KI WISSEN
Automotive AI Powered by Knowledge

1.16 Leveraging Knowledge for Traffic Sign Detection

Gurucharan Srinivas | DLR

Detection

Motivation

Various deep learning models have been developed for object detection, each with different architectures. However, the vanilla versions of these models perform well when trained on large-scale datasets. When the training dataset is small, their performance may suffer. To address this issue, there are two possible solutions. The first solution is to use data-specific augmentation techniques to expand the dataset. The second solution is to use prior domain knowledge to supervise the model's training. While this method can be effective, it requires unique adaptations to the model's architecture or loss functions. This will enable the models to go beyond mere perception, granting them cognitive capabilities for reasoning and interpretability, and improving their performance in complex or novel situations.

Hybrid Architecture and Datasets

Our architecture design, illustrated in Figure 1, is inspired by the hybrid architecture in [3]. Our "Knowledge Module" utilizes relationship semantics, such as co-occurrence correlation between traffic sign classes. We integrate this knowledge within the FasterRCNN [4] object detector framework. We train our models using the DFGTS [2] and MTSD [1] traffic sign datasets.

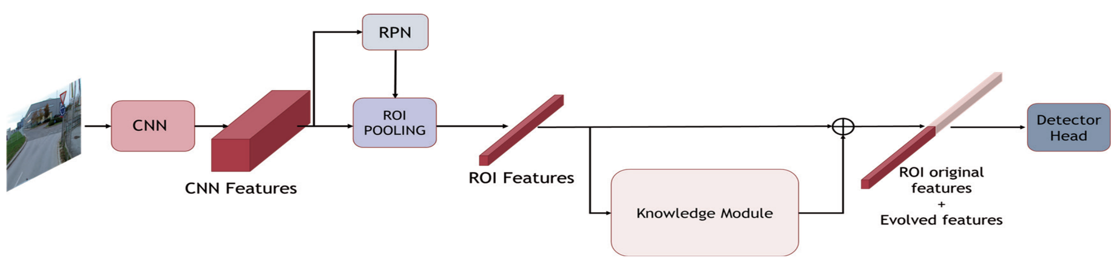

Figure 1: Illustrates the Hybrid Architecture for integrating prior knowledge within FasterRCNN framework.
(© Deutsches Zentrum für Luft- und Raumfahrt, e. V.)

Knowledge Mining and Representation

Annotations from training dataset is used for knowledge mining.

Algorithm 1 Extract and Normalize Co-occurrence Matrix

```
Initialize a square, symmetric matrix C of size n × n, where n is the number
  of unique objects in the dataset.                          ▷ Initialization
Create a mapping M from object identifiers to indices in C.
for each sample in the dataset do                       ▷ Iterate over samples
    for each object o_i in the sample do
        if o_i not in M then
            Add o_i to M with a new index.
        end if
        for each object o_j in the sample do          ▷ Pair-wise co-occurrence
            if o_j not in M then
                Add o_j to M with a new index.
            end if
            if o_i ≠ o_j then                          ▷ Exclude self-co-occurrence
                Increment C[M[o_i], M[o_j]] and C[M[o_j], M[o_i]] by 1.        ▷
  Symmetric update
            end if
        end for
    end for
end for
for each element C[i, j] in C do                             ▷ Normalization
    Calculate o_ii as the sum of the ith row, excluding C[i, i].
    Calculate o_jj as the sum of the jth row, excluding C[j, j].
    Normalize C[i, j] by √o_ii × √o_jj if o_ii × o_jj > 0. ▷ Apply normalization
end for
```

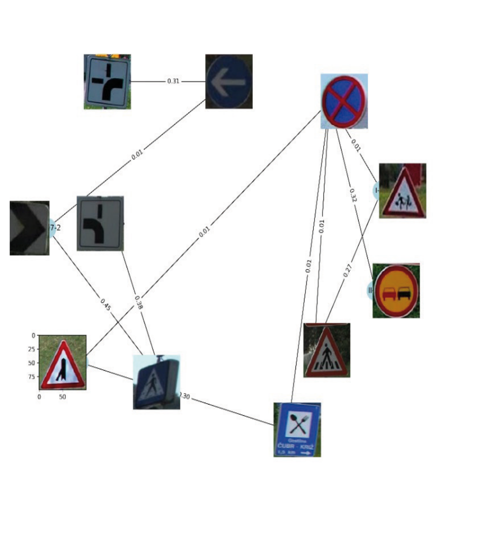

Figure 2: (Left) Algorithm to extract co-occurrence relationships from the annotated dataset. (Right) Illustration of sampled sub-graph depicting co-occurrence relationship knowledge extracted from DFGTS dataset. The nodes represent the unique classes and the edges represent co-occurence strength between class pairs in the dataset.
(© Deutsches Zentrum für Luft- und Raumfahrt, e. V.)

Knowledge Module

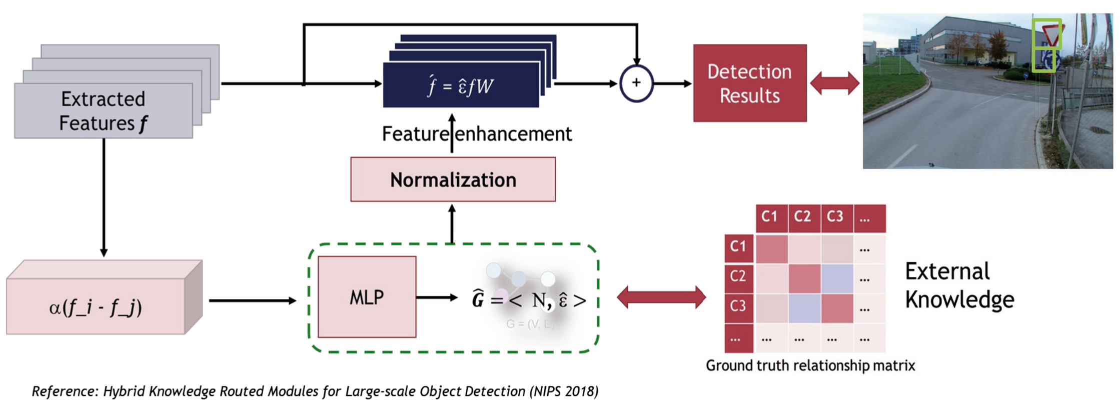

Figure 3: Internal specifics of the „Knowledge Module“, which enhances features of the ROI regions (i.e., extracted region features, f‘) using the learned edge weight from the relationship matrix.
(© Deutsches Zentrum für Luft- und Raumfahrt, e. V.)

The symmetric co-occurrence matrix, which is established and constructed using algorithm 1 (as shown in Figure 2), can be envisioned as a knowledge graph ($\mathbb{Q}$). The knowledge module is comprised of stacked Multilayer Perceptron (MLP) that are trained to predict the edge weights between region-pairs using supervision from the ground truth graph $\mathbb{Q}$. The input to the MLP is the $\mathbb{L}1$ difference between the features of each region pair (f_i, f_j). Furthermore, these learned edges between regions are then used to compute enhanced features, which encode region-wise correlations.

Results

The table shows mean Average Precision (mAP) and Average Recall (AR) for two models: a Hybrid model and a vanilla version of Faster R-CNN. The metrics are broken down by object size: large (L), medium (M), and small (S).

Models	mAP	AP_L	AP_M	AP_S	AR_L	AR_M	AR_S
Hybrid	54.92	**71.01**	33.63	**2.10**	74.37	37.15	2.24
FasterRCNN	**55.17**	70.7	**34.6**	1.87	74.4	**38.88**	2.24

Table 1: „FasterRCNN“ is with ResNet101 backbone. Whereas, the „Hybrid“ model is the FasterRCNN (ResNet101) with Knowledge Module.

The Hybrid model's metrics match the baseline, prompting us to seek out more advanced knowledge forms and representations.

Additonal Knowledge Mining

- Associative rule mining → Using „A-priori Algorithm“.
- Generate pseudo labels → Using cross sensor information.

References:

[1] Ertler, Christian, et al. "The mapillary traffic sign dataset for detection and classification on a global scale." , 2020.
[2] Tabernik, Domen, and Danijel Skočaj. "Deep learning for large-scale traffic-sign detection and recognition.“, 2019.
[3] Jiang, Chenhan, et al. "Hybrid knowledge routed modules for large-scale object detection.“, 2018.
[4] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks.“, 2015.

Partners

External partners

For more information contact:
gurucharan.srinivas@dlr.de
franz.andert@dlr.de
karsten.liesner@dlr.de

KI Wissen is a project of the KI Familie. It was initiated and developed by the VDA Leitinitiative autonomous and connected driving and is funded by the Federal Ministry for Economic Affairs and Climate Action.

www.kiwissen.de   @KI_Familie   KI Familie

KI FAMILIE
VDA LEITINITIATIVE

Funded by the European Union
NextGenerationEU

Supported by:
Federal Ministry for Economic Affairs and Climate Action
on the basis of a decision by the German Bundestag