

## 1.2 COMMON DEFINITIONS

The following definitions will be used throughout the document:

- **Ego Car:** The car that collected the data, i.e. the car on which the cameras and the LIDAR are mounted.
- **Ego Lane:** The lane in which the Ego Car is driving. If there are no lane markings, it corresponds to the straight path that can be extrapolated from the travelling direction of the Ego Car.
- **Ego Road:** The road where the Ego Car is driving, including the shoulder/emergency lane, intersections and roundabouts, as well as entrances/exits on highways and to parking lots, but not the parking lot itself (unless the Ego Car is in the parking lot). If there is a road edge in the form of a road divider (e.g. a barrier or fence), a curb (e.g. to a sidewalk) or a patch of grass, then the Ego Road refers only to the part of the road that is on the same side of the road edge as the Ego Car. If there is no visible road divider, then the Ego Road refers to the entire road, including both travelling directions. If there is no road edge, e.g. separating the road and a walking/bicycle area, then the Ego Road includes that walking/bicycle area.
- **Road Edge:** The edge where the Ego Road ends and a new surface not directly intended for driving begins. Such an edge can be towards many types of materials. For example, the road edge could be at the transition to entities such as guardrails and wire-railings, barriers of e.g concrete or stone, road surface edges to e.g gravel, sand, grass or snow.
- **Drivable surface:** Any surface not already marked as Ego Road, which is reachable by the ego-car in the frame, where it can drive without a substantial damage. This includes any surface type e.g. asphalt, concrete, snow, ice, gravel, sand, small vegetation, small curbs, etc. Things that are excluded from drivable surface are objects larger than 15x15x15 cm **and** with rigid body. E.g. a plastic bag should be annotated as drivable while a shopping cart should not.
- **Conventional Road:** Road on which most traffic situations occur, e.g. highways, country roads, city streets. **Non-conventional** road categories include parking lots, shared pedestrian/car areas, very low speed residential areas, garages, indoor/underground parking lots, driveways, yards. The distinction is relevant for annotating debris, which should not be annotated on non-conventional roads.
- **Scene:** A single image from one of the surround cameras and the associated points from the LIDAR, i.e. only the segment of the LIDAR point cloud that is associated with the corresponding image plane.
- **Scene Sequence:** A series of consecutive Scenes in which objects can be tracked.
- **360 Scene:** A full set of images from all surround cameras from the same moment and the associated LIDAR points from a complete 360-degree LIDAR sweep.
- **360 Scene Sequence:** A series of consecutive 360 Scenes in which objects can be tracked.

## 7 Landmarks

*Vision and LIDAR annotations of landmarks*

## 7.1 INTRODUCTION

This document contains guidelines and requirements for annotating landmarks (such as traffic signs or guide posts) in **camera images** and associated **LIDAR point clouds**. The annotations consist of **two- and three-dimensional markings** as well as **properties**.

**Note:**

All landmarks that are **visible in the camera image** should be annotated, i.e. those visible to the human eye on the camera field of view (if necessary - using image enhancement tools like histogram equalization, brightness and contrast correction, etc.), no matter how small, occluded, blurred or truncated they are, even if they have no corresponding 3D annotation.

However, the following landmarks do not count as visible even if they exist in corresponding lidar point clouds or data from other sensors:

- Landmarks fully occluded by other objects in the frame, unless some of their pixels are visible e.g. through windows of a vehicle which is occluding the landmark.
- Landmarks fully occluded by water droplets or dirt on the camera lens, in the way that there is no hint of their existence in the image.
- Landmarks hidden in the dark/overexposed areas of the image which cannot be seen even after using image enhancement tools (like for example histogram equalization, brightness and contrast correction, etc.).

If there are any doubts whether a landmark is visible or not, it is preferred to annotate it with property `unclear` set to `true` over leaving it unannotated. Check the [Landmarks properties](#) chapter for details of that property usage.

This also means that only landmarks that are present in any of the camera images should be annotated in the point cloud, and there should be no 3D annotations in the point cloud outside of the combined field of view of all cameras available in the Scene. Note, that area of the camera frustum occluded by the camera housing or the body of the ego vehicle does **not** count as the field of view, so no objects should be annotated there, neither in the image space nor in the 3D point clouds. Similarly, if an object can be seen in the point cloud inside the combined field of view of all cameras, but is fully occluded by some other object in the scene and hence not visible in any of the images, the 3D annotation should also be left out.

2D annotations of the landmarks and their properties should only be based on what is visible in the particular frame. No context should be transferred from the neighboring surround camera images or data from other sensors (like e.g. LIDAR point clouds), as well as from previous and following sequential timestamps. The only exception is the estimation of the sizes of the bounding boxes for the occluded and truncated objects - it is correct to use any available information to estimate those sizes accurately.

**Note:**

- There is a difference between annotation of frames and annotation of sequences. **For frames**, there should be no landmarks with only a 3D marking, since we only want to annotate objects that are visible in the image. However, **for sequences** landmarks that are only visible in the LIDAR point cloud should be included with only a 3D marking (the object still has to be inside the camera field of view). An example would be that in a sequence, a landmark is first visible in the image and LIDAR, thus it has both a 2D and 3D marking. Then in the next frame it is occluded in the image but still visible in the LIDAR, in which case only the 3D marking would be added (and no marking if this frame would not be in a sequence). Finally, in the next frame when the landmark is visible in the image again it will have a 2D and 3D marking again. It is also important that the `annotation_uuid` of a particular landmark is the same in all samples it occurs throughout a sequence.
- For annotation tasks on multiple images from the same scene (like Scenes from fisheye cameras used for "fisheye\_static\_objects\_3d" project), the `annotation_uuid` of the landmarks should be kept consistent in all images where

the landmark is visible.

### 7.1.1 DATA FORMAT

The data will consist of Scenes with single or multiple camera images from the cameras mounted on the Ego Car, together with associated LIDAR point cloud, all taken at approximately the same timestamp.

The data may also contain Scene Sequences, i.e. time-sequential Scenes.

Typically all provided data should be annotated. However, for selected tasks only a subset of provided data will be requested to be annotated. For such tasks a list will be provided indicating precisely which images, Scenes and LIDAR point clouds should be annotated.

### 7.1.2 ANNOTATION FORMAT

The annotations should be delivered in a frame-by-frame fashion. The annotations of a frame should be provided as a JSON-formatted file `xyz.json`, where `xyz` would correspond to the `scene_id` of the frame. The ID commonly consists of a vehicle identifier, two timestamps, and finally a camera/source name.

A **schema**-file is provided for validation, and three example jsons. For landmarks, there are two paths for supplying Cosmoswith the annotations:

1. Joint 3D+2D information, as in example `landmarks_annotation_example.json`.
2. Separate 3D and 2D information, as in examples `landmarks_annotation_example_2d-only.json` and `landmarks_annotation_example_3d-only.json`.

### 7.1.3 PROPOSED WORKFLOW

If a landmark is present in the image (using image enhancement tools, if necessary), follow the steps:

1. Assign `annotation_uuid`
2. Assign the two-dimensional pixel positions
3. Is the landmark present in the point cloud?
  - Yes → Assign a three-dimensional bounding box and its rotation in the point cloud
  - No → Do not assign a three-dimensional bounding box
4. Is it hard/impossible to see what class of landmark it is (or multiple indistinguishable landmarks)?
  - Yes → Assign `class="Inconclusive"` and skip rest of steps
  - No → Assign the correct `class` and go to the next step
5. Is it an unclear landmark (i.e hard to set properties confidently)?
  - Yes → Set property `unclear=true` and skip rest of the steps
  - No → Set property `unclear=false` and go to the next step
6. Assign landmark properties

If a landmark is outside of the camera view or fully occluded by the camera housing or the body of the ego vehicle, but can be “seen” in the point cloud, **do not annotate it**.

## 7.2 LANDMARK DEFINITIONS

{"TrafficSign", "TrafficGuide", "TrafficBeacon", "TrafficSignal", "DynamicBarrier", "Inconclusive"}

A landmark is an object or feature that is easily seen and recognized from a distance, especially one that enables someone to establish their location. For traffic situations, this can be signs or lights. The landmarks that should be annotated are:

- **Traffic Sign** - A traffic sign contains symbols and/or text that relates to the current traffic condition or any restriction. It is commonly upheld by a pole, however, the pole should be excluded from the sign's annotation marking.
- **Traffic Guide** - A traffic guide is designed to be highly visible and occur in multiples to form lines or rows. This is meant to help guide the traffic flow. Common occurrences are on/beside barriers, on construction sites, or to indicate the road edge for snowy roads.
- **Traffic Beacon** - A traffic beacon is a single light source with a continuous state: always on, always off, or flashing. These can be seen as restrictions, for example at pedestrian crossings, or as a warning for a certain traffic condition, such as construction site. Note that only the light should be annotated, and the pole should be excluded.
- **Traffic Signal** - A configuration of one or several lights where the state of the lights indicate a certain driving restriction or condition. The lights may be arranged vertically, horizontally, or in some mixed configuration.
- **Dynamic Barrier** - A dynamic barrier control the flow of cars by completely blocking the path. They are usually pole-like objects that can be positioned either horizontally to block the passage of vehicles, or vertically to allow flow of traffic.

More detailed explanations and image examples are provided below. Note that only traffic-related landmarks should be annotated. Yet if there is a landmark that has the form of a traffic landmark, a sign or a beacon for example, but too blurry to see if it is related to traffic or not, the landmark can be annotated and set as `unclear=true`.

Important to note is that **any artificial object appearing on a poster/billboard or as a reflection should not be marked at all.**

### 7.2.1 TRAFFIC SIGN

A *traffic sign* contains symbols and/or text that relates to the current traffic condition or any restriction. Note that if the sign is upheld by a post, **the actual post should be excluded from the sign's markings**

Some additional information:

- All signs related to traffic must be annotated.
- Annotations are to be added on a signboard-basis, thus for multiple signboards attached to the same signpost, each sign board should be annotated as its own traffic sign. This holds even though they might have a shared context. Note that there is a special property for supplementary/complementary signs called `complementary_to_landmark`.
- Multiple traffic signs printed on the same signboard must get one joint annotation.
- Traffic signs seen from behind or in such a way that the front side is not visible should also be marked and specified with the annotation property `traffic_content_visible="False"`.
- The traffic sign class includes signs for other traffic participants than vehicles, for example pedestrian or bicycle signs. Note that these should be set with the corresponding property `is_for_other_traffic_participant="True"`.

**What to include:** Warning triangles (temporary during car failure), variable message signs (LED displays - even when turned off), invalidated signs (taped over for example), informational signs, navigational signs, tourist attraction signs, signs in tunnels, etc.

**What to exclude:** Advertisement billboards and signs. Anything not related to traffic.

Some examples of what to include (non-exhaustive) are:



More examples of what to include are:



A warning triangle sign.



Tunnel-related signs.



Tourist or attraction signs.



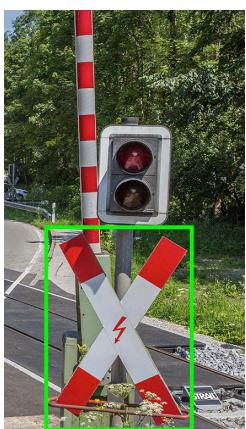
A sign mounted on a bollard, where only the actual sign (blue part) should be included in the sign's marking.



Some traffic signals have signs inside the traffic signal board. In these cases, the sign should be included but annotated separately from the traffic signal. The sign's 2D and 3D markings should start below the light parts of the traffic signal as shown in the example.



Road signs related to construction work should be**included**.



Railroad crossing signs, even though they look a bit special, should be**included**.



Smaller electronic signs containing small symbols such as an arrow or cross should also be considered as signs and annotated as such.

Examples to exclude:



Advertisement billboards (even if they look odd) should be**excluded**.



Another example of Advertisement billboards that should be excluded.

## 7.2.2 TRAFFIC GUIDE

A *traffic guide* is designed to be highly visible and occur in multiples to form lines or rows. This is meant to help guide the traffic flow. Common occurrences are on/beside barriers, on construction sites, or to indicate the road edge for snowy roads.

Some additional specification:

- Vertical structure.
- A height of around 0 to 2 meters.
- Designed to have high visibility (e.g. using saturated colors, high-contrast colors or reflectors).
- Meant to guide the traffic flow.
- Usually occurs in multiples forming lines.
- Also occurs on objects requiring attention, for example on overpasses or beginning of road-side barriers.
- The stickers on the vehicles should be excluded.

The types of traffic guides are:

- Reflector: A reflector placed on a pole or on top of a barrier on the side of the road.
- Attention: Highly reflective guides, which uses line segments or arrows to indicate direction or danger. These line segments can be vertical, left-leaning diagonal, right-leaning diagonal, or horizontal. These guides are commonly seen at exits, extreme curves (chevron-like), construction areas, tunnels/overpasses, and more. These can also occur on dynamic barriers, cones, and poles.
- SnowMarker: High, thin, reflective poles on side of the road used to show where to drive if the road is covered with snow.

Examples of these traffic guides can be seen below.

Type	Image example
Reflector	

Type	Image example				
Attention	             				
SnowMarker	  				

Below are also examples of edge cases, both those that should be included and excluded.



Similar patterns as `type="Attention"` that occur on overpasses or the entrance of tunnels should be **included**, however, with property `odd=true`.



There are also indications of height restrictions in the form of multiples of sticks with attention patterns on them. These should be treated the same as the above example for bridges and overpasses, and thus be **included** as a single landmark, with `type="Attention", odd=true`.



**Exclude** attention markers that are part of the pole (left image), however, if the marker is on its own board mounted on the pole (right image), then it should be **included**.



**Exclude** very small reflectors mounted on top of barriers.



Additionally for reflector-like objects, small reflectors on their own very thin pole-like objects should be **excluded**. These thin pole-like objects should also be **excluded**.

Any attachments on the side, e.g additional reflectors or arrows, mounts, or feet holding the guide should be **excluded** from the annotation marking. The following images are examples of things that should be excluded from the annotation: (a) a foot holding an attention marker, (b) a small foot on a reflector post, (c) an attachment on a reflector post.

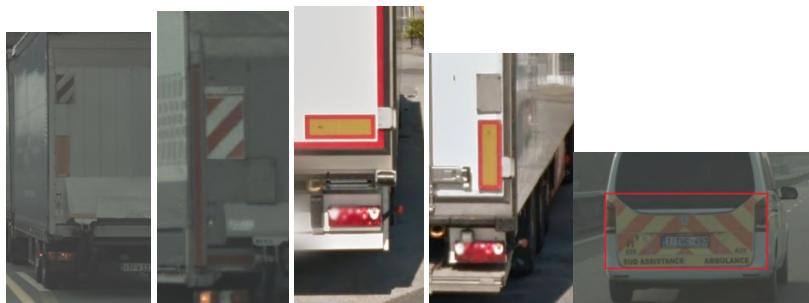


(a)

(b)

(c)

The sticker guides on vehicles, large or small, should be **excluded**. The following are examples of these type of guides.



### 7.2.3 TRAFFIC BEACON

A *traffic beacon* is a single light source with a continuous state: always on, always off, or flashing. These can be seen as restrictions, for example at pedestrian crossings, or as a warning for a certain traffic condition, such as a construction site. Note that only the light should be annotated, and the **pole should be excluded**. What is also to be **excluded** are lamps or lighting fixtures that are just meant to light up an area.

The 2D and 3D markings should include the light source and its corresponding container/box/fitting. For the last (rightmost) image this would include the larger black box around the actual yellow light fixture.

Examples are (non-exhaustive):



## 7.2.4 TRAFFIC SIGNAL

A *traffic signal* is a configuration of several lights where the state of the lights indicate a certain driving restriction or condition. The lights may be arranged vertically, horizontally, or in some mixed configuration. Only the head of the traffic signal is of interest, and **the poles should be excluded from the signal's annotation**.

**What to include:** Traffic signals for any traffic participant, i.e pedestrian lights, bicycle signals, or vehicle traffic lights at junctions or railway crossings.

**What to exclude:** Any light arrangement that does not control stop-and-go of any traffic participant.

Examples of what to include are (non-exhaustive):

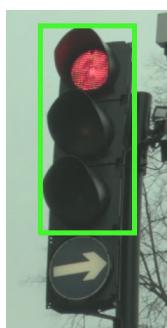


Example to exclude:



Speed cameras should be **excluded**, and also their mounting poles should be **excluded**.

**Edge cases:**



Some traffic signals have signs inside the traffic signal board. In these cases, the traffic signal should cover the light parts but the sign/signs should be excluded from the 2D and 3D markings.



Note that traffic signal markings should also exclude the signs attached to the side of the traffic signal.

## 7.2.5 DYNAMIC BARRIER

A *dynamic barrier* controls the flow of cars by completely blocking the path. They are usually pole-like objects that can be positioned either horizontally to block the passage of vehicles, or vertically to allow flow of traffic. There are also varieties that has more resemblance with gates rather than poles, and also blockers which appear from the ground. Only the part that moves and can block the flow of cars should be included in the markings.

Dynamic barriers are commonly seen, but not exclusively, at railway crossings, parking lots, and bridges.

Note that any traffic signs or beacons mounted on the “arm” should be **excluded** from the annotation, and annotated separately as traffic signs and beacons.

Examples are (non-exhaustive):



## 7.3 MARKINGS

The markings should specify the location of the landmark in the **image** and **point cloud**. A two-dimensional marking is required for all landmarks, while the three-dimensional should only be included if the landmark is present in the point cloud. There is no minimum pixel size limit on the objects that should be annotated. No matter how small, occluded, or truncated the landmark is - if it is clear to the human eye that there is an object, the two-dimensional markings and landmark properties must be set. Finally, the three-dimensional marking of the landmark should only be included when there are LIDAR points on the object.

### 7.3.1 2D PIXEL POSITIONS

The two-dimensional markings should be extracted using the **image**.

**Extreme points:** For each object, the top-most, bottom-most, left-most and right-most points belonging to the object should be marked. These four outer points: top, left, bottom, and right; should be defined in a **coordinates** array in that order. If there are several pixels that correspond to the outer-most part of an object, for example if there are several pixels along the same pixel-row that are all equally “top-most”, select the pixel at the center of that row for the top-most pixel.



**OK extreme  
points.**



**NOT OK extreme  
points.**

**Extrapolate:** Extrapolation should be done for landmarks that are not fully visible, either due to truncation or occlusion. Note that in both of these cases it can be beneficial to look at the 3D-marking, since those objects can sometimes be seen better seen in the point cloud. If the full object can not be seen in either the image or point cloud, the extrapolated sizes should be estimated to one's best guess.



A **truncated** landmark, i.e a landmark that extends beyond the image border.



An **occluded** landmark, where another object is blocking parts of the landmark.

A **maximum error of 2 pixels** in terms of the marking pixel positions in the horizontal and vertical directions is tolerated.

The 2D marking is defined as a GeoJSON MultiPoint as:

```
{
  "type": "MultiPoint",
  "coordinates": [
    [1142.68, 669.75], // The top-most extreme point, in order [x, y]
    [1143.42, 676.46], // The right-most extreme point, in order [x, y]
    [1131.28, 688.04], // The bottom-most extreme point, in order [x, y]
    [1129.16, 670.3] // The left-most extreme point, in order [x, y]
  ]
}
```

### 7.3.2 3D PARAMETERS

The three-dimensional markings should be extracted using the **image(s) and point cloud**. There is no minimum requirement of amount of points to estimate the center location, only that there is enough information to make an approximation. If there are several points, the center-most point, or a combination of the points, should be used. If there are few lidar points for the object, the image can be used to place the 3D markings in a good location. Bare in mind that the calibration is not always perfect. However, be careful to not overfit to the image too much when the point cloud already contains sufficient information to create the 3D markings.

**Ghost points:** For very reflective objects (which is common for landmarks), it is common for the point cloud to include “ghost points”, i.e points that do not correspond to an object in the real world. There are two ways to avoid including these in the 3D marking, and thus getting a too large 3D bounding box:



(1) Use reflection values on points  
to avoid the ghost points.  
(Yellow points are of high reflectance)



(2) Set a reasonable size using the image  
2D marking, and align in the middle  
of the point cloud cluster.

Sometimes a combination of the two can make the most sense.

**Parameters** - The 3D marking is defined by a location, size, and orientation:

- The **location** is described as:

```
{
  "type": "Point",
  "coordinates": [11.3, 5.2, 80.3] // The 3D coordinates in the order [x, y, z]
}
```

which is a location corresponding to an approximation of the center point of the landmark. These location parameters should be given in the same units, and in the same coordinate system, as in the provided data.

- The **size** is described by a length, width, and height using floating point numbers: `size_3d_length`, `size_3d_width`, and `size_3d_height`.
- The **orientation** is given in terms of quaternions, and should correspond to the orientation of the 3D box. Yaw, pitch, and roll can be derived from the quaternions. The parameters are in floating point numbers: `orientation_3d_qw`, `orientation_3d_qx`, `orientation_3d_qy`, and `orientation_3d_qz`.

The direction of a landmark is the direction of its contents (for signs, the direction is defined as the normal to the face of the sign). This is straightforward for traffic signs, traffic signals, etc. However some landmarks are bi-directional (reflector posts) in which case any of the two possible directions is OK. For the third case of no content any direction in yaw is OK (if it fits the shape of the landmark), but be sure to set the property `traffic_content_visible="Undefined"`.



If traffic signs are bent or curved in some way, this should be reflected in their 3D parameters.

### 7.3.3 EQUAL EXTRAPOLATION FOR 2D & 3D



Here is an example where the 2D box is hard to set, since the object is truncated, but the point cloud had enough points to set a good 3D box. In this case the extrapolated part of the 2D box can be set to the same as the 3D box. This also applies to occluded objects.

## 7.4 LANDMARKS PROPERTIES

The landmark properties should help distinguish the markings between different types of landmarks. The name of each property is given in **boldface** and the names of the possible values are given inside the {curly brackets}. No other names or properties or values are allowed.

### 7.4.1 SHARED

- **annotation\_uuid**

{“string, e.g uuid”}

Each landmark should be assigned with one `annotation_uuid` that is globally unique. If a landmark is truncated and present in more than one surround camera image, that object should have the same `annotation_uuid` in all the images it appears in. Moreover, if a landmark appears in several consecutive Scenes in a Scene Sequence, it should have the same `annotation_uuid` in all the Scenes it is present in.

- **class**

{“TrafficSign”, “TrafficGuide”, “TrafficBeacon”, “TrafficSignal”, “DynamicBarrier”, “Inconclusive”}

Every marked landmark should be assigned one of these landmark classes. Note that there can be properties that are specific for a certain landmark class.

An `Inconclusive` landmark is a landmark for which at least one of the following cases is true:

- It is in some way unclear to the human eye whether there is a landmark, e.g. due to the landmark being too small, or the image being too unsharp or blurred.
- It is unclear whether it is a single landmark or if there are more than one.
- Several highly overlapping landmarks are indistinguishable in the sense that it is practically impossible to individually mark the landmarks.

In either of these cases, the outer points of that (potential) landmark, or group of objects, should still be marked in the image. The purpose of marking these “Inconclusive”-landmarks is to make it possible to discard the corresponding specific pixel area in the image. **An inconclusive landmark has no additional properties.**

- **unclear**

{“false, true”}

An `unclear`-landmark is a landmark for which the class is evident, but the image provides too little information to confidently set the properties of the landmark. This can be due to e.g. the landmark being too small, occluded by some water or dirt on the camera lens or the image being too unsharp, blurred, dark or overexposed which cannot be fixed with image enhancement tools (like histogram equalization, brightness and contrast correction, etc.). Note that if it is not clear whether the landmark is of the chosen `class` then that should instead be set as `“Inconclusive”`, and this and the following properties should be omitted.

**If property `unclear=true`, all remaining properties should be left out by omitting the `ObjectProperties` key in the annotation file.**

The purpose of marking these `unclear`-landmarks is to make it possible to discard that corresponding specific pixel area in the image and that corresponding volume in the LIDAR point cloud. So, in cases when annotator is not sure whether the landmark is visible or not in the camera frame, it is preferred to annotate it with property `unclear` set to `true` over leaving it unannotated. If there are multiple landmarks with dubious visibility close to each other, it is preferable to include them as one object with the `unclear` property set to `true` instead of marking them individually.

- When an area is not annotated, it implicitly means that definitely no landmarks are present there in the image data. However, if there actually is some information in the image that is just not visible to the human eye, it would be impossible to find these edge cases.
- When an area is annotated with property `unclear` set to `true`, we are explicitly told that this area might require extra attention and that it is not clear whether there is a landmark or not.

Here is an example where motion-blur is causing landmarks to be “qualified” for being set as `unclear=true`:



Another example of landmarks (traffic signs) which should have `unclear=true` because the signs are far away and there are very few pixels to set the other properties (specially `traffic_content_visible`):



- odd**

`{false, true}`

Assign `true` if the landmark is particularly odd or rare, but for which the oddity is not reflected in terms of any of the other properties, e.g. a guide post that is painted in rainbow colors, or a very fancy traffic sign.

- is\_on\_dynamic\_object**

`{false, true}`

Whether the landmark is mounted or placed on another object, e.g on a trailer. This could be during transport of a landmark, in which case it can be considered inactive, or on a traffic pilot car where the landmark is relevant for the road users.

- occlusion\_ratio**

`{"None", "Light", "Medium", "Heavy", "VeryHeavy"}`

Estimate how much of the landmark that is occluded by something **in the active field of view of the camera in the image, excluding any parts of the ego vehicle body and camera housing**, according to the ranges:

- None: 0%
- Light: 1% - 20%

- Medium: 21% - 50%
- Heavy: 51% - 80%
- VeryHeavy: 81% - 100%

For a truncated landmark, the occlusion ratio should be estimated based on only the part of the landmark that is inside the active field of view of the camera. The truncated part of the landmark does not count as an occluded part. If the occlusion ratio is at the limit between two values, choose the higher value, e.g. "Medium" instead of "Light".

#### • **traffic\_content\_visible**

{"True", "False", "Undefined", "Inconclusive"}

Whether the content presented on a landmark is visible in the image. The "content" of a landmark refers to any symbols, text, reflector, light, etc. that is on the landmark. Set:

- "True" if such content of the landmark is **visible** from the camera. This also applies to traffic signals and electronic signs that are in an off-state, but where the content still would be visible from the camera. Note that this does not necessarily mean that it is possible to understand what the landmark means (like the meaning of a sign), but rather if it is evident that its content is facing and visible to the camera. For example, front-facing traffic signs should always be annotated as having their traffic content visible, even if it is not possible to discern the specific sign type.
- "False" if the content of the landmark is not visible from the camera.
- "Undefined" when there is no content on the landmark.
- "Inconclusive" if it is unclear whether the content is visible or if it is unclear whether there is any content presented on the landmark.

#### • **is\_for\_construction**

{false, true}

Whether the landmark is temporarily placed where it is because of some construction work or similar.

Some examples of this for cones, striped attention markers, and signs are:



Note that in the United States, all traffic signs in color orange are meant for construction. Some examples of signs that should have `is_for_construction:true` are as following:



## 7.4.2 CLASS SPECIFIC

### 7.4.2.1 Traffic sign

The Properties listed below should be assigned to all landmarks with `class=TrafficSign`.

- **`is_for_ego_road`**

`{"False", "True", "Inconclusive"}`

Whether the traffic sign is relevant/applicable for the ego road. See section “Definitions” for a definition for the ego road.

- **`is_for_ego_lane`**

`{"False", "True", "Inconclusive"}`

Whether the traffic sign is relevant/applicable for the ego lane. See section “Definitions” for a definition for the ego lane.

- **`is_for_other_traffic_participant`**

`{"False", "True", "Inconclusive"}`

Whether the sign is not relevant for a vehicle but concerns another traffic participant (i.e pedestrian, bicycle, train, boat, etc). An example of `is_for_other_traffic_participant="True"` would be a sign indicating that bicycles should be on the left, and pedestrian on the right side. Signs that are relevant for **both** vehicles and other traffic participants, such as pedestrian crossing signs, should have `is_for_other_traffic_participant="False"`. Note that for the vast majority of the signs in the frames, the signs are for vehicles, so the go to value for this property should be `False`.

- **`complementary_to_landmark`**

`{annotation_uuid of closest main sign or landmark if any*, "Inconclusive", "None"}`

If the sign is complementary to another “main sign” or landmark, add that ID here. Examples of complementary signs are: small “distance to” signs for exits, restriction sign with concerned traffic participants in a complementary sign under, and more. This should include all cases where a sign is not interpretable in itself, but needs the context of the sign and/or landmark right next to it. The `complementary_to_landmark` property should only be set if `traffic_content_visible="True"`.

**If the sign complements more than 1 “main” sign or landmark, the closest main sign or landmark should be reported.**

Below, some examples of traffic signs that are complementary to other signs, can be seen.



Two other examples, where the sign should have the ID of the landmark in the property list, in these cases a traffic signal.



- **is\_sticker**

{false, true}

Whether the sign is a sticker on a truck, caravan, or trailer (dynamic object, usually large). This should only be set to true in combination with `is_on_dynamic_object=true`.



### 7.4.2.2 Traffic Guide

The Properties listed below should be assigned to all landmarks with `class=TrafficGuide`.

- **type**

{"Reflector", "Attention", "SnowMarker", "Inconclusive"}

Assign a type to the traffic guide. Examples of these are provided in the definition of a traffic guide.

### 7.4.2.3 Traffic Beacon

There are no specific properties for traffic beacons.

## 7.4.2.4 Traffic Signal

The Properties listed below should be assigned to all landmarks with `class="TrafficSignal"`.

- **is\_for\_ego\_road**

`{"False", "True", "Inconclusive"}`

Whether the landmark is relevant/applicable for the ego road. See section “Definitions” for a definition for the ego road.

- **is\_for\_ego\_lane**

`{"False", "True", "Inconclusive"}`

Whether the landmark is relevant/applicable for the ego lane. See section “Definitions” for a definition for the ego lane.

- **is\_for\_other\_traffic\_participant**

`{"False", "True", "Inconclusive"}`

Whether the signal controls the stop-and-go of another traffic participant than a vehicle. Examples of

`is_for_other_traffic_participant="True"` would be for pedestrian, bicycle, or tram signals.

## 7.4.2.5 Dynamic Barrier

There are no specific properties for dynamic barriers.

## 7.4.3 SUMMARY

A summary over landmark properties.

Class	Property	Values
All	<code>scene_id</code>	CosmosID of the frame, of the form <code>&lt;car_name&gt;_&lt;camera_time&gt;_&lt;lidar_time&gt;_&lt;camera_name&gt;</code> , e.g <code>delta_1254672205930444_1570636987913882_FC</code>
All	<code>guideline_version</code>	The current version of the guidelines (each release changes the version of all projects)
All	<code>project</code>	<code>{"static_objects_3d", "fisheye_static_objects_3d"}</code>
All	<code>task_id</code>	Annotation company ID, can be UUID or similar
All	<code>annotation_uuid</code>	Globally unique ID for the landmark
All	<code>class</code>	<code>{"TrafficSign", "TrafficGuide", "TrafficBeacon", "TrafficSignal", "DynamicBarrier", "Inconclusive"}</code>
All except <code>"Inconclusive"</code>	<code>unclear</code>	<code>{false, true}</code>
All except <code>"Inconclusive"</code>	<code>odd</code>	<code>{false, true}</code>
All except <code>"Inconclusive"</code>	<code>is_on_dynamic_object</code>	<code>{false, true}</code>
All except <code>"Inconclusive"</code>	<code>occlusion_ratio</code>	<code>{"None", "Light", "Medium", "Heavy", "VeryHeavy"}</code>

Class	Property	Values
All except "Inconclusive"	traffic_content_visible	{"True", "False", "Undefined", "Inconclusive"}
All except "Inconclusive"	is_for_construction	{true, false}
"TrafficSign"	is_for_ego_road	{"False", "True", "Inconclusive"}
"TrafficSign"	is_for_ego_lane	{"False", "True", "Inconclusive"}
"TrafficSign"	is_for_other_traffic_participant	{"False", "True", "Inconclusive"}
"TrafficSign"	complementary_to_landmark	{"annotation_uuid of closest main sign or landmark if any*", "Inconclusive", "None"}
"TrafficSign"	is_sticker	{false, true}
"TrafficGuide"	type	{"Reflector", "Attention", "SnowMarker", "Inconclusive"}
"TrafficSignal"	is_for_ego_road	{"False", "True", "Inconclusive"}
"TrafficSignal"	is_for_ego_lane	{"False", "True", "Inconclusive"}
"TrafficSignal"	is_for_other_traffic_participant	{"False", "True", "Inconclusive"}

## 7.5 ILLUSTRATIVE EXAMPLES

Each illustrative example includes the 2D and 3D marking of a single landmark. Note that even if there are several landmarks to be annotated, only one is shown in each illustration.

### 7.5.1 TRAFFIC SIGN

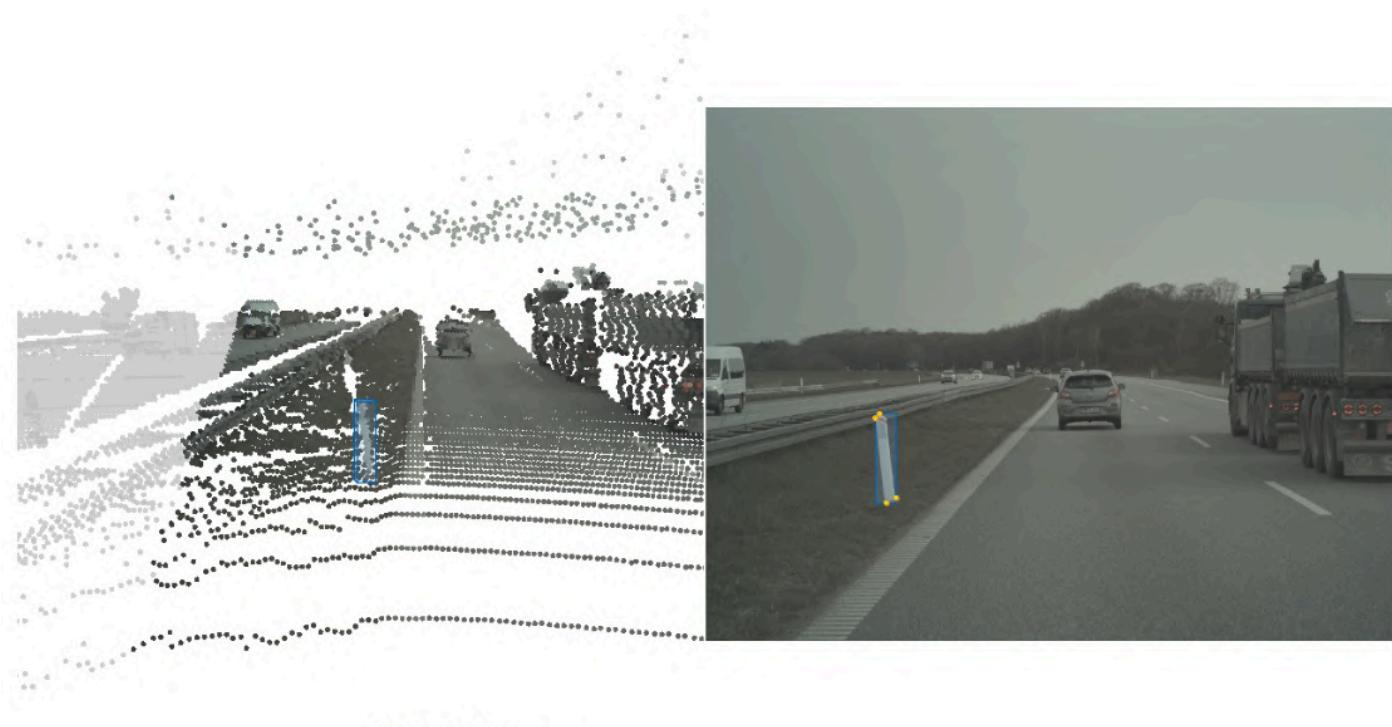
The below example includes the 2D and 3D markings of a `class="TrafficSign"`. This example would have properties: `traffic_content_visible="True"`, `is_for_ego_road="True"`, and `is_for_ego_lane="False"`.



Note that in a full annotation of this frame, there should be two traffic signs.

## 7.5.2 TRAFFIC GUIDE

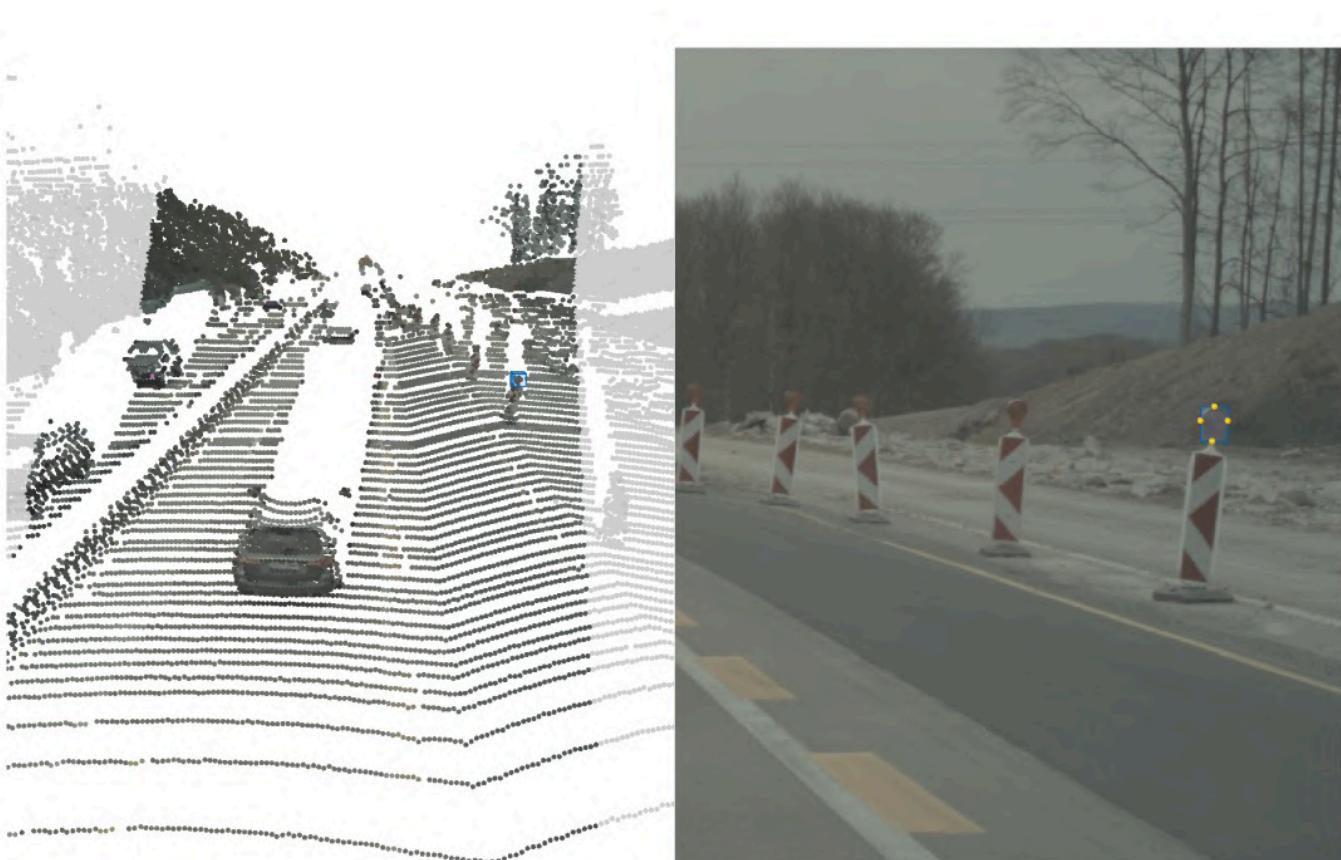
The below example includes the 2D and 3D markings of a `class="TrafficGuide", type="Reflector"`. This example would have properties: `type="Reflector"`, and `traffic_content_visible="True"`.



Note that in a full annotation of this frame, all reflector posts along the way should be annotated.

### 7.5.3 TRAFFIC BEACON

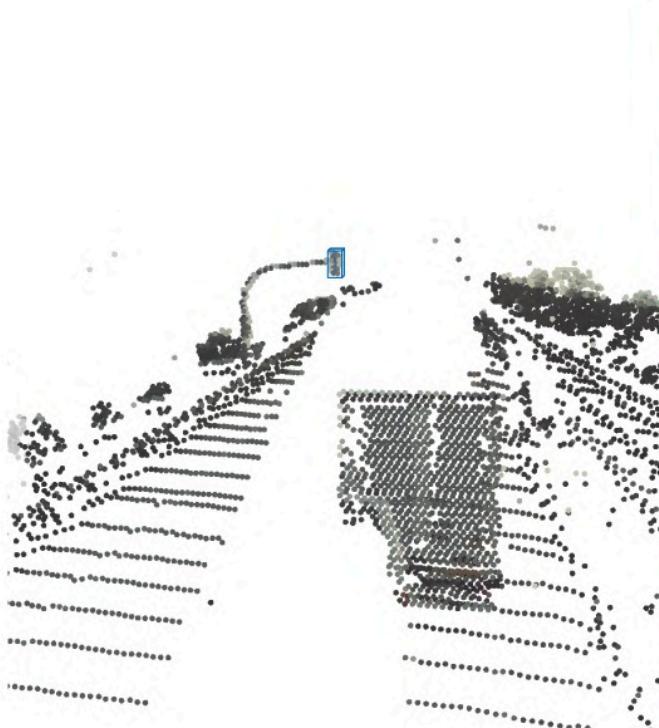
The below example includes the 2D and 3D markings of a `class="TrafficBeacon"`. This example would have property `traffic_content_visible="True"`.



Note that in a full annotation of this frame, there should be five beacons and five attention traffic guides. Also note that the feet of the traffic guides should be excluded from the marking.

## 7.5.4 TRAFFIC SIGNAL

The below example includes the 2D and 3D markings of a `class="TrafficSignal"`. This example would have property `traffic_content_visible="False"`.



Note that in a full annotation of this frame, there should be two traffic signals.

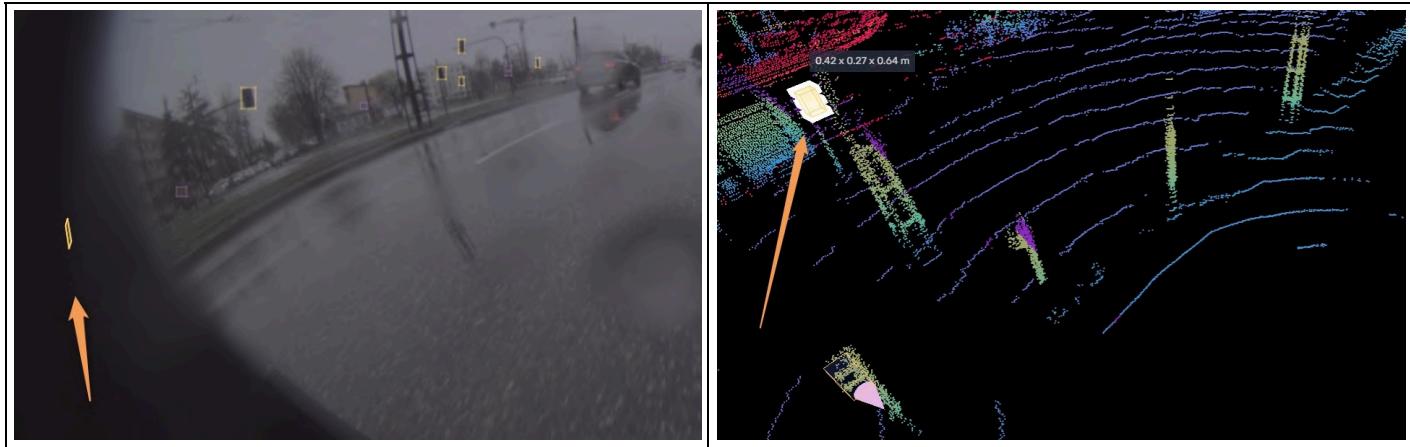
## 7.5.5 DYNAMIC BARRIER

Dynamic barriers are quite uncommon. However, they should be similarly annotated as the other examples, where the markings tightly encapsulate the landmark in both 2D and 3D.

## 7.5.6 LANDMARKS BEHIND THE CAMERA HOUSING OR EGO VEHICLE BODY

Landmarks fully occluded by the camera housing or the body of the ego vehicle should not be annotated at all.

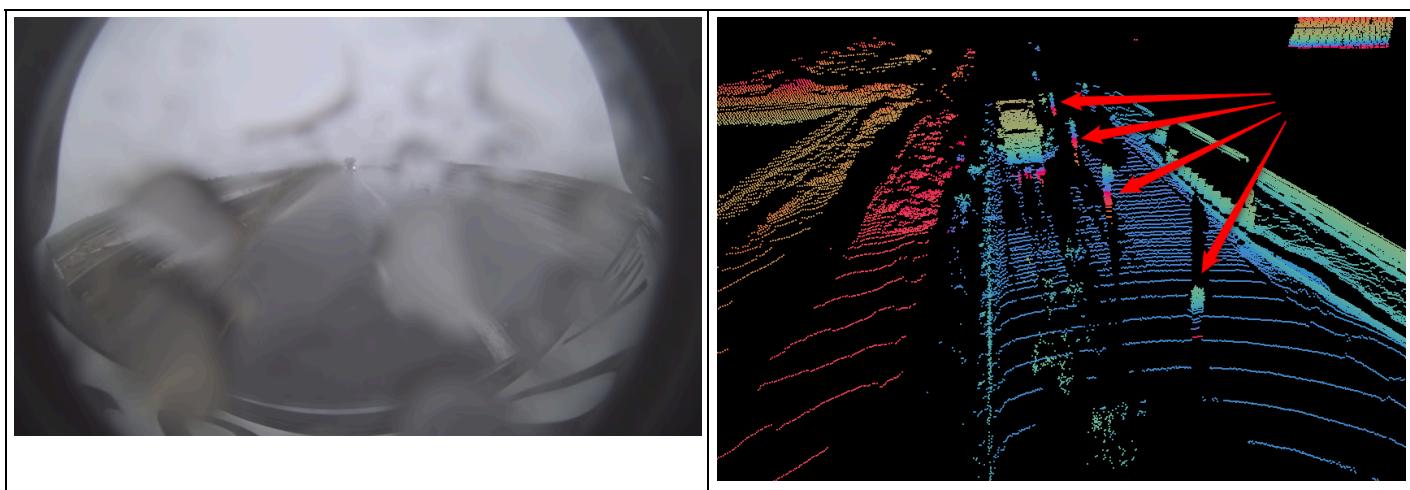
On the example below the traffic lights on the left marked with an arrow should not be annotated as they are fully covered by the housing of the fisheye camera, even though they are clearly visible in the lidar point cloud.



## 7.5.7 LANDMARKS FULLY OCCLUDED BY WATER OR DIRT ON CAMERA LENS

Traffic Guides in the image below which are fully occluded by droplets of water, without any pixels hinting their existence, should not be annotated, even though they are clearly visible in the lidar point cloud.

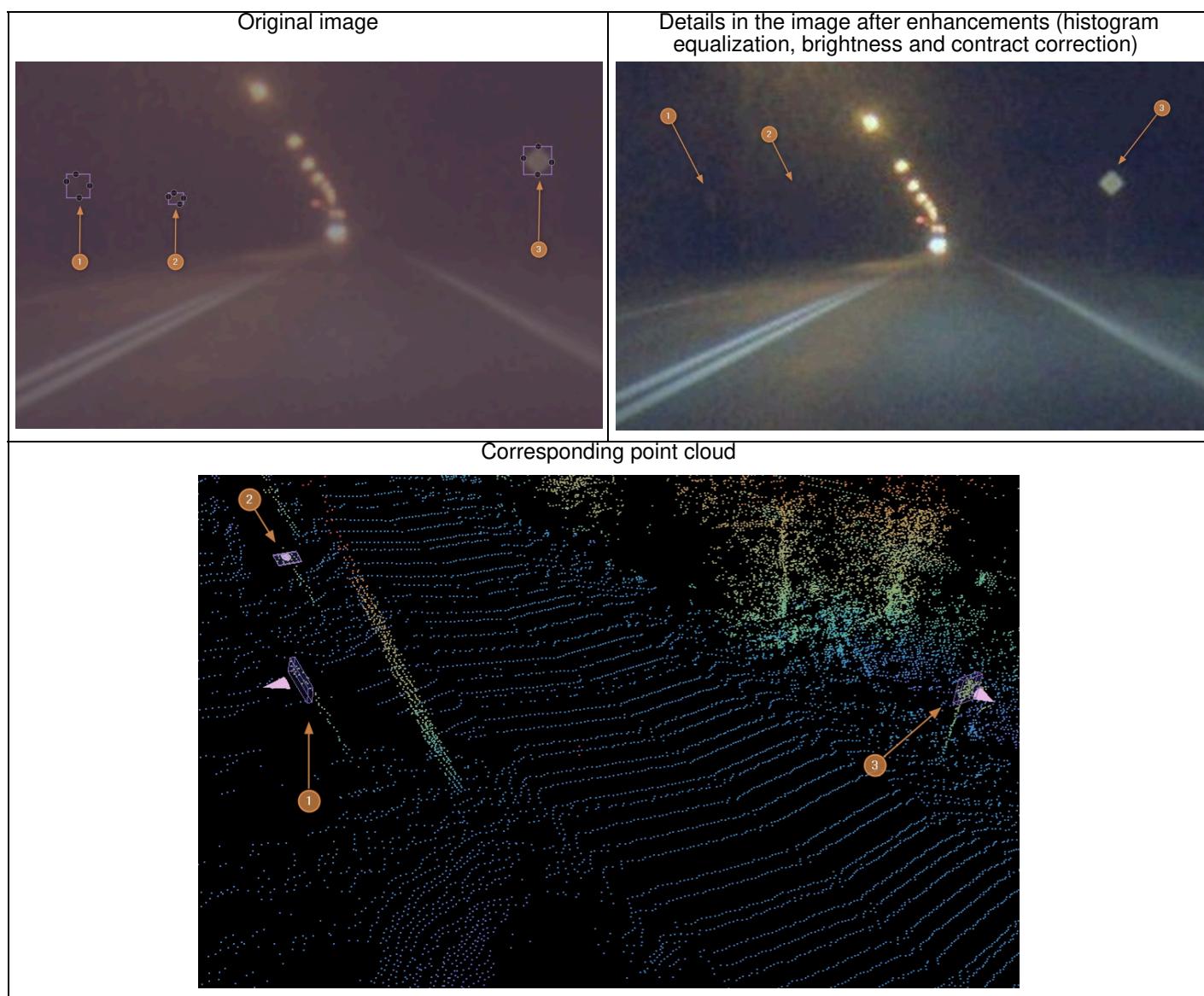
In cases when it is not clear if there is some information about the landmark in the image, it is always safe to annotate it with property `unclear=true`. In this case, however, it is clear that nothing is visible there and no information could be retrieved using image enhancement tools.



## 7.5.8 LANDMARKS HIDDEN IN THE DARKNESS

In the image below traffic signs marked with 1 and 2 are hidden in the darkness, without any pixels hinting their existence. After using image enhancement tools (like histogram equalization, brightness and contrast correction) the sign marked with 1 becomes visible so should definitely be annotated. The one marked with 2 is still not visible so it is fine to leave it not annotated, even though it is clearly visible in the corresponding lidar point cloud. Note, that only the pixels for the sign itself should be considered - the pole holding the sign is not considered as a hint for the existence of the sign.

In cases when it is not clear if there is some information about the landmark in the image, it is always safe to annotate it with property `unclear=true`. So, in this case annotating the sign marked with 2 this way would also be correct. If there were multiple not clearly visible landmarks close to each other (e.g. traffic signs mounted on the same pole or visually located close to each other in the image) they could also be annotated as a single `unclear=true` object, if they are not separated by any other, clearly visible object.



## 8 Traffic Signs

*Vision annotations of traffic signs*

## 8.1 INTRODUCTION

This document contains instructions for annotating traffic signs and their content in camera images.

**Traffic Signs** are loosely defined as plates or similar structures, indicating traffic rules to road users and showing useful information or warnings. Traffic signs are often, but not always, attached to some form of signpost.

This guideline document is a continuation of the Landmark Guidelines. There are currently two ways to annotate traffic signs:

In most cases, the annotation projects based on this document should process all landmark annotations with `class="TrafficSign", traffic_content_visible="True"` OR `class="TrafficSign", unclear=true` and add additional traffic sign properties according to this module. Note that the ID of the traffic sign in the landmark project should equal the ID in this project.

Alternatively, projects based on this document could also process camera images directly. The first step is then to annotate all front facing traffic sign landmarks according to the landmarks module, with the 2D bounding box geometry and landmark properties, including but not limited to:

- `is_on_dynamic_object`
- `occlusion_ratio`
- `traffic_content_visible`
- `is_for_construction`
- `is_for_ego_road`
- `is_for_ego_lane`
- `is_for_other_traffic_participant`
- `complementary_to_landmark`
- `is_sticker`

Finally, any additional traffic sign properties should be annotated according to this module. Note that for properties that are present in both the landmarks module and the traffic sign module, like `class`, `unclear` and `odd`, the traffic sign properties takes precedence. Also worth noting is that the traffic sign property `imported_from_landmark` should always be set to `false` when processing camera images directly like described above.

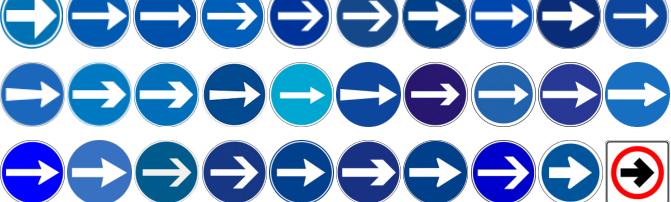
## 8.2 CLASSES

All the classes with their category, type and associated examples are represented in the next table. Each category and type includes example images in two general standards for traffic signs: the Vienna convention standard and MUTCD standard. Many countries based their traffic sign on one of these standards with some added variations. US and Canada are following MUTCD, the European countries and China are based on the Vienna convention, and Ireland and Japan use a mix of the two standards.

In addition to the signs represented in the table, a Sign Class `NotListed` is introduced that should be the default for signs that do not belong to any of the classes listed in the table. The purpose of this class is that when this document is updated with an extended list of classes, back-filling old annotations only require revisiting the traffic signs with `class="NotListed"`. The class assignment of these annotations can then be updated if their true class was added to the list.

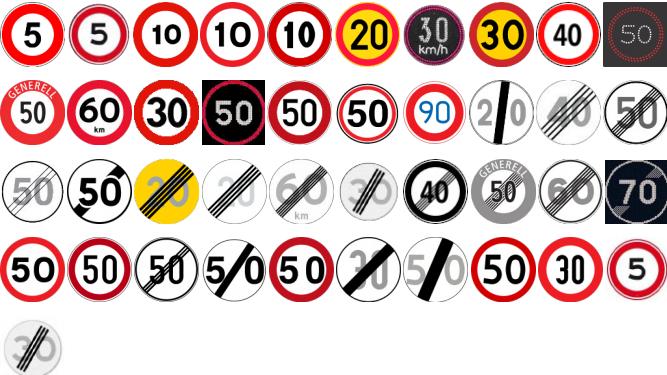
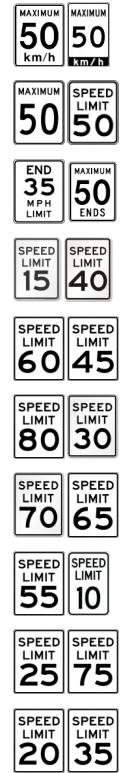
Sign class	Type	Vienna Convention	MUTCD
Checkpoint	TollStation	 	
Indication	CameraSurveillance		
Indication	Tunnel		
Mandatory	MinimumSpeedLimit		

Sign class	Type	Vienna Convention	MUTCD
Mandatory	PassOnEitherSide		
Mandatory	PassOnThisSideLeft		
Mandatory	PassOnThisSideRight		
Mandatory	ProceedStraight		
Mandatory	ProceedStraightOrTurnLeft		
Mandatory	ProceedStraightOrTurnRight		

Sign class	Type	Vienna Convention	MUTCD
Mandatory	Roundabout		
Mandatory	TurnAhead		
Mandatory	TurnLeft		
Mandatory	TurnLeftAhead		
Mandatory	TurnRight		
Mandatory	TurnRightAhead		

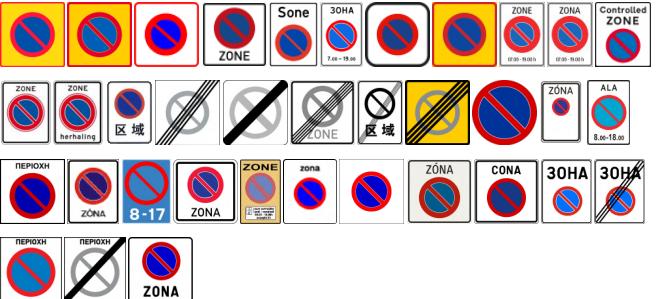
Sign class	Type	Vienna Convention	MUTCD
Priority	GiveWay		
Priority	GiveWayOncoming		
Priority	PrioOverOncoming		
Priority	PriorityRoad		

Sign class	Type	Vienna Convention	MUTCD
Priority	Stop		
Prohibitory	EndOfAll		
Prohibitory	HeightLimit		

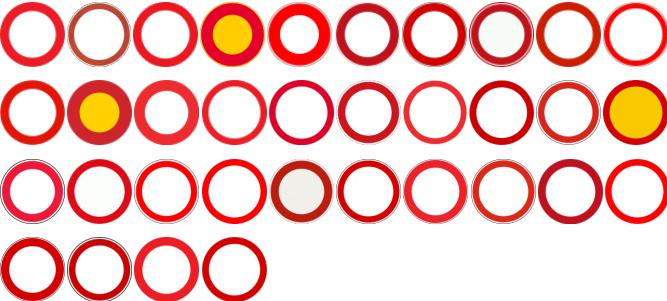
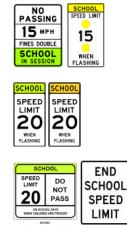
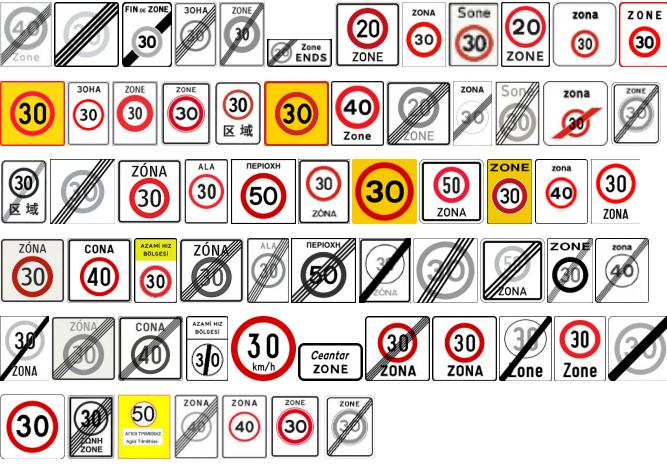
Sign class	Type	Vienna Convention	MUTCD
Prohibitory	MaximumSpeedLimit		
Prohibitory	MaximumSpeedLimitConditional		

Sign class	Type	Vienna Convention	MUTCD
Prohibitory	MaximumSpeedLimitNight		
Prohibitory	NoEntry		

Sign class	Type	Vienna Convention	MUTCD
Prohibitory	NoMotorVehicles		
Prohibitory	NoMotorVehiclesExceptMotorcycles		
Prohibitory	NoOvertaking		   

Sign class	Type	Vienna Convention	MUTCD
Prohibitory	NoOvertakingTruck		
Prohibitory	NoParking		
Prohibitory	NoParkingZone		
Prohibitory	NoPedestrianCrossing		

Sign class	Type	Vienna Convention	MUTCD
Prohibitory	NoStopping		
Prohibitory	NoStoppingZone		
Prohibitory	NoTurn		
Prohibitory	NoUTurn		

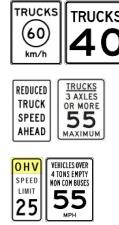
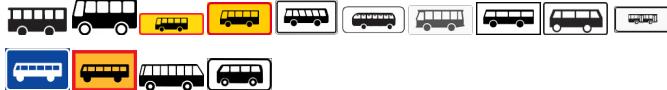
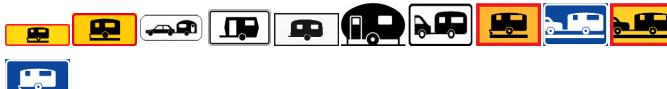
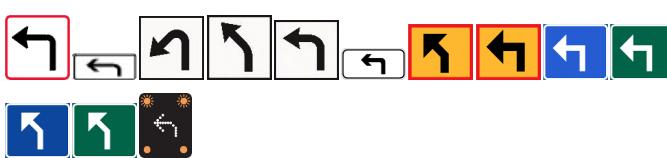
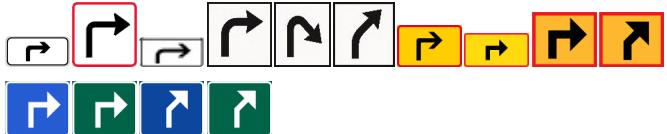
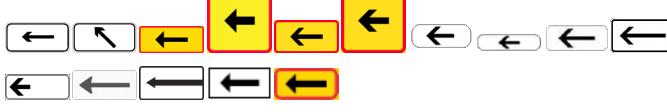
Sign class	Type	Vienna Convention	MUTCD
Prohibitory	RoadClosed		
Prohibitory	SpeedLimitDual		
Prohibitory	SpeedLimitSchool		
Prohibitory	SpeedLimitZone		
Prohibitory	WeightLimit		
Prohibitory	WidthLimit		

Sign class	Type	Vienna Convention	MUTCD
RoadType	BuiltupAreaIllustration		
RoadType	BuiltupAreaText		
RoadType	CyclingStreet		
RoadType	LimitedAccessRoad		
RoadType	LivingStreet		

Sign class	Type	Vienna Convention	MUTCD
RoadType	MotorwayBlue		
RoadType	MotorwayGreen		
RoadType	PedestrianStreet		
Special	AdvisorySpeedLimit		
Special	AdvisorySpeedLimitAhead		
Special	AdvisorySpeedLimitCurve		

Sign class	Type	Vienna Convention	MUTCD
Special	AdvisorySpeedLimitRamp		
Special	BicycleCrossing		
Special	BicyclePathWay		
Special	BikeOrPedestrianCrossing		
Special	BikeOrPedestrianPathWay		

Sign class	Type	Vienna Convention	MUTCD
Special	OneWay		
Special	Parking		
Special	PedestrianCrossing		
Special	PedestrianPathWay		

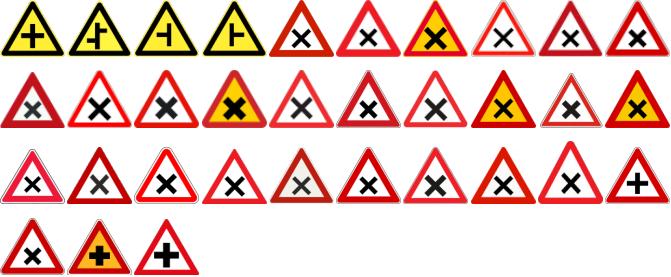
Sign class	Type	Vienna Convention	MUTCD
Special	SpeedLimitOtherUS		
Supplementary	Bus		
Supplementary	Caravan		
Supplementary	DirectionArrowLeft		
Supplementary	DirectionArrowRight		
Supplementary	EndOfRestriction		
Supplementary	Exit		
Supplementary	ExitLeft		
Supplementary	ExitRight		
Supplementary	ExtentArrowBidirectional		
Supplementary	ExtentArrowLeft		

Sign class	Type	Vienna Convention	MUTCD
Supplementary	ExtentArrowRight		
Supplementary	Fog		
Supplementary	ForDistance		
Supplementary	InDistance		
Supplementary	MotorHome		
Supplementary	Motorcycle		
Supplementary	PassengerCar		
Supplementary	Rain		
Supplementary	Reminder		
Supplementary	RestrictionZone		
Supplementary	SchoolZone		
Supplementary	SnowOrIce		
Supplementary	StartOfRestriction		

Sign class	Type	Vienna Convention	MUTCD
Supplementary	TimeLimit		
Supplementary	Tractor		
Supplementary	Trailer		
Supplementary	Truck		
Supplementary	VehicleTypeCombined		
Supplementary	WeightLimit		
Supplementary	Wet		
Warning	AnimalDomestic		

Sign class	Type	Vienna Convention	MUTCD
Warning	AnimalWild		

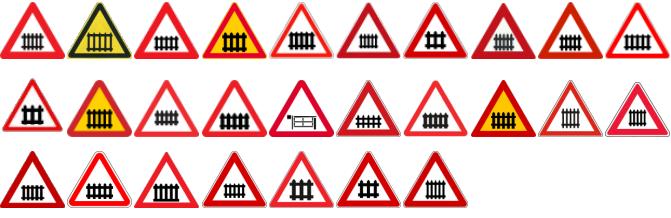
Sign class	Type	Vienna Convention	MUTCD
Warning	Children		
Warning	CongestionAhead		
Warning	Crossing		

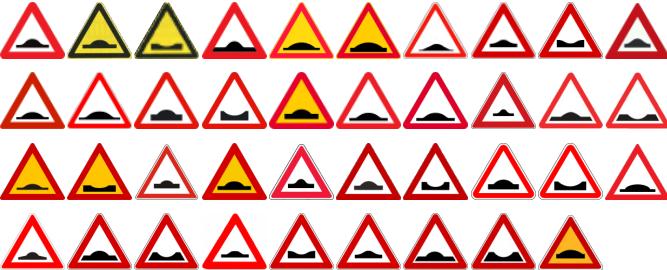
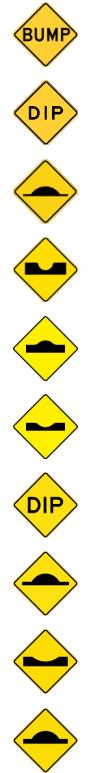
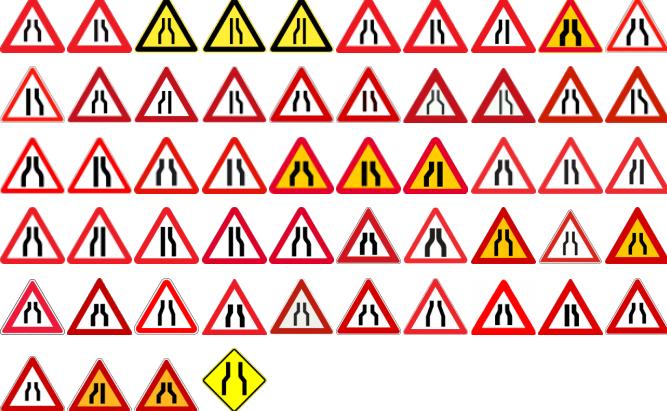
Sign class	Type	Vienna Convention	MUTCD
Warning	CrossroadsWithAMinorRoad		
Warning	CrossroadsWithPriorityToTheRight		

Sign class	Type	Vienna Convention	MUTCD
Warning	Curve		
Warning	Cyclists		

Sign class	Type	Vienna Convention	MUTCD
Warning	DoubleCurve		
Warning	DrawBridge		

Sign class	Type	Vienna Convention	MUTCD
Warning	FallingRock		
Warning	Fog		
Warning	GenericWarning		
Warning	HorseRiding		

Sign class	Type	Vienna Convention	MUTCD
Warning	LevelCrossing		
Warning	LevelCrossingNoBarriers		
Warning	MergingTraffic		
Warning	OverheadBridge		

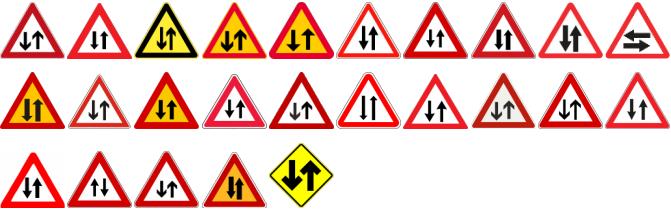
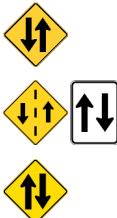
Sign class	Type	Vienna Convention	MUTCD
Warning	RoadBump		
Warning	RoadNarrows		

Sign class	Type	Vienna Convention	MUTCD
------------	------	-------------------	-------

Warning	RoadWork		                   

Sign class	Type	Vienna Convention	MUTCD
Warning	RoughRoad		
Warning	Roundabout		
Warning	Slippery		

Sign class	Type	Vienna Convention	MUTCD
Warning	SnowOrIce		 
Warning	SteepHill		      
Warning	TrafficSignalAhead		    
Warning	TunnelAhead		  

Sign class	Type	Vienna Convention	MUTCD
Warning	TwoWayTraffic		

## 8.2.1 IMPORTANT EXAMPLES AND NOTES:

- **Minor variations** of the signs in the table above is OK.



These signs are Prohibitory\_NoParking



These are Prohibitory\_NoStopping

- The blue rectangular sign with a white arrow should have class=Special\_OneWay.



class=Special\_OneWay

- Text-based signs can vary a lot in appearance between countries, so for class=RoadType\_BuiltupAreaText, try to be as careful as possible. Note that navigational/directional and road-number signs should be set as class=NotListed\_None. Some examples of what is NOT class=RoadType\_BuiltupArea, and should be set as class=NotListed\_None, are:



- There are several types of speed limit signs that should be carefully assigned to the correct class. Here is a short table of all available types, examples were included in the table above.

Class	Available speed limit types
Mandatory	MinimumSpeedLimit
Prohibitory	MaximumSpeedLimit, MaximumSpeedLimitConditional, MaximumSpeedLimitNight, SpeedLimitZone, SpeedLimitSchool, SpeedLimitDual

Class	Available speed limit types
Special	<code>AdvisorySpeedLimit</code> , <code>AdvisorySpeedLimitAhead</code> , <code>AdvisorySpeedLimitCurve</code> , <code>AdvisorySpeedLimitRamp</code> , <code>SpeedLimitOtherUS</code>



Note: Class of `Prohibitory_SpeedLimitDual` shows two speed values and the `speed_value` should be the maximum speed for cars(not trucks). In this case the `speed_value` property should be set to the Maximum speed limit (first number). For example this image should be set as `class=Prohibitory_SpeedLimitDual` and `speed_value=70`.



Note: Some signs with the `contains_inner_signs` property set to `true` contain inner speed limit signs. For these signs, the `speed_value` should also be set to the maximum speed limit for cars, similarly to the `Prohibitory_SpeedLimitDual` above. For example, this image should be set as `class=NotListed`, `contains_inner_signs=true` and `speed_value=100`.



Note: The electrical signs that indicate the speed of the ego vehicle should **not** be annotated as an electrical speed sign, instead as class `"NotListed"`.

- All speed limit signs related to **other cars** should be set to `class=Special_SpeedLimitOtherUS`.
- The speed limit signs on VMS boards are `Prohibitory` if they have a red circle around the speed value, and they are `Advisory` without such a circle.



Example of `Prohibitory_MaximumSpeedLimit`



Example of `Special_AdvisorySpeedLimit`

- There are several road number signs in the road that should not be mistaken with speed limit signs. These signs are smaller in size and more rectangular with arbitrary numbers. Here are a few examples of such road signs:



- The following electronic speed limit should be considered as `Prohibitory_MaximumSpeedLimit` with the correct speed value and Electric hardware type:



- Here is a table of sign classes which are often misannotated:

Class	Misannotated as

Class	Misannotated as
 Special_AdvisorySpeedLimit	 Prohibitory_MaximumSpeedLimit
 Prohibitory_NoParkingZone, is_end_sign=true	 Prohibitory_EndOfAll
 Prohibitory_NoStoppingZone	 Prohibitory_NoParkingZone
 Special_OneWay	 Mandatory_ProceedStraight
 NotListed	 Supplementary_Bus
 NotListed	 Supplementary_Bus

## 8.3 PROPERTIES

The following section describes all properties that have to be assigned to each traffic sign.

Note that the `annotation_uuid` property should be set for all traffic signs, and be the same as the corresponding landmark `annotation_uuid` for that sign.

- **annotation\_uuid**

`{*string, e.g uuid*}`

A unique ID that remains the same for the same object over multiple frames. This should be identical to the `annotation_uuid` of the traffic sign in the landmark project as well.

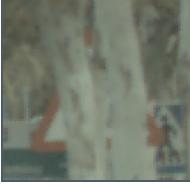
- **unclear**

`{false, true}`

An unclear traffic sign is one where it is impossible to distinguish it between the different classes and types. Also, an electronic sign that is in an off-state should be considered as unclear.

**If the traffic sign is unclear, this should be set to `true` and all other properties except for `annotation_uuid` should not be annotated.**

Below, a few examples of unclear signs with an explanation is shown.

Example	Reason
	Far away sign, very few pixels to see or read the sign.
	Highly occluded sign that is unreadable, this could be any warning sign.
	Too dark sign that is faded, this sign could be either <code>Prohibitory_NoStopping</code> or <code>Prohibitory_NoParking</code> .

- **class**

`{*string*}`

A combination of the sign class and type, separated by underscore (e.g `class="Supplementary_Bus"`). The available values

are described in the table below.

Class	Available types
Checkpoint	{TollStation}
Indication	{Tunnel, CameraSurveillance}
Mandatory	{MinimumSpeedLimit, PassOnEitherSide, PassOnThisSideLeft, PassOnThisSideRight, ProceedStraight, ProceedStraightOrTurnLeft, ProceedStraightOrTurnRight, Roundabout, TurnAhead, TurnLeft, TurnLeftAhead, TurnRight, TurnRightAhead}
Priority	{GiveWay, GiveWayOncoming, PrioOverOncoming, PriorityRoad, Stop}
Prohibitory	{EndOfAll, HeightLimit, MaximumSpeedLimit, MaximumSpeedLimitConditional, MaximumSpeedLimitNight, NoEntry, NoMotorVehicles, NoMotorVehiclesExceptMotorcycles, NoOvertaking, NoOvertakingTruck, NoParking, NoParkingZone, NoStopping, NoStoppingZone, NoTurn, NoUTurn, RoadClosed, SpeedLimitZone, SpeedLimitSchool, SpeedLimitDual, NoPedestrianCrossing, WeightLimit, WidthLimit}
RoadType	{BuiltUpAreaIllustration, BuiltUpAreaText, CyclingStreet, LimitedAccessRoad, LivingStreet, Motorway, PedestrianStreet}

Class	Available types
Special	{AdvisorySpeedLimit, AdvisorySpeedLimitAhead, AdvisorySpeedLimitCurve, AdvisorySpeedLimitRamp, BicycleCrossing, BicyclePathWay, BikeOrPedestrianCrossing, BikeOrPedestrianPathWay, OneWay, PedestrianCrossing, SPedestrianPathWay, speedLimitOtherUS, Parking}
Supplementary	{Bus, Caravan, DirectionArrowLeft, DirectionArrowRight, EndOfRestriction, Exit, ExitLeft, ExitRight, ExtentArrowBidirectional, ExtentArrowLeft, ExtentArrowRight, Fog, ForDistance, InDistance, Motorcycle, Rain, Reminder, RestrictionZone, SchoolZone, SnowOrIce, StartOfRestriction, TimeLimit, Tractor, Trailer, Truck, VehicleTypeCombined, WeightLimit, Wet}
Warning	{AnimalDomestic, AnimalWild, Children, CongestionAhead, Crossing, CrossroadsWithAMinorRoad, CrossroadsWithPriorityToTheRight, Curve, Cyclists, DoubleCurve, DrawBridge, FallingRock, Fog, GenericWarning, HorseRiding, LevelCrossing, LevelCrossingNoBarriers, MergingTraffic, RoadBump, RoadNarrows, RoadWork, RoughRoad, Roundabout, Slippery, SnowOrIce, TrafficSignalAhead, TwoWayTraffic, TunnelAhead, SteepHill, OverheadBridge}
NotListed	{None}

The class `NotListed` should be used when the sign is not included in the list of signs to annotate. In these cases the type is None, thus `class="NotListed"`.

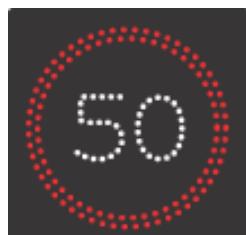
- **hardware\_type**

{"Normal", "Prism", "Electronic", "Combination"}

Which type of sign it is, material-wise. The types are:

- Normal - what we mostly consider a normal sign type, usually made of a thin piece of metal
- Prism - changeable sign, usually made of small sheets of metal that can rotate to change the content of the sign
- Electronic - usually made out of LEDs to make the content of the sign changeable
- Combination - if the sign consist of multiple different hardware types, for example a normal sign with a small electronic part within it

Below, a few examples of normal, prism, electronic and combined signs can be found.

normal	electronic	prism	combined
     		   	  

Sometimes it can be hard to distinguish a normal sign from a prism. It is important to know that prism signs are in comparison to normal signs very uncommon, so better to assume that a sign is normal if uncertain.

- **is\_end\_sign**

{false, true}

Whether this sign signals the end of a previous sign. This is usually indicated by having a diagonal line through the sign.

<code>is_end_sign=false</code>	<code>is_end_sign=true</code>
	



Note: in certain countries there can just be a supplementary sign to indicate the end of the restriction. In this case the top-most "main" sign would have `is_end_sign=false` and the supplementary sign would have `is_end_sign=true, complementary_to_landmark=<ID of closest sign>`. The complementary property is set in the landmark project for the sign.

- **speed\_value**

{\*integer\*, null}

The value of the speed on the sign, if any. For maximum, recommended, or minimum speed signs this could be a factor of 5 as 5, 10, 20, 25, 30,... etc. In case of other values, do **not** set the speed value and instead set the odd property to true. This also applies for restricted speeds on zone signs.

- **is\_marked\_as\_deactivated**

{"True", "False", "Partially", "Inconclusive"}

Whether the sign was marked as deactivated. This sometimes happens due to construction work, but is**not** common. The marking usually is done by placing some kind of colored cross on the traffic sign that indicated to vehicles that they should temporarily ignore the sign's content. Note that this is not the same as `is_end_sign`, but can sometimes be visually similar.

- True: The traffic sign is marked as deactivated.
- False: The traffic sign is not marked as deactivated (the most common case).
- Partially: Only some parts of the traffic sign are marked as deactivated.
- Inconclusive: For some reason the annotator is unable to give a clear judgement, e.g. due to occlusion.

- **broken**

{"No", "Slightly", "Significantly"}

Whether the traffic sign is broken in some way. This covers damage (e.g. holes in traffic sign object or parts of it being broken off) and visible tilt (with the most extreme case of tilt being that the traffic sign object is lying flat on the ground).

- No: The traffic sign object shows no visible signs of being broken. This is also to be assigned if damage on the traffic sign is not visible due to occlusion.
- Slightly: The traffic sign object is slightly damaged, e.g. a small hole or it has a visible tilt.
- Significantly: The traffic sign object is significantly damaged, e.g. a part of it being broken off or by being strongly angled towards the ground (up to lying flat).

- **odd**

{false, true}

Whether the traffic sign is in any way odd, or deviating from what would be considered normal for that particular sign. This could include some odd marking, unusual speed limit values or any variation that are meant as funny. Examples of `odd=true` are:



- **contains\_inner\_signs**

`{false, true}`

If the sign in any way contains a visualization of another sign that can also be seen on its own, then the sign should have `contains_inner_signs: true`. This should exclude the signs that already have a class type other than `NotListed`, like `Prohibitory_NoParkingZone` or `Prohibitory_SpeedLimitZone` where this property should be `false`.

Examples:

<code>contains_inner_signs=true</code>	<code>contains_inner_signs=false</code>