Traffic Lights Detection

This document describes the data format and relevant evaluation criteria of the traffic lights detection task in Baidu Apollo program.

1. Introduction of Datasets

It provides 20,000 frames of image data, which can be divided into 10,000 frames of training sets and 10,000 frames of testing sets. 200 frames of the testing sets are extracted as the sample images. The data are mainly collected from 8:00 to 17:00, concentrated in some road sections in Beijing, which includes the data of sunny days, cloudy days and foggy days. The data are 1080P colorized images. The traffic lights are vertical (including three rounds or arrows). The width of lamps marked is greater than 10 pixels. The lights can be divided into green lights and non-green lights according to the colors. Non-green lights include red lights, yellow lights and black lights (the color is not determined. It may be very dark or not glow.) In these 20,000 frames of images, there are 27,787 green lights and 43,852 non-green lights (including 36,880 red lights, 1,785 yellow lights and 5,187 black lights).

2. Acquisition Equipment

Two cameras for the data collection of traffic lights are mounted on the car roof, one of which uses a lens with a focal length of 25mm and the other uses a lens with a focal length of 6mm (the open data are collected by a telephoto camera).

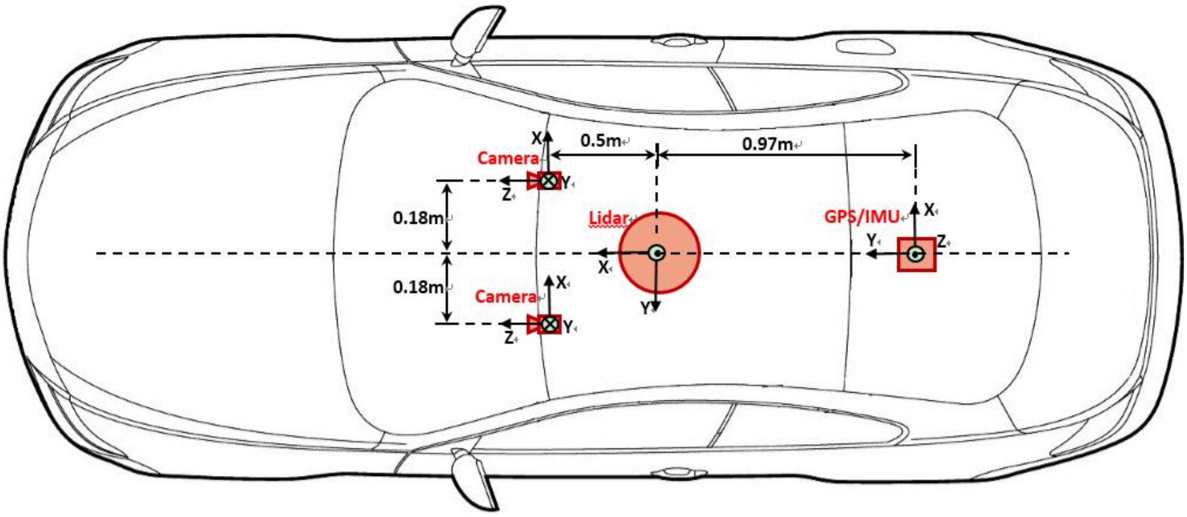

Figure 1 Vertical View of Data Acquisition Equipment

Parameters of the telephoto and short-focus cameras are as shown in Table 1:

Table 1 Parameters of the telephoto and short-focus cameras

Camera model	**LI-USB30-AR023ZWDR**
Resolution @ frame rate	1920 x 1080 @ 30 fps
Number of A / D bit depth	12 bits
Sensor module	ON Semiconductor AR023Z 1080p HD
Sensor exposure mode	Line exposure
Sensor size	1/2.7”
Pixel size	3.0 um
Color output	colorized
Focal length of lens	6 mm, 25 mm
Data interface	USB 3.0

3. Format Description

3.1 Format of Training Sets

The training set is organized according to the following directory structure:

```
train_data/                          // it contains the top folders of the training data
 ├──── images                        // it contains image files in the training data, which are
numbered from 0 to 9999
 |    ├──── 00000.jpg
 |    ├────   :
 |    └──── 09999.jpg
 ├── ─  labels                        // it contains the annotation files corresponding to the
                                       images,which are numbered from 0 to 9999
 |    ├──── 00000.txt
 |    ├────   :
 |    └──── 09999.txt
 └──── list                          // list files of images and annotations
```

Formats of annotation files in the labels

Each annotation file corresponds to a frame of image, where each line corresponds to a traffic light box. The column is defined as shown in Table 2, where the coordinate origin is the upper-left vertex of the image. The columns are separated by a space character.

Table 2 Formats of annotation files

Column No.	**Definition**
1	Types of traffic lights. Use 1/2 of values to correspond to the non-green / the green lights.
2	Coordinate positions on the left of the light box
3	Coordinate positions on the top of the light box
4	Coordinate positions on the right of the light box
5	Coordinate positions on the bottom of the light box

Formats of list files

Each line is the relative path of a pair of images and the annotation files. There is a space character between two columns. For example:

```
images/00000.jpg labels/00000.txt
images/00001.jpg labels/00001.txt
     :
     :
images/09999.jpg labels/09999.txt
```

3.2 Format of Testing Sets

The format of the testing sets is the same as that of the training sets, but there are no labels folders. The organizational form is as follows:

```
test_data/                      // it contains the top folders of the testing data
├── images                      // it contains image files in the testing data, which are numbered
│                                  from 0 to 9999
│  ├── 00000.jpg
│  ├──    :
│  └── 09999.jpg
└── list                        // list files of images
```

Each line of the list file is the relative path of a frame of image. For example:

```
images/00000.jpg
     :
images/09999.jpg
```

3.3 Format of Prediction Results

Users need to save the test result according to the agreed output format. The platform will perform evaluation according to the period specified by users.

All results are stored in the result files in the order of the images in the list file of the testing set. In the result files, each line corresponds to a traffic light box. The column is defined as shown in Table 3, where the coordinate origin is the upper-left vertex of the image. The columns are separated by a space character.

Table 3 Format of the test results

Column No.	**Definition**
1	Relative paths of images (same as those in the list files of the testing set)
2	Types of traffic lights. Use 1/2 of values to correspond to the non-green / the green lights.
3	Confidence of the detection
4	Coordinate positions on the left of the light box
5	Coordinate positions on the top of the light box
6	Coordinate positions on the right of the light box
7	Coordinate positions on the bottom of the light box

3.4 Evaluation Metrics

Intersection-over-union (IoU) threshold is 0.5, that is, the detection box is considered to be correct when the IoU of the detection box to the ground truth box is greater than 0.5. If there are several detection boxes whose IoUs of the ground truth box are greater than 0.5, the detection box with the highest confidence is selected to be the correct result. The algorithm is evaluated through the precision and recall rate as well as the average precision (AP). Among them, AP is the area under the precision-recall rate curve.

4 . Document of User Interfaces

4.1 Training Interfaces

Users need to implement their own training scripts, run_train.sh.
Among them, the training script needs to satisfy:

1. The location of the training data is specified in the script. The data preprocessing or format conversion should be conducted as needed.
2. The training logs need to be stored in the logs/train.log. The training loss should be output in the format of ‘Iteration %d, loss = %f’ for the platform to parsing and drawing.
3. The snapshot model needs to be stored in models/, named in the format of model.$Iteration (for example: the storage path for the model in the 10th round is models/model.10).

The platform will call the training script as follows.

```
./run_train.sh
```

4.2 Prediction Interfaces

Users need to implement their own prediction script, run_predict.sh.
The prediction script is called by the platform based on the period specified by users. The prediction script needs to satisfy:

1. The first parameter is the address of the model file.
2. The second parameter is the path of the input data folder. The data format refers to section 3.2.
3. The third parameter is the address of the prediction result file. The output format refers to section 3.3.
4. The prediction log is stored in the logs/predict.log.

The platform will call the prediction script as follows

```
./run_predict.sh $modelfile $image_root_dir $predict_result_file
```