AI lab project: Traffic sign detection

Packages used:

- The ElementTree XML API - https://docs.python.org/3/library/xml.etree.elementtree.html
- OpenCV - https://opencv.org/
- glob - https://docs.python.org/3/library/glob.html
- numpy - https://numpy.org/doc/
- random - https://docs.python.org/3/library/random.html
- sklearn.ensemble - https://scikit-learn.org/stable/modules/ensemble.html

In order to edit the file, run imports and functions.

In [9]:
```
import xml.etree.ElementTree as ET
import cv2
import glob
import numpy as np
import random
from sklearn.ensemble import RandomForestClassifier
```

Change this cell type to code and paste all functions in main.py except main()

Algorithms used:

parse_data(path) - `main.py, lines 9-79` : Parsing .xml files and putting data in a custom structure.

Structure below contains list of samples, where values of first sample, first object are printed. All samples are arranged in a dictionary structure. Dictionary keys are: **{** name, image, size: { width, height }, object: [{ type, bounds: { xmin, ymin, xmax, ymax }, obj_image }] **}**

In [15]:
```
train_data = parse_data('train\\annotations\\')
print(train_data[0]['name'])
# print(train_data[0]['image']) # large matrix, feel free to uncomment
print(train_data[0]['size']['width'])
print(train_data[0]['size']['height'])
print(train_data[0]['object'][0]['type'])
print(train_data[0]['object'][0]['bounds']['xmin'])
print(train_data[0]['object'][0]['bounds']['ymin'])
print(train_data[0]['object'][0]['bounds']['xmax'])
print(train_data[0]['object'][0]['bounds']['ymax'])
# print(train_data[0]['object'][0]['obj_image']) # large matrix
```

```
road0.png
267
400
2
98
62
208
232
```

balance_dataset(data, ratio) - `main.py, lines 81-87` : Reduce elements of dataset, randomly.

In [16]:
```
i = 0
for sample in train_data:
    i += 1
print('Num of elements:', i)

print('Balancing with low ratio:')
balanced = balance_dataset(train_data, 0.005)
for sample in balanced:
    print(sample['name'])
```

```
Num of elements: 877
Balancing with low ratio:
road299.png
road770.png
road739.png
road514.png
```

learn_bovw(data) - `main.py, lines 89-108` : Creates a dictionary of visual words from cropped traffic light images. Creates a voc.npy file. Remember to comment out after creating a dictionary file.

extract_features(data) - `main.py, lines 111-126` : Creates key points in cropped images and determines if they match the vocabulary in the dictionary.

train(data) - `main.py, lines 130-144` : Trains random forest model and returns it from function.

predict(rf, data) - `main.py, lines 146-156` : Performs prediction using trained model and adds results as "type_pred" (int) entry in object_data.

check_size(sample, objects) - `main.py, lines 159-170` : Returns True if size of object is greater or equal to 10% of the image size.

In [22]:
```
check_size(train_data[0], train_data[0]['object'][0])
```

Out[22]:
```
True
```

print_predicted(data) - `main.py, lines 173-193` : Prints detected crosswalks.

test_print(data) - `main.py, lines 196-207` : Function used for debugging.