Knowledge-Aware Object Detection in Traffic Scenes

Jean-Francois Nies[1,2], Syed Tahseen Raza Rizvi[1], Mohsin Munir[1], Ludger van Elst[1]
and Andreas Dengel[1,2]
[1]*German Research Center For Artificial Intelligence (DFKI) Kaiserslautern, Germany*
[2]*RPTU Kaiserslautern-Landau, Kaiserslautern, Germany*
{*firstname.lastname*}*@dfki.de*

Keywords: Knowledge Graphs, Perception, Computer Vision, Autonomous Driving, Knowledge Integration.

Abstract: Autonomous driving is a widely popular domain that empowers the autonomous vehicle to make crucial decisions in a constantly evolving traffic scenario. The role of perception is pivotal in the secure operation of the autonomous vehicle in a complex traffic scene. Recently, several approaches have been proposed for the task of object detection. In this paper, we demonstrate that the concept of *Semantic Consistency* and the ensuing method of *Knowledge-Aware Re-Optimization* can be adapted for the problem of object detection in intricate traffic scenes. Moreover, we also introduce a novel method for extracting a knowledge graph encoding the semantic relationship between the traffic participants from an autonomous driving dataset. We also conducted an investigation into the efficacy of utilizing diverse knowledge graph generation methodologies and in- and out-domain knowledge sources on the efficacy of the outcomes. Finally, we investigated the effectiveness of knowledge-aware re-optimization on the Faster-RCNN and DETR object detection models. Results suggest that modest but consistent improvements in precision and recall can be achieved using this method.

1 INTRODUCTION

The problem of *Object Detection* (Zou et al., 2023) can be expressed informally as the question ”*What objects are where?”*. More formally, it incorporates both the definition of bounding boxes, which encompass objects of interest as closely as possible, and the ability to assign correct labels to each such box. Recently, research conducted in this field, primarily utilizing convolutional neural networks (CNN), has already resulted in the development of object detection systems that are capable of surpassing human performance for specific tasks. (Altenberger and Lenz, 2018). Such systems typically employ deep CNNs and are widely used in a range of practical applications, such as autonomous vehicles and facial detection systems.

Although a human observer would have the ability to reason about the scene it is observing, a model such as a convolutional neural network would have limited ability to dismiss or correct the erroneous detections (LeCun et al., 2015). The presence of additional objects, the time and location of the objects, and their relative positions in relation to one another may all provide crucial clues to an observer. However, the incorporation of this context into a machine learning model is a complex undertaking. Contextual information necessitates the provision of machine-readable form, frequently resulting in substantial amounts of supplementary data, which can be processed alongside conventional inputs. Nonetheless, this approach has garnered significant attention. In addition to immediate enhancements in performance, *Informed Machine Learning* may also offer’soft’ benefits in terms of explainability and accountability. (von Rueden et al., 2023)

Although it is feasible to construct models that incorporate external knowledge from the ground up, there may be instances where the essential training data is not accessible, or where a proprietary model must be regarded as a ’black box’. In either case, the ability to perform knowledge-based re-optimization of the outputs of arbitrary models may be desirable.

In this paper, we adapt the knowledge aware object detection approach for the task of object detection in traffic scenes. This paper investigates the impact of semantic consistency on the effectiveness of the object detection approaches. We also examined the impact of combining diverse knowledge sources and the methods for estimating semantic consistency on the final detections. Moreover, we examine the efficacy of various sources and knowledge-graph generation

Nies, J., Rizvi, S., Munir, M., Elst, L. and Dengel, A.
Knowledge-Aware Object Detection in Traffic Scenes.
DOI: 10.5220/0012378100003636
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 3*, pages 617-624
ISBN: 978-989-758-680-4; ISSN: 2184-433X
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

techniques on the performance of the more recent Deformable Transformer (DETR) architecture.

2 RELATED WORK

The field of object detection is a dynamic area of research that has numerous immediate applications. It should therefore come as no surprise that several approaches are currently under active investigation.

Convolutional neural networks remain a fundamental part of object detection models, but many recent models have enhanced their capabilities by introducing different additional mechanisms. For example, the Faster R-CNN model (Ren et al., 2016) is widely used as a baseline for other object detection models. It utilizes an RPN or region proposal network to identify regions of interest before applying a convolutional neural network to the actual task of recognition within these regions. By contrast, the DETR architecture (Carion et al., 2020) prepends its CNN backbone to another model, composed of encoding and decoding transformers, such that the CNN acts as a dimensionality reduction mechanism.

On the other hand, the YOLO family of models (Redmon and Farhadi, 2018) typifies the one-shot object detection approach, named in contrast to the two stages of an FRCNN-like detector with separate region proposal and classification stages. A YOLO-style model, as indicated by expanding the acronym to ”You Only Look Once”, instead uses a single stage.

Although these models are capable of extracting large amounts of relevant information from input images, they do not generally leverage external, contextual knowledge such as causal or semantic relations between different depicted objects and image metadata. For example, an image containing cars is more likely to contain an omnibus than a lobster. Given the intuitive nature of this methodology, it is not surprising that the issue of incorporating additional knowledge in object detection has been examined from numerous perspectives beyond the one that is at the core of this paper.

Liu et al.(2021) (Liu et al., 2021) employed a *similarity network* to measure the pairwise semantic similarity between objects, in a method somewhat similar to the one discussed in this paper. However, unlike semantic consistency, this is not related to the likelihood of co-occurence of concepts. Instead, semantic similarity is a measure of the likelihood that two objects belong to the same class, regardless of which class this may actually be. Others, such as Zhu et al. (Zhu et al., 2021) have proposed architectures that are capable of a more sophisticated integration of not only semantic, but also spatial information, in order to enhance the performance of object detection models. Chen et al. (Chen et al., 2020) demonstrated that such an approach can yield improved performance, especially on small objects.

Menglong et al. (2019) (Menglong et al., 2019) applied a similar approach to the task of image classification or object recognition. For this, they described a method in which labels assigned by one or several existing classifiers are used to construct a knowledge graph. The semantic similarity between any two categories is assigned based on the frequency at which these classifiers confuse them. This information is then utilized to enhance image classification by modifying the classifiers output to better reflect the domain knowledge gathered, and determining the degree of confidence associated with outputs based on their plausibility.

The construction of knowledge graphs is moving beyond simple text databases, and into the field of Multi-Model Knowledge Graphs, organically integrating image and text data into a single structure and explicitly connecting images to explicit visual properties, as in Zhu et al. (Zhu et al., 2022).

MacAodha et al. (Aodha et al., 2019) took advantage of the fact that many images available today are tagged with temporal and geographic metadata and extended the conventional object detection models to consider this metadata as additional context. This is accomplished by utilizing an assessment of the co-occurrence of objects at specified times and locations.

Von Rueden et al. (von Rueden et al., 2023) propose the concept of ’informed machine learning’ as a broad term that encompasses all forms of machine learning that utilize external information sources, and present a comprehensive and useful taxonomy.

Finally, Castellano et al. (Castellano et al., 2022) demonstrate a novel application of these techniques by utilizing a combination of knowledge graphs and deep learning techniques to analyze artworks, and provide a corresponding knowledge graph.

3 DATASET

We evaluated the impact of Knowledge-Aware Re-Optimization on the Cityscapes Dataset (Cordts et al., 2016), which was originally designed for scene understanding as applied to urban scenes. It comprises a total of 5000 stereoscopic images, each of which is accompanied by meticulous annotations at the instance and pixel levels, as well as supplementary annotations with coarser annotations. In our instance, we take into account the annotations at the instance

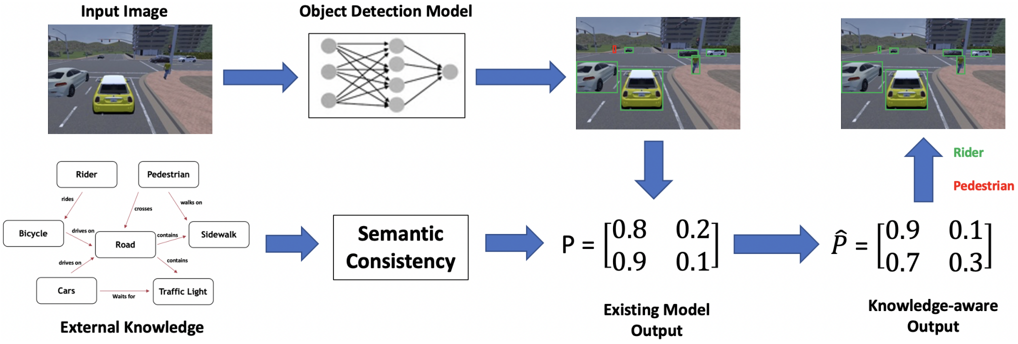

Figure 1: The basic principle of knowledge-based re-optimization. Using a semantic consistency matrix, the model output p is changed to a re-optimized output $\hat{p}$.

Table 1: Distribution of classes across the partial Cityscapes dataset.

Class	Training	Validation
Person	12044 (33.1%)	3450 (34.4%)
Rider	1080 (2.9%)	542 (5.4%)
Car	19113 (52.6%)	4378 (43.7%)
Truck	312 (0.8%)	93 (0.9%)
Bus	245 (0.6%)	98 (0.9%)
Train	118 (0.3%)	23 (0.2%)
Motorcycle	492 (1.3%)	148 (1.4%)
Bicycle	2903 (7.9%)	1281 (12.7%)
Total	36307	10013

level.

In order to incorporate the depth information required for certain applications, the dataset comprises pairs of stereoscopic images, i.e., images that were simultaneously captured by two cameras situated at a slight distance from each other but oriented in the same direction. This would allow for depth perception, as the perspective of the left and right eye would differ from human vision. As depth information was beyond the scope of our experiments, we adhered to the methodology outlined by T. Beemelmanns (Beemelmanns, 2022) for formatting cityscape datasets. This entails that only the left-hand image was retained from each pair of stereoscopic images, resulting in a dataset consisting of single images.

The number of instances in the training and validation components of the dataset is given in table 1. This table also illustrates a limitation of the dataset, namely the large class imbalance present across both the training and validation sets. Thus, the number of e.g., pedestrians (designated as ’Person’ in the dataset label but distinct from a ’Rider’ on a bicycle or motorcycle) in the validation set exceeds the number of trains or buses by two orders of magnitude. The classes corresponding to Trucks, Buses, and Trains are particularly underrepresented. Nonetheless, the ratio of each classification in relation to the total number of instances in both sets remains consistent, with the exception of the categories ’Car’ and ’Bicycle’, wherein the training and validation sets differ by 5 percentage points.

4 METHODOLOGY

In this paper, we adapt the concept of *semantic consistency* (Fang et al., 2017) for the domain of autonomous driving to encode the relationship between traffic participants as knowledge. We investigate the impact of knowledge integration on the performance of object detection models in traffic scenes by utilizing environmental knowledge. This knowledge-aware re-optimization framework relies on a semantic consistency matrix to ensure consistency. For any two concepts, this matrix gives a semantic consistency value, which is an indication of how likely instances of these two concepts are to occur simultaneously in an image. The semantic consistency of a concept with itself is also important, as multiple instances of the same concept in a single image are more likely for some concepts than others.

The framework as a whole is devoid of any particular object detection method or implementation, instead treating the object detection model as a black box. The model’s output is modified to better align with the semantic consistency matrix, whereby the scores for each class are elevated or decreased to enhance the overall consistency of the detection. To regularize the model and avoid being overly restricted in cases which do not conform to the expected distribution, significant changes in class scores are also penalized. An overview of the re-optimization workflow is shown in 1. For our experiments, we used FRCNN (Ren et al., 2016) and DETR (Carion et al., 2020) object detection models.

The main challenge for this approach lies in obtaining the semantic consistency matrix, S. We compared the performance of baseline models with three different methods, including the frequency-based method and the knowledge graph-based method, as well as our novel hybrid method.

4.1 External Knowledge Sources

4.1.1 Frequency-Based Semantic Consistency

The frequency-based approach generates a semantic consistency matrix directly from the annotated training data utilized for the backbone model, necessitating the availability of this training set. Fang et al. (Fang et al., 2017) have noted that this may pose a drawback in practical applications where the training data may be proprietary or otherwise unavailable, and also report a lower performance for this method. Conversely, however, it may be useful for situations where no suitable knowledge graph is available. The frequency-based method for determining semantic consistency is based on the co-occurrence of concepts. Hence, it is presumed that the objects that frequently occur together in the training set exhibit a greater degree of semantic consistency, as per equation 1.

$$S_{l,l'} = max(log\frac{n(l,l')N}{(n(l)n(l'))},0) \tag{1}$$

where $n(l)$ and $n(l')$ are the individual frequencies of the respective concepts l, l', $n(l,l')$ denotes the number of co-occurrences of the two concepts and N is the total number of instances. As the number of co-occurrences of multiple instances of a single class may also be relevant, a distinct 'handshake' equation is employed in this instance, adhering to the same conventions, as per equation 2

$$n(l,l) = n(l)\frac{n(l)-1}{2} \tag{2}$$

4.1.2 Knowledge Graph-Based Semantic Consistency

The Knowledge Graph-Based Semantic Consistency method is the second approach for assessing semantic consistency. Unlike the frequency-based method, it possesses the capability to be utilized even on pre-trained models that lack training data.

To achieve this objective, it is imperative to construct an external knowledge graph whose vertices represent distinct concepts and whose edges connect the concepts that are semantically related. In order to process the knowledge graph, we adhered to the methodology outlined by Lemmens et al. (Lemmens et al., 2023), who provided significant elucidation to the initial work performed by Fang et al.(Fang et al., 2017). This method involves two stages.

Initially, the pre-existing graph, in our instance, ConceptNet5 (Speer et al., 2012), has been cropped to exclusively encompass positive relations (i.e., omitting relations such as antonyms and contradictions) and is limited, without any loss of generality, to the English-language versions of concepts.

Subsequently, the semantic consistency matrix is derived from this knowledge graph by means of a series of random walks commencing from each concept of interest, corresponding to a desired label of the object detection model. The random walk then traverses the graph, but also has a low probability of resetting to the original node with every step to avoid remaining stuck in local groups (Random Walk with Restart, RWR) (Tong et al., 2006). The probability of reaching any given node from the starting node eventually converges towards a steady-state after a sufficient number of iterations.

4.1.3 Hybrid Semantic Consistency

In this paper, we evaluated the performance of our own hybrid approach for estimating semantic consistency, derived from a combination of the frequency-based and knowledge-graph based approaches. It is intuitive that the semantic consistency of a well-designed out-of-domain and generic knowledge graph will be correlated with the degree to which these concepts are intertwined within the corpus on which the graph is based. However, this may not be ideal in every situation. If a model is to be applied in a specific use-case instead of general-purpose object detection, the class distribution it encounters may be different from that suggested by a consistency matrix derived from a generic knowledge-graph. As an illustration, a knowledge graph derived from an encyclopedia may exhibit a limited number of mentions of pedestrians, while a greater emphasis is placed on trains. Whereas, an object detection model intended for self-driving vehicles is likely to encounter the opposite scenario.

As the knowledge graph-based method demonstrated superior performance in comparison to the frequency-based method in general, we investigated the feasibility of creating a knowledge graph that is tailored to our dataset based on frequency data. We presume that a meticulously crafted dataset may possess greater relevance to the domain of application of the model than a generic knowledge model. The steps of hybrid approach are as follows:

- We generate a matrix M_1 of co-occurrences for each class across our dataset.

- Then we generate a knowledge graph G based on M_1 by creating a vertex and concept for each label present in the co-occurrence matrix, and connecting them when the corresponding number of co-occurrences exceeds a threshold γ.
- A random walk is performed on this new graph G, and the semantic consistency matrix is generated as in the case of ordinary graph-based semantic consistency.

We thus obtain a semantic consistency matrix with the same structure as that produced by other methods.

4.2 Knowledge-Based Re-Optimization

After generating our semantic consistency matrix S, we proceed with the actual re-optimization. This is accomplished by determining the minimal loss function that takes into account both semantic consistency and the original input.

Formally, given two bounding boxes b, b', we call $P_{b,l}, P_{b',l'}$ the probability returned by the model for a label l resp. l' for either box. The loss function to be minimized is given by equation 3, where L is the number of concepts and B number of bounding boxes.

$$E(\hat{P}) = (1-\varepsilon)\sum_{b=1}^{B}\sum_{b'=1,b'\neq b}^{B}\sum_{l=1}^{L}\sum_{l'=1}^{L} S_{l,l'}(\hat{P}_{b,l}-\hat{P}_{b',l'})^2 + \varepsilon\sum_{b=1}^{B}\sum_{l=1}^{L} BS_{l,*}(\hat{P}_{b,l}-P_{b,l})^2 \quad (3)$$

This function can be decomposed into a sum of two terms, weighted by the hyperparameter $\varepsilon \in (0,1)$, which must be determined in practice for each dataset. ε governs the relative weight of the semantic consistency values and the backbone's output. It is therefore normally constrained to a range of $0 < \varepsilon < 1$. The first term in equation 3 effectively demands that the total square error between the final output for semantically similar concepts be minimized. The semantic consistency acts like a weight for the nested sum operations, meaning that differences between wholly orthogonal concepts ($S_{l,l'} = 0$) will have no effect at all on the final value.

The second term, in equation 3, imposes a cost for overly large deviations from the backbone model's output. The square error between the final output $\hat{P}$ and the model's output P is modified by a coefficient and added for all classes and bounding boxes, effectively acting as a check on the re-optimization. The cost function given in 3 is then minimized. This is accomplished by setting the gradient of $E(\hat{P})$ w.r.t. $\hat{P}_{b,l}$ to zero.

4.3 Metrics

The implementation of Lemmens et al. (Lemmens et al., 2023) made use of an Area-Under-Curve (AUC) style of average precision calculation, computing class-wise average precision for a range of recall thresholds running from 0 to 1.0. We retained this system, as it is especially useful for automatic parameter searches, where it implicitly balances the priority of precision and recall even if only the former is explicitly targeted.

4.4 Experiment Structure

As a first step, we established a baseline of performance metrics by evaluating the performance of both architectures on the validation set of the cityscapes dataset. We then attempted to find an optimal combination of re-optimization hyperparameters. As there are several such parameters, we resorted to an automatic parameter search using Optuna (Akiba et al., 2019), including the weight parameter ε, the number of adjacent boxes and classes considered for re-optimization, and an internal score threshold for re-optimized outputs. Given that the precision-recall curve metrics used already encompass recall in computing precision, we verified that choosing accuracy as the sole optimization target had no negative effect and carried out our experiments on this base.

Initial evaluation indicated that the best results on the unoptimized baseline were obtained when considering the 100 highest-scoring detections. As this criterion was also used by Fang et al. (Fang et al., 2017), we opted to follow this convention. However, we also considered an alternative method, using a score threshold instead of a static number of detections, and the use of different values for both the number of detections and the threshold. The results are detailed in section 5.

5 RESULTS

We investigated various implementations of baseline models and found that numerous object detection frameworks differed in terms of input formats, output formats, or both, presenting a challenge for integration with knowledge-based re-optimization, even when an explicit API is provided. As an example, Yolov3 (Redmon and Farhadi, 2018), as implemented in the mmdetection framework, necessitates image input to be provided as a file path, and returns a list of bounding boxes and confidence scores per class. In contrast, the implementation of Faster R-CNN built

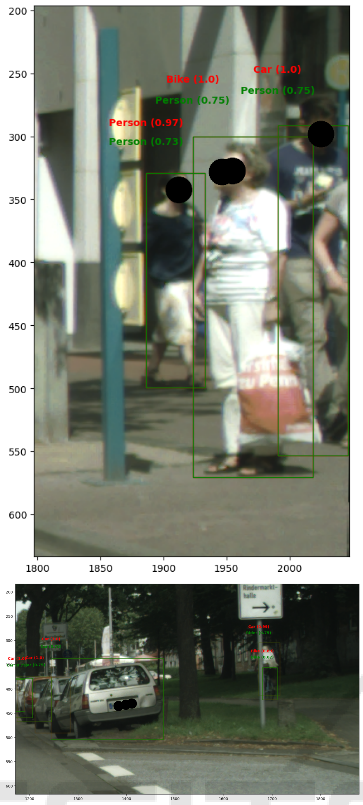

Figure 2: Examples of un-optimized output (red) being adjusted to a re-optimized label (green), with the corresponding confidence score. (Faces and license plates redacted manually).

into Pytorch necessitates image inputs in the form of pytorch tensors, and returns a dictionary containing labels, scores, along with bounding boxes in continuous tensors. It is therefore necessary to implement explicit format conversions for each architecture.

5.1 Faster-RCNN

In this section, we present and analyze the key results obtained with the Faster R-CNN architecture. These findings were computed based on the 100 top-scoring detections, and are presented for the optimal hyperparameter configuration for the given experiment. Two examples of the effects of re-optimization are given in Figure 2. We trained an FRCNN model for 40 epochs using a learning rate of 0.01, decaying to 0.0001 after 15 epochs, based on a pretrained ResNet50 backbone in pytorch.

All methods, as shown in table 2, provided some gain in recall, with the largest gains being made using the knowledge graph-based semantic consistency, which achieved a gain of 0.28%. However, the knowledge-graph (KG) method also resulted in a 0.13% decrease in mAP. The hybrid method fared worse in both respects, with smaller recall gains (0.27%) and greater mAP losses (0.34%). Finally, the frequency-based method resulted in an increase in recall slightly smaller than the previous methods at 0.24%, but also no loss of mAP (+0.05%). These results are broadly consistent with those reported for other application of the same method. (Lemmens et al., 2023).

Table 2: Effects of different re-optimization methods on the FRCNN model's performance. Numbers in brackets give the difference from the baseline.

Method	mAP	Recall	Small	Med	Large
Baseline	25.35	35.60	**12.26**	34.77	57.43
KG	25.22	**35.88**	11.93	**35.40**	57.93
Freq.	**25.40**	35.84	12.21	35.11	57.80
Hybrid	25.01	35.87	11.94	35.00	**58.40**

Table 3: Class-wise breakdown of Precision and Recall using Re-Optimization with an FRCNN backbone.

Class	AP			
	Base	KG	Freq	Hybrid
Person	25.29	24.92	24.42	**25.32**
Rider	29.76	29.40	**29.75**	29.73
Car	**44.28**	43.74	44.26	42.73
Truck	19.36	19.45	**19.49**	19.46
Bus	36.83	36.86	36.86	**36.90**
Train	12.14	**12.15**	**12.15**	11.82
Motorcycle	14.87	15.00	15.04	**15.11**
Bicycle	**20.28**	20.26	20.33	19.95

Class	Recall			
	Base	KG	Freq	Hybrid
Person	33.51	33.78	33.48	**33.94**
Rider	**41.31**	40.26	41.25	40.18
Car	49.87	50.67	49.82	**50.76**
Truck	31.51	32.58	**32.69**	32.37
Bus	46.37	**46.84**	46.73	46.63
Train	23.48	**23.91**	**23.91**	**23.91**
Motorcycle	27.18	27.65	27.72	**27.99**
Bicycle	31.21	**31.35**	31.14	31.19

Upon further examination of the results, it is evident that the average values discussed earlier do not result from a uniform change in all classes in a single direction. The knowledge graph-based and hybrid methods traded the mAP values for recall on certain classes, while achieving increased precision on others. It appears that the classes that have a smaller representation in the dataset, such as buses and motorcycles, are benefited by this approach. Thus, e.g. the hybrid method increased precision for Motorcycles by 0.24% and recall by 0.81% Conversely, the precision scores for the classes with the greatest support in the dataset, such as Person and Car, exhibit a slight decrease in precision. Cars thus soffere a loss of 0.04% in precision with the hybrid method, despite a 0.89% increase in recall. The frequency-based method was more consistent, with increases in precision across the board, except for the Car class. The latter effect may be tentatively explained by the fact that this class makes up a substantially larger portion of the training set (53.6 %), from which the frequency-based consistency matrix is derived, than of the validation set (43.7%), to which it is applied.

5.2 Deformable Transformers

In this section, we consider the results achieved with the different re-optimization methods applied to an implementation of the DETR architecture (Carion et al., 2020) on the Cityscapes dataset. As before, we will focus primarily on metrics computed for the 100 highest-scoring detections.

Performance for DETR was typically lower than for FRCNN, primarily due to slightly lower recall, as may be seen in table 4. However, the effects of the knowledge-aware re-optimization largely followed the same pattern. A challenge stemmed from the large imbalance of classes within the cityscapes dataset. While FRCNN was not visibly affected by this, we found that DETR suffered from degraded performance on the underrepresented classes. Based on the findings discussed below, it seems probable that a larger, more balanced training set or data augmentation focued on the least-represented classes would have greatly strengthened the results achieved with DETR.

The DETR model was trained for 50 epochs using the default hyperparameters set in the Deformable-DETR implementation by Zhu et al. (Zhou et al., 2020) with ResNet50-Backbone using the default pretrained weights provided by the implementation.

Table 4: Effects of different re-optimization methods on the DETR model's performance. Numbers in brackets give the difference from the baseline.

Method	mAP	Recall	Small	Med.	Lrg
Baseline	**21.84**	35.48	17.46	44.63	62.71
KG	21.81	35.63	**17.47**	**44.71**	**63.18**
Freq.	21.58	**35.70**	17.46	44.70	63.06
Hybrid	21.81	35.61	17.46	**44.71**	63.03

Table 5: Class-wise breakdown of Precision and Recall using Re-Optimization with a DETR backbone.

Class	AP			
	Base	KG	Freq	Hybrid
Person	**28.58**	28.54	28.53	28.40
Rider	27.53	27.59	27.59	**27.74**
Car	49.03	48.95	48.95	**49.10**
Truck	11.88	11.88	11.91	**12.32**
Bus	17.13	17.04	17.05	**18.33**
Train	2.52	2.55	2.53	**3.94**
Motorcycle	14.70	14.62	14.61	**15.44**
Bicycle	23.36	23.33	23.33	**23.71**

Class	Recall			
	Base	KG	Freq	Hybrid
Person	40.75	**40.77**	**40.77**	40.10
Rider	37.22	**37.24**	37.23	36.94
Car	57.66	**57.71**	57.71	**57.23**
Truck	29.73	**30.57**	29.46	28.51
Bus	29.25	**29.42**	29.39	29.40
Train	23.77	**23.84**	**23.84**	23.19
Motorcycle	**26.76**	**26.76**	**26.76**	26.74
Bicycle	38.71	38.72	38.71	**37.88**

At 2.52% in the baseline, precision for the 'Train' class is very low compared to the other classes, with only 2.52%. This may be attributed to its limited representation in the training set, with a mere 118 (0.3% of total) instances, and in particular in the validation set, with only 21 samples for trains (0.2% of total).

Conversely, when comparing the performance of Faster RCNNN in table 3 with that of DETR in table 5, it is observed that DETR's precision and recall for the highest-scoring classes 'Person' and 'Car' match or surpass FRCNN's performance (57.66% vs 44.28 precision for cars, resp. 40.75% vs. 33.51 recall for pedestrians) despite lower average values overall. As shown in table 1, these are also the most represented classes in the dataset.

However, the overall results for knowledge-based re-optimization on the Deformable DETR backbone exhibit a clearer pattern of improvements overall when compared to FRCNN, with the hybrid re-optimization method achieving a greater increase in Precision across 7 out of 8 classes. On the under-represented 'Train' class, the hybrid re-optimization method was also able to achieve a larger precision increase of 1.42%, as well as an increase of 1.2% for the 'Bus' class.

6 CONCLUSIONS

This paper employs the knowledge-based object detection approach to identify objects in traffic scenes. In this paper, we investigated the potential of utilizing knowledge-aware re-optimization for object detection on the Cityscapes dataset by applying the reoptimization model to the baseline models in the domain of autonomous driving. These experiments were conducted employing three distinct methodologies for the generation of knowledge graphs, including our novel hybrid knowledge graph generation approach.

The evaluation results suggest that certain object detection models may benefit from treating the knowledge-aware component. It was observed that more advanced architectures, such as transformers, possess sufficient sophistication that external knowledge as currently implemented may have a detrimental impact on their performance. Hence, it is imperative for the community to explore new approaches to incorporate practical knowledge into the object detection frameworks.

ACKNOWLEDGEMENTS

This work was supported by the German Ministry for Economic Affairs and Climate Action (BMWK) project KI-Wissen under Grant 19A20020G.

REFERENCES

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Altenberger, F. and Lenz, C. (2018). A non-technical survey on deep convolutional neural network architectures.

Aodha, O. M., Cole, E., and Perona, P. (2019). Presence-only geographical priors for fine-grained image classification. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9595–9605.

Beemelmanns, T. (2022). cityscapes to coco conversion.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham. Springer International Publishing.

Castellano, G., Digeno, V., Sansaro, G., and Vessio, G. (2022). Leveraging knowledge graphs and deep learning for automatic art analysis. *Knowledge-Based Systems*, 248:108859.

Chen, S., Li, Z., and Tang, Z. (2020). Relation r-cnn: A graph based relation-aware network for object detection. *IEEE Signal Processing Letters*, 27:1680–1684.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685.

Fang, Y., Kuan, K., Lin, J., Tan, C., and Chandrasekhar, V. (2017). Object detection meets knowledge graphs. International Joint Conferences on Artificial Intelligence.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Lemmens, J., Jancura, P., Dubbelman, G., and Elrofai, H. (2023). [re] object detection meets knowledge graphs. *ReScience C*, 9(1).

Liu, Y., Zhang, Z., Niu, L., Chen, J., and Zhang, L. (2021). Mixed supervised object detection by transferring mask prior and semantic similarity. *Advances in Neural Information Processing Systems*, 34.

Menglong, C., Detao, J., Ting, Z., Dehai, Z., Cheng, X., Zhibo, C., and Xiaoqiang, X. (2019). Image classification based on image knowledge graph and semantics. In *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 81–86. IEEE.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149.

Speer, R., Havasi, C., CHAIDEZ, J., VENEZUELA, J., and KUO, Y. (2012). Conceptnet 5. *Tiny Transactions of Computer Science*.

Tong, H., Faloutsos, C., and Pan, J.-Y. (2006). Fast random walk with restart and its applications. In *Sixth international conference on data mining (ICDM'06)*, pages 613–622. IEEE.

von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., Walczak, M., Garcke, J., Bauckhage, C., and Schuecker, J. (2023). Informed machine learning – a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.

Zhu, C., Chen, F., Ahmed, U., Shen, Z., and Savvides, M. (2021). Semantic relation reasoning for shot-stable few-shot object detection. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8778–8787.

Zhu, X., Li, Z., Wang, X., Jiang, X., Sun, P., Wang, X., Xiao, Y., and Yuan, N. J. (2022). Multi-modal knowledge graph construction and application: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Zou, Z., Chen, K., Shi, Z., Guo, Y., and Ye, J. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276.