Real-time Large-scale Traffic Sign Detection and Recognition

Brief about the project, TODO lists, misc.

**Abbreviations**

TSDR - Traffic Signs Detection and Recognition

# 

# Goals

* highly accurate TSDR (mAP > 90%) in real time (>20 FPS?)
* harnessing the power of GPUs (2-3 cards in parallel)
* using large-scale DFGTSD database (provided by Slovenian company DFG Consulting) with 200 categories of traffic signs and ~7000 FHD images
* Benchmark developed pipeline on other databases too (GTSDB, GTSRB, BTS, STSD, LISA, ...)

# Dataset

* DFGTSD database: <https://www.vicos.si/Downloads/DFGTSD>  
  Sizes of images (KB) in this repository are different from Aleksej’s
* Some of the selected images are on GDrive in folder ***data***
* Annotations (v1.1) are in folder ***data/annotations***
* Note: there is original database (with captured images in real world = default) and augmented one with artificially generated images (beneficial for CNN training).

Some datasets we have found through the papers reading:

* Lim2017, Vienna traffic rules (KR-D, KR-N, DE-D): <https://figshare.com/articles/Traffic_Sign_Recognition_Testsets/4597795/1>
* Hasan Fleyeh repository: <http://users.du.se/~hfl/>

# Literature

Some of the most important (?) papers are in the folder ***literature***.

The starting point are papers *Avramovic2018* and *Tabernik2019*.

# Sitemap of repository (GDrive)

* **data**
  + **annotations**
    - **augmented:** ground truth for dataset with augmentation
    - **default:** no augmentation. JSON files were parsed and saved to Matlab format (eg. train.json.mat). We have also joined train and test parts into one for convenience (joined\_train\_test.mat).
  + **masks**: binary masks as the result of image preprocessing and thresholding
    - **RGB\_Aleksej**: initial results by Aleksej (using SaliencyDetection.m)
  + **original**: small subsample of DFGTSD dataset
  + **preprocessed\***: results from preprocessing step with histogram equalization (heq) and color constancy (cc).
  + **tileImages\***: original and preprocessed images are put together for easier comparison and thresholds adjustment (using Matlab colorThresholder app).
* **docs**
* **findROI:** prototype code for finding regions of interest (areas with traffic signs) and helpers for reading annotations, image tiling, etc.
  + **sandbox**: scripts with tests and experimentation.
* **literature**

# Proposed approach

Two-stages approach:

1. fast and coarse detection of traffic signs -> extraction of K patches/ROIs (708 x 708px)
2. YOLO-based TSDR on extracted ROIs only and using K GPUs, one for each ROI

## 1. findROI

TODO



1. INPUT: RGB image   
   size 1920 x 1080 x 3 uint8 or