**Introduction**:

1. ROad event Awareness Dataset (ROAD): the first of this kind of dataset. Comprises 22 videos, originally from Oxford RobotCat dataset

2. Dataset purpose: detect road events. Triplets: (1) a moving agent; (2) action it performs; (3) corresponding scene locations.

3. Baseline incremental algorithm: inflating RetinaNet along time (3D-RetinaNet), mean average precision: 16.8% for frame-level, 6.1% for video level, at 50% overlap.

4. Possible tasks: (1) complex (road) activity detection; (2) future road event anticipation; (3) modelling of sentient road agents in terms of mental states.

5. Detector only sees what is on the road, not understands the scene context.

**Dataset Characteristics**: multi-label, multi-task

1. Road Events (REs) triplets: E = (Ag, Ac, Loc), three classes. Res are represented as ‘tubes’, i.e., time series of frame-wise bounding box detection.

2. size: 122k labelled video frames, 560K detection bounding boxes. 1.7M unique individual labels: 560K agent labels, 640K action label, 499K location labels.

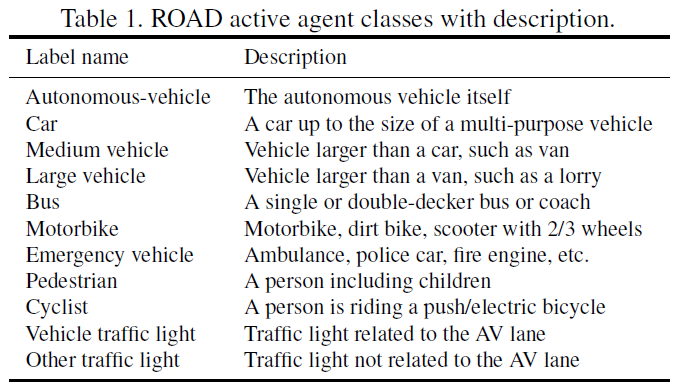
3. Dataset design principle: (1) each RE consists of triplet labels; (2) each RE can have multiple instances of same label type, i.e., action label, ‘*moving away*’ and ‘*turning left*’.

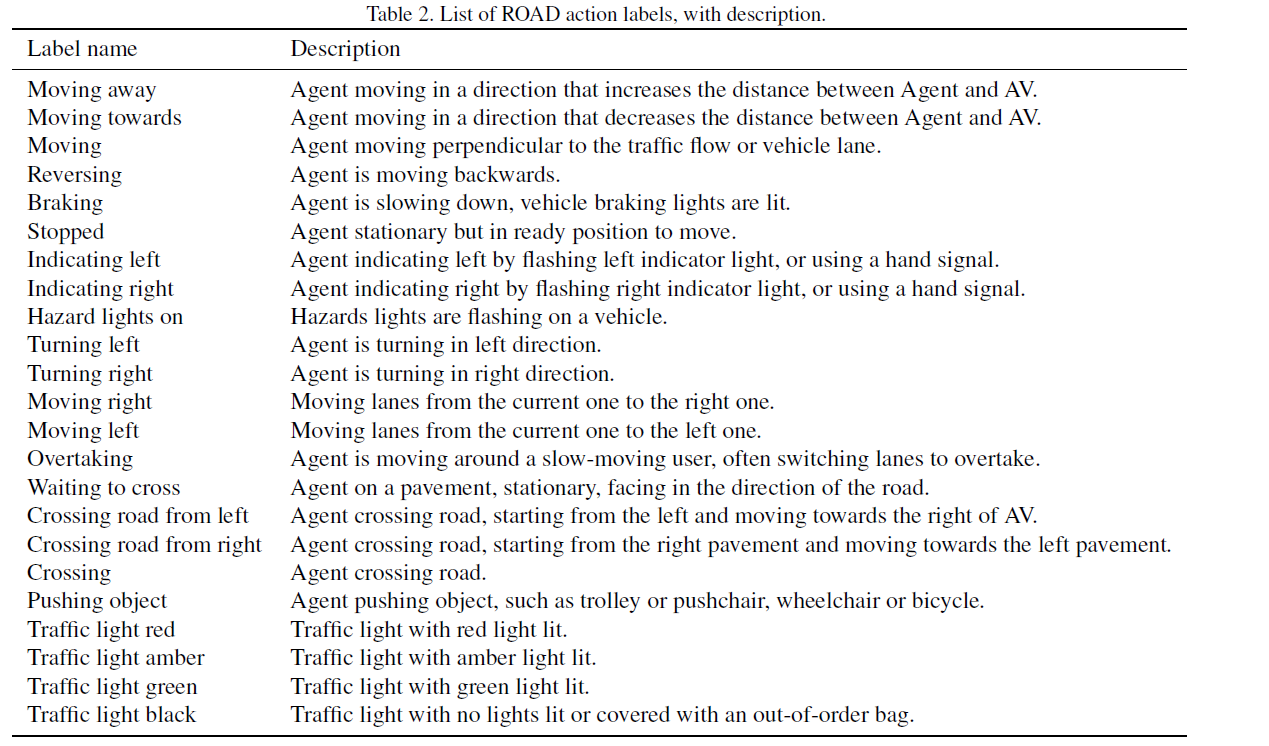
4. Online, incremental detection.

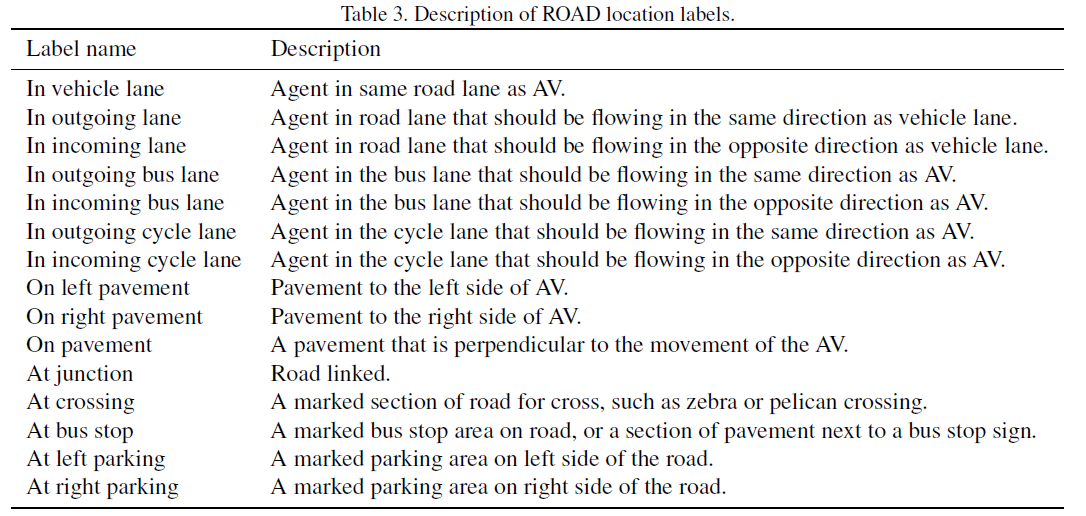
**Dataset Details:**

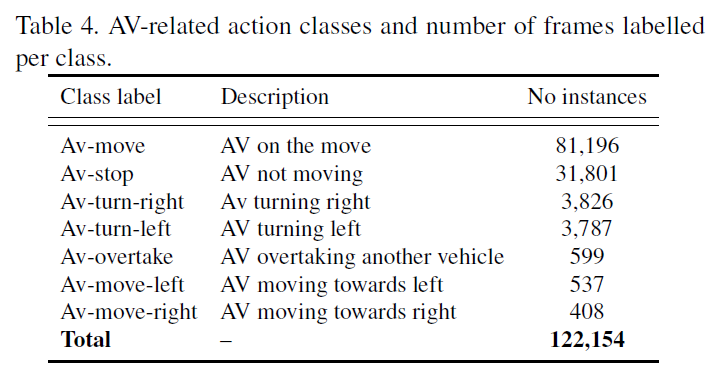
1. REs is annotated by a bounding box around each agent, links the boxes over time to form ‘tubes’

2. (1) *agent*: only active agent considered and not consider pedestrian simply walking normally on the pavement. pedestrian, car, bus, cyclist, etc. (2) *action*: each agent can perform one or more actions, moving away, moving towards, crossing, etc. AV own action is seperated (3) *location*: in vehicle lane, on right pavement, in incoming lane, etc.









3. data selection: weather conditions, times of the day, types of scenes.

4. example:



5. single modal dataset: only images without Lidar, Radar, GPS/IMU data

**Tasks**:

1. tasks: (1)-(5) are detection tasks, (6) is frame-level action recognition task. Spatiotemporal.

(1) agent detection: localize an active agent using a bound box (frame-level) or a tube (video-level), label it with a class label from Table 1.

(2) action detection: localize active agent occupied in performing an action from Table 2.

(3) location detection: attach labels from Table 3 to relevant bounding box or tube

(4) agent-action detection: attach bounding box or tube to a pair agent-action.

(5) road event detection: attach a triplet of labels to each bounding box or tube.

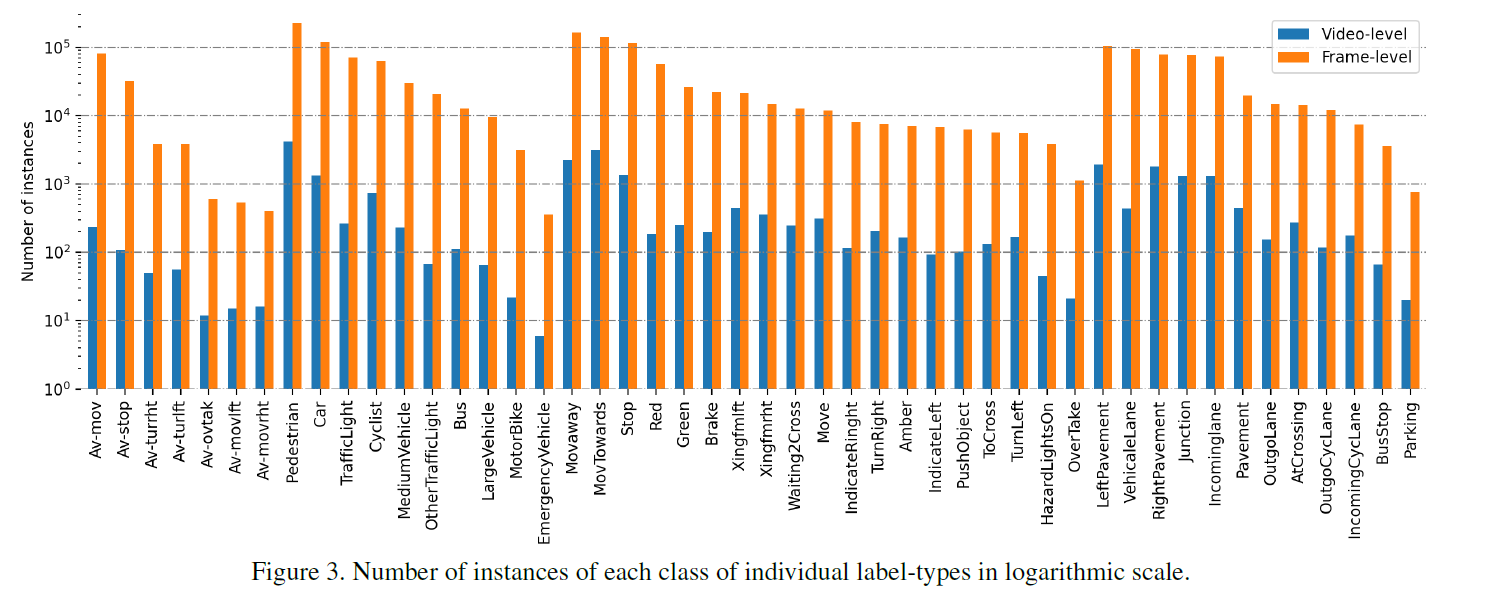
(6) temporal segmentation of AV actions: frame-level action classification, assign a label from Table 4.

2. frame-level detection: identifying the bounding boxes of the instances in each video frame, together with relevant class labels.

3. video-level detection: regress a whole series of temporally-linked bounding boxes (tubes) together with the relevant class labels.

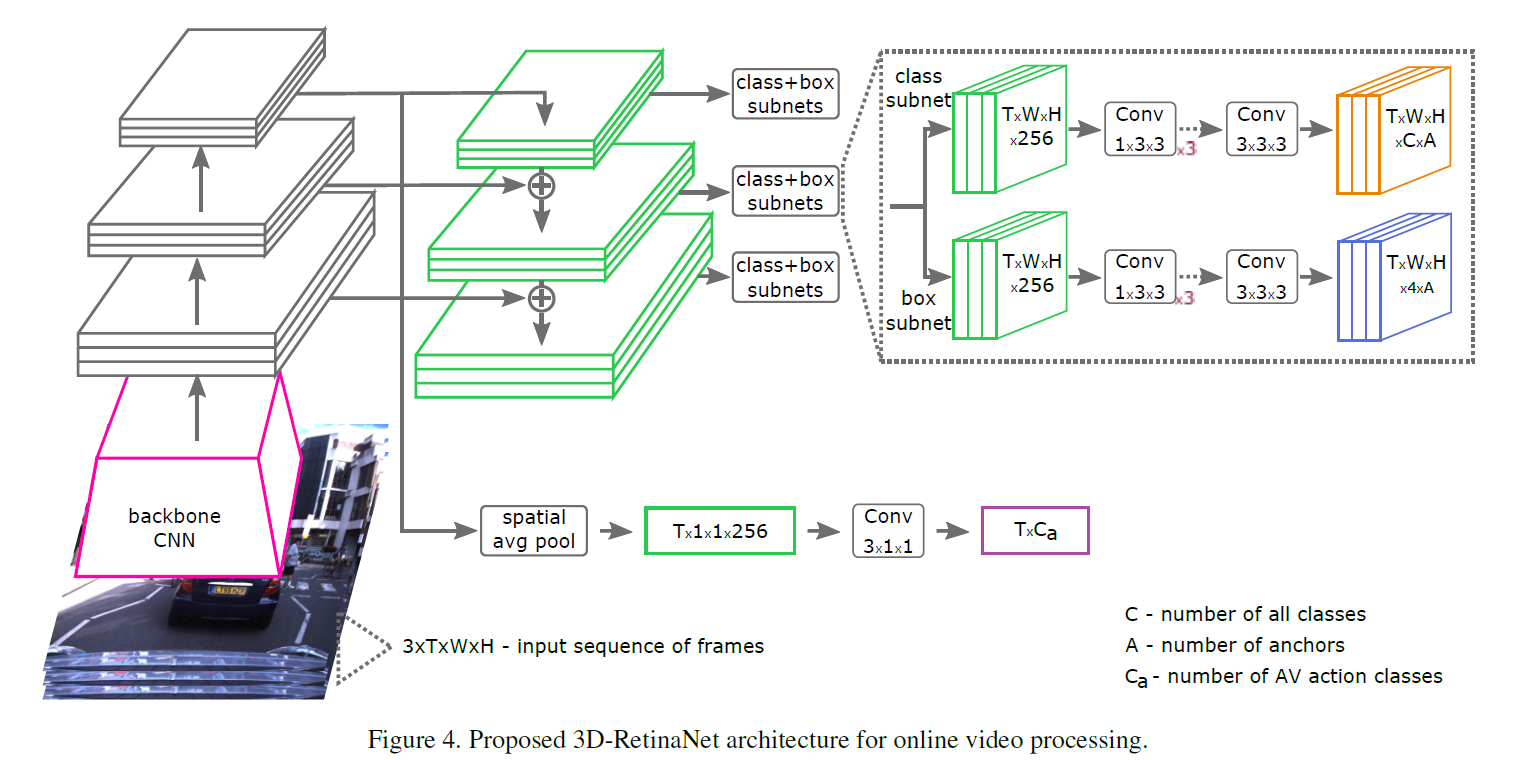
4. dataset distribution:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Videos | Video length | Frames | Bounding boxes | Total Individual labels |
| 22 | 8min | 122K | 560K | 1.7M |
| Agent labels | Action labels | Location labels | Agent-action labels | Triplet event labels |
| 560K | 640K | 499K | 603K | 454K |



**Baseline model**:

1.3D-RetinaNet: 3D feature pyramid network, 3D-FPN with focal loss.



Backbone CNN (grey blocks, ResNet50): output a series of forward feature pyramid maps. Can be 2D (ResNet50) or 3D (Inflated 3D ResNet50, I3D) backbones.

Lateral layers (first three green blocks): produce final feature pyramid composed by T feature maps.

Two-stage detection (dot line blocks): (1) stage one: output bounding box with four coordinates; (2) stage two: output classification scores for each anchor location (A possible locations).

Bottom block: AV action prediction heads. From last feature map of the pyramid.

Loss function: binary cross-entropy-based focal loss because dataset is multi-label nature? Another purpose to use focal loss is to expect the network can deal with long tail and class imbalance. Note: class-based focal loss might be better than focal loss.