Intelligent vehicles. In *Springer Handbook of Robotics*, pages 1627–1656. Springer, 2016.

[7] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *European conference on computer vision*, pages 44–57, 2008.

[8] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, 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.

[9] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.

[10] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733, 2017.

[11] Rohan Chandra, Uttaran Bhattacharya, Aniket Bera, and Dinesh Manocha. Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8483–8492, 2019.

[12] John Chang, Ming-Fang Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and others et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[13] Zhengping Che, Guangyu Li, Tracy Li, Bo Jiang, Xuefeng Shi, Xinsheng Zhang, Ying Lu, Guobin Wu, Yan Liu, and Jieping Ye. D ^{2}-city: A large-scale dashcam video dataset of diverse traffic scenarios. *arXiv preprint arXiv:1904.01975*, 2019.

[14] J. Chen, C. Tang, L. Xin, S. E. Li, and M. Tomizuka. Continuous decision making for on-road autonomous driving under uncertain and interactive environments. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1651–1658, June 2018.

[15] Yiping Chen, Jingkang Wang, Jonathan Li, Cewu Lu, Zhipeng Luo, Han Xue, and Cheng Wang. Lidar-video driving dataset: Learning driving policies effectively. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5870–5878, 2018.

[16] Yukyung Choi, Namil Kim, Soonmin Hwang, Kibaek Park, Jae Shin Yoon, Kyounghwan An, and In So Kweon. Kaist multi-spectral day/night data set for autonomous and assisted driving. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):934–948, 2018.

[17] Matthias Codevilla, Felipe Alexey et al. Dosovitskiy. End-to-end driving via conditional imitation learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9, 2018.

[18] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of CVPR 2016*, pages 3213–3223, 2016.

[19] Vincent T Covello and Miley W Merkhofer. An evaluation of the state of the art. In *Risk Assessment Methods*, pages 239–265. Springer, 1993.

[20] Fabio Cuzzolin, Andrea Morelli, Bogdan Cirstea, and Barbara J. Sahakian. Knowing me, knowing you: Theory of

mind in AI. *Psychological Medicine*, 50(7):1057–1061, May 2020.

[21] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[22] Li Ding, Jack Terwilliger, Rini Sherony, Bryan Reimer, and Lex Fridman. MIT DriveSeg (Manual) Dataset, 2020.

[23] Markus Enzweiler and Dariu M Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2179–2195, 2008.

[24] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[25] E. Galceran et al. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction. *Theory and experiment. Auton. robots*, 41(6), 2017.

[26] L. Fridman et al. Arguing machines: Perception-control system redundancy and edge case discovery in real-world autonomous driving. ArXiv preprint ArXiv:1710.04459, 2017.

[27] L. Zhao et al. Ontology-based driving decision making: A feasibility study at uncontrolled intersections. *IEICE T INF SYST.*, 100:1425—-1439, 2017.

[28] Z. Fang and A. M. López. Is the pedestrian going to cross? answering by 2d pose estimation. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1271–1276, June 2018.

[29] 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.

[30] 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.

[31] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[32] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[33] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, et al. A2d2: Aev autonomous driving dataset. *Note: http://www. a2d2. audi Cited by*, 1(4), 2019.

[34] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. A better baseline for ava. *arXiv preprint arXiv:1807.10066*, 2018.

[35] G Gkioxari and J Malik. Finding action tubes. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2015.

[36] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense, 2017.

[37] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions, 2018.

[38] Chunhui Gu, Chen Sun, Sudheendra Vijayanarasimhan, Caroline Pantofaru, David A Ross, George Toderici, Yeqing Li, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. *arXiv preprint arXiv:1705.08421*, 2017.

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[40] Constantin Hubmann, Marvin Becker, Daniel Althoff, David Lenz, and Christoph Stiller. Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1671–1678. IEEE, 2017.

[41] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3192–3199, 2013.

[42] YG Jiang, J Liu, A Roshan Zamir, G Toderici, I Laptev, M Shah, and R Sukthankar. Thumos challenge: Action recognition with a large number of classes. *http://crcv.ucf.edu/THUMOS14*, 2014.

[43] Hojung Jung, Yuki Oto, Oscar M Mozos, Yumi Iwashita, and Ryo Kurazume. Multi-modal panoramic 3d outdoor datasets for place categorization. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4545–4550. IEEE, 2016.

[44] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *Proc. Int. Conf. Computer Vision*, 2017.

[45] Joao Kay, Will Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, and others et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[46] R Kesten, M Usman, J Houston, T Pandya, K Nadhamuni, A Ferreira, M Yuan, B Low, A Jain, P Ondruska, et al. Lyft level 5 av dataset 2019. *urlhttps://level5. lyft. com/dataset*, 2019.

[47] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, and Jianbo Shi. Consistent optimization for single-shot object detection, 2019.

[48] Yu Kong, Zhiqiang Tao, and Yun Fu. Deep sequential context networks for action prediction. In *Proceedings of the*

IEEE conference on computer vision and pattern recognition, pages 1473–1481, 2017.

[49] Yu Kong, Zhiqiang Tao, and Yun Fu. Adversarial action prediction networks. *IEEE transactions on pattern analysis and machine intelligence*, 42(3):539–553, 2018.

[50] Kirsten Korosec. Toyota is betting on this startup to drive its self-driving car plans forward. Available at: `http://fortune.com/2017/09/27/toyota-self-driving-car-luminar/`.

[51] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[52] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[53] Zhaofan Li, Dong et al. Recurrent tubelet proposal and recognition networks for action detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 303–318, 2018.

[54] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017.

[55] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[56] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[57] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[58] Srikanth Malla, Behzad Dariush, and Chiho Choi. Titan: Future forecast using action priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11186–11196, 2020.

[59] Markus et al. Maurer. *Autonomous driving: technical, legal and social aspects*. Springer Nature, 2016.

[60] Raúl Quintero Mínguez, Ignacio Parra Alonso, David Fernández-Llorca, and Miguel Ángel Sotelo. Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition. *IEEE Transactions on Intelligent Transportation Systems*, 20(5):1803–1814, 2018.

[61] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–8, 2019.

[62] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4990–4999, 2017.

[63] Lukáš Neumann, Michelle Karg, Shanshan Zhang, Christian Scharfenberger, Eric Piegert, Sarah Mistr, Olga Prokofyeva, Robert Thiel, Andrea Vedaldi, Andrew Zisserman, et al. Nightowls: A pedestrians at night dataset. In *Asian Conference on Computer Vision*, pages 691–705. Springer, 2018.

[64] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*, pages 3153–3160. IEEE, 2011.

[65] Gaurav Pandey, James R. McBride, and Ryan M. Eustice. Ford campus vision and lidar data set. *International Journal of Robotics Research*, 30(13):1543–1552, 2011.

[66] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[67] Abhishek Patil, Srikanth Malla, Haiming Gang, and Yi-Ting Chen. The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9552–9557. IEEE, 2019.

[68] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *European conference on computer vision*, pages 452–465. Springer, 2010.

[69] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream r-cnn for action detection. In *European Conference on Computer Vision*, pages 744–759, 2016.

[70] Quang-Hieu Pham, Pierre Sevestre, Ramanpreet Singh Pahwa, Huijing Zhan, Chun Ho Pang, Yuda Chen, Armin Mustafa, Vijay Chandrasekhar, and Jie Lin. A* 3d dataset: Towards autonomous driving in challenging environments. *arXiv preprint arXiv:1909.07541*, 2019.

[71] Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *International conference on machine learning*, pages 4218–4227. PMLR, 2018.

[72] Vasili Ramanishka, Yi-Ting Chen, Teruhisa Misu, and Kate Saenko. Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[73] Amir Rasouli, Iuliia Kotseruba, Toni Kunic, and John K Tsotsos. Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6262–6271, 2019.

[74] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 206–213, 2017.

[75] A. Rasouli and J. K. Tsotsos. Autonomous vehicles that interact with pedestrians: A survey of theory and practice. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):900—-918, 2020.

[76] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 6517—-6525, 2017.

[77] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[78] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras. Human motion trajectory prediction: A survey. arXiv preprint arXiv1905.06113, 2019.

[79] Suman Saha, Gurkirt Singh, and Fabio Cuzzolin. Amtnet: Action-micro-tube regression by end-to-end trainable deep architecture. In *Proc. Int. Conf. Computer Vision*, 2017.

[80] Suman Saha, Gurkirt Singh, and Fabio Cuzzolin. Two-stream amtnet for action detection. *arXiv preprint arXiv:2004.01494*, 2020.

[81] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. *arXiv preprint arXiv:1608.01529*, 2016.

[82] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip H. S. Torr, and Fabio Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. In *British Machine Vision Conference*, 2016.

[83] 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, 2018.

[84] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[85] Gurkirt SingH and Fabio Cuzzolin. Recurrent convolutions for causal 3d cnns. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[86] Gurkirt Singh, Suman Saha, and Fabio Cuzzolin. Predicting action tubes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[87] Gurkirt Singh, Suman Saha, and Fabio Cuzzolin. Tramnet-transition matrix network for efficient action tube proposals. In *Asian Conference on Computer Vision*, pages 420–437. Springer, 2018.

[88] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3637–3646, 2017.

[89] Lin Song, Shiwei Zhang, Gang Yu, and Hongbin Sun. Tacnet: Transition-aware context network for spatio-temporal action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11987–11995, 2019.

[90] Khurram Soomro, Haroon Idrees, and Mubarak Shah. Predicting the where and what of actors and actions through online action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2648–2657, 2016.

[91] K Soomro, AR Zamir, and M Shah. Ucf101: a dataset of 101 human action classes from videos in the wild (2012). *arXiv preprint arXiv:1212.0402*, 2012.

[92] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012.

[93] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset, 2019.

[94] Jinhui Tang, Xiangbo Shu, Rui Yan, and Liyan Zhang. Coherence constrained graph lstm for group activity recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[95] P. Wang, C. Chan, and A. d. L. Fortelle. A reinforcement learning based approach for automated lane change maneuvers. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1379–1384, June 2018.

[96] Peng Wang, Xinyu Huang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[97] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[98] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, June 2015.

[99] Philippe Weinzaepfel, Xavier Martin, and Cordelia Schmid. Human action localization with sparse spatial supervision. *arXiv preprint arXiv:1605.05197*, 2016.

[100] John Winn and Jamie Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 37–44. IEEE, 2006.

[101] Christian Wojek, Stefan Walk, and Bernt Schiele. Multi-cue onboard pedestrian detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 794–801. IEEE, 2009.

[102] Christian Wolf, Julien Mille, Eric Lombardi, Oya Celiktutan, Mingyuan Jiu, Moez Baccouche, Emmanuel Dellandréa, Charles-Edmond Bichot, Christophe Garcia, and Bülent Sankur. The LIRIS Human activities dataset and the ICPR 2012 human activities recognition and localization competition. Technical report, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/École Centrale de Lyon, 2012.

[103] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019.

[104] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–272, 2019.

[105] Yu Yao, Mingze Xu, Chiho Choi, David J Crandall, Ella M Atkins, and Behzad Dariush. Egocentric vision-based future vehicle localization for intelligent driving assistance systems. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9711–9717. IEEE, 2019.

[106] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.

[107] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3221, 2017.

[108] Jiaojiao Zhao and Cees GM Snoek. Dance with flow: Two-in-one stream action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9935–9944, 2019.