Received: 18 August 2023 | Revised: 23 November 2023 | Accepted: 5 January 2024 | Published online: 15 January 2024

Artificial Intelligence and Applications
2024, Vol. 00(00) 1–7
DOI: 10.47852/bonviewAIA42021556

RESEARCH ARTICLE

How GNNs Can Be Used in the Vehicle Industry

Felipe Macías Granado[1] and **Lama Alkhaled**[1,*]

[1]*Department of Computer Science, Lulea University of Technology, Sweden*

Abstract: Graph neural networks (GNNs) have garnered substantial interest across different fields, including the automotive sector, owing to their adeptness in comprehending and managing data characterized by intricate connections and arrangements. Within the automotive realm, GNNs can be harnessed in diverse capacities to elevate effectiveness, safety, and overall operational excellence. This study is centered on the assessment of various GNN models and their potential performance within the automotive sector, utilizing widely recognized datasets. The objective of the study was to raise awareness among researchers and developers working on vehicle intelligence systems (VIS) about the potential benefits of utilizing GNNs. This could offer solutions to various challenges in this field, including comprehending complex scenes, managing diverse data from multiple sources, adapting to dynamic situations, and more. The research explores three distinct GNN models named ViG, point-GNN, and few-shot GNN. These models were evaluated using datasets such as KITTI, Mini Imagenet, and ILSVRC.

Keywords: vehicle intelligence system (VIS), few-shot learning, Graph Neural Network (GNN), Vision Transformer

1. Introduction

Applying artificial intelligence, especially machine learning, in visual applications offers a broad range of possibilities. It finds utility in various areas, including different areas such as fault diagnosis analysis [1], surveillance and security [2], natural language processing (NLP) [3], and numerous other fields [4]. This study specifically concentrates on studying computer vision (CV) as one of the main fields of artificial intelligence that is used in transportation and traffic management, referred to as an intelligent transportation system (ITS) [5].

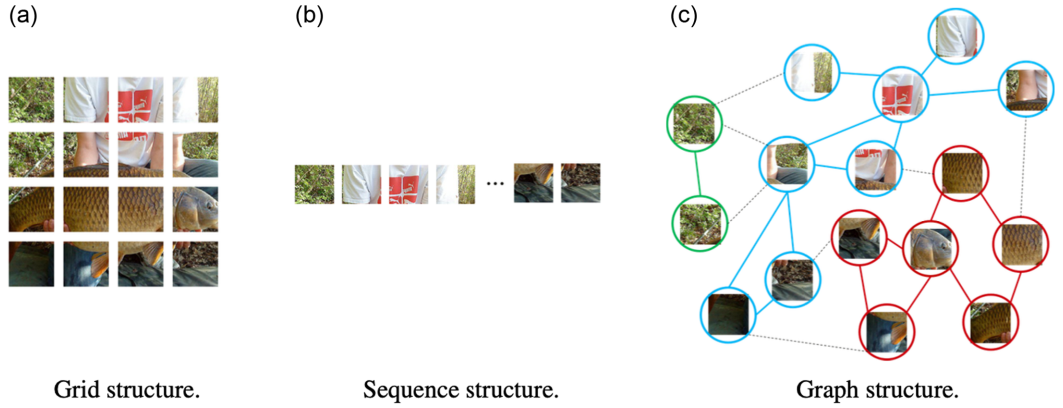
The main praxis of CV has been characterized by convolutional neural networks (CNNs). In image recognition applications, these networks have always been the most used since they have always worked relatively well. Their main drawbacks arise due to their apparent differences with human vision [6]. For example, a CNN will classify two identical images as different if noise is added to one of them. In addition, such networks also experience problems in recognizing images of the same object rotated, which is also very easy for human eyes.

The application of CV in addressing challenges within the vehicle industry is not a recent development. It commenced by introducing conventional image processing techniques to identify specific objects such as license plates in vehicles [7]. Subsequently, the research extended its scope to present more intricate challenges, examining how traditional CV techniques could operate on images captured by drones to offer valuable insights into the automotive sector [8].

*__Corresponding author:__ Lama Alkhaled, Department of Computer Science, Lulea University of Technology, Sweden. Email: lama.alkhaled@ltu.se

To advance in the direction of models closer to the behavior of human vision, several research fronts have been opened. On the one hand, vision transformers (ViT), whose main advantage over convolutional networks lies in their ability to capture long-range dependencies within an image, have begun to be used [9]. Moreover, combined with CNNs (CMTs or convolutional meets transformers), good performance in classification tasks is achieved [10]. On the other hand, self-attention-based architectures are widely used in areas such as NLP. In the area of CV, CNNs are still much more dominant. Many works have tried to combine these architectures with self-attention [11, 12]. Even with good results and consuming less computational resources, CNNs are still state-of-the-art in this area of machine learning. This is because such networks have not scaled effectively on modern hardware due to the use of specialized attention patterns [13]. Graph neural networks (GNNs) have emerged in response to the need to work with graph models. This concept has been introduced first by Gori et al. [14]. This need arises because the nature of certain data makes it much easier and more computationally efficient to work with nodes that can be related to each other. These relationships build up dependencies that can be exploited to predict different patterns of behavior. If we can classify certain behaviors, we can also predict and design them. Examples of applications where such networks can greatly improve performance include the modeling of certain physical systems, where there are certain phenomena that can be related to each other. For this same reason, GNNs are also used to predict protein interface and to classify diseases. They are useful for exploiting the relationship between GNN nodes. In the scope of this paper, GNNs are used in the field of CV and more specifically for the vehicle industry. GNNs have emerged as a powerful tool in the automotive field, revolutionizing various

Figure 1
Image to graph procedure. In the following steps on building the graph, some semantic information can be included to form more sensible relationships between nodes



aspects of vehicle-related applications. GNNs excel in modeling complex relationships and dependencies within graph-structured data, making them well suited for tasks involving intricate connections, such as traffic flow analysis, road network optimization, and ITs. In the automotive industry, GNNs are employed to enhance predictive maintenance by analyzing the relationships among various components in a vehicle, predicting potential failures, and optimizing maintenance schedules [15]. Additionally, GNNs contribute to autonomous vehicle development by modeling and understanding the interactions between different entities on the road, aiding in decision-making processes for safe navigation. The ability of GNNs to capture and leverage intricate relationships within graph-based datasets positions them as a key technology for addressing challenges and driving innovations in the dynamic and interconnected realm of automotive systems [16]. In the context of CV, GNNs are used to partition an image into a semantic graph consisting of a set of objects and their semantic relationships. This procedure can be seen in Figure 1. The theoretical concept will be faced in the following points of this section, where the main advances of the vision GNN model will be discussed [17].

2. GNN Models for Vision

The selection of GNN models such as ViG, point-GNN, and few-shot GNN over other GNN models depends on the specific requirements and characteristics of the given task or application. One of the main issues while working on vehicle-related problems is invariant representation of the collected objects from the captured scenes, this kind of known issue might recommend the use of the view-invariant representations. ViG model is designed to capture view-invariant representations in graph-structured data. Also, it is beneficial when the graph data involve diverse and multiperspective information. The other selected model is point-GNN. It is specifically designed for tasks involving point cloud data. It excels in scenarios where the input data are represented as a set of points in space. This makes it suitable for applications such as 3D object recognition, segmentation, and processing point cloud data from sensors like LiDAR. Lastly, few-shot GNN is going to be tested, as it is tailored for scenarios where the available labeled data are scarce. It addresses the challenge of learning from a small amount of labeled samples. This is particularly valuable in situations where obtaining a large labeled dataset is impractical or expensive. This kind of problem is related to most of the CV applications, and the vehicle is one of them.

2.1. ViG model

The main architecture of the model is based on the incorporation of two modules that complement each other. First, there is a module called Grapher, whose main mission is to aggregate and update graph information using graph convolution. Second, an fast forward network (FFN) module with 2 networks is used for node feature transformation and to promote node diversity. The following paragraphs will go into more detail on each of the modules. In Figure 2, the framework of the model is shown.

Figure 2
Framework of the ViG model

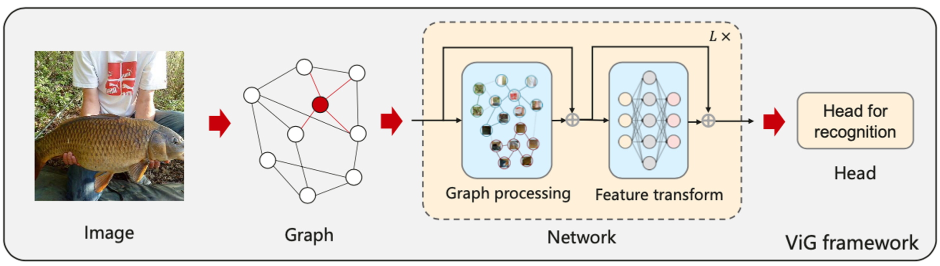


Table 1
Result of ViG model

Model	Params	Flop (B)	Top-1%	Top-5 %
ViG- Ti	7.1	1.3	73.9	92
ViG-S	22.7	4.5	80.4	95.2
ViG-B	86.6	17.7	82.3	95.9

Comparing this model with previous graph convolutional network (GCNs), it improves the feature diversity due to the introduction of more feature transformations and nonlinear activations. The general procedure consists of applying a linear layer before and after the graph convolution. With this, the Grapher module can be expressed as in Equation (1).

$$Y = \sigma(GraphConv(XW_{in}))W_{out} + X \tag{1}$$

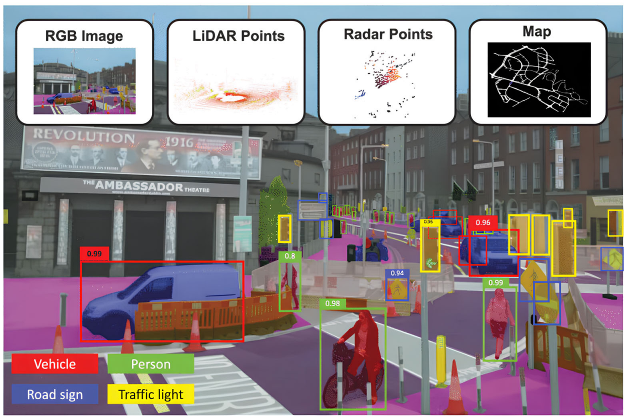
(Han et al., [17])
where W_{in} and W_{out} are the weights of the layers and σ is the activation function.

2.2. Point-GNN model

The major challenges that we should keep in mind while working on vehicle-related problems are high accuracy, high robustness, and real-time responses [19, 20]. For example, in a situation like the one in Figure 3, any slightest failure with respect to the factors mentioned above can result in a major issue in the industrial world [21, 22].

The architecture proposed by the authors of this paper is divided into three important sections: the construction of a graph starting from a point cloud, a GNN of T iterations in charge of object recognition and bounding box merging and scoring. Figure 4 shows the approach followed [23].

Figure 3
Example of a complex situation in an autonomous vehicle environment



2.2.1. Graph construction

Given a point cloud of N points $P = \{p_1, p_2, \ldots, p_N\})$, a graph G = (P,E) is constructed, being E expressed in Equation (2)

$$E = \{(p_i, p_j) \mid ||x_i - x_j||_2 < r\} \tag{2}$$

(Shi & Rajkumar, [23])

Table 2
Result of point-GNN model on KITTI

	Car		
Models	Easy	Moderate	Hard
Point-GNN	88.33	79.47	72.29
Voxel net [18]	81.97	65.46	62.85
	Pedestrian		
Models	Easy	Moderate	Hard
Point-GNN	51.92	43.77	40.14
Voxel net [18]	57.86	53.42	48.87
	Cyclist		
Models	Easy	Moderate	Hard
Point-GNN	78.6	63.48	57.07
Voxel net [18]	67.16	47.65	45.11

Then, in order to finish building the graph, a cell list is used to find point pairs that are within a given cutoff distance. It is also notable to mention that building a graph from thousands of points compromises computer performance, so the authors propose to use a voxel downsampled point cloud $\hat{P}$ for the graph construction.

2.2.2. Graph neural network

Vertex features are refined and updated by adding features along the edges. Vertex features are updated as follows (see Equation (3)):

$$\left.\begin{aligned} v_i^{t+1} &= g^t(\rho(\{e_{ij}^t|(i,j) \in E\}), v_i^t) \\ e_{ij}^t &= f^t\left(v_i^t, v_j^t\right) \end{aligned}\right\} \tag{3}$$

(Shi & Rajkumar, [23])
The autoregistration mechanism is then introduced to predict an alignment offset under the assumption that the central vertex contains some structural features from the previous iterations. Finally, multilayer perceptrons (MLP) are used as shown in Equation (4).

$$\left.\begin{aligned} x_i^t &= MLP_h^t(s_i^t) \\ e_{ij}^t &= MLP_f^t\left(\left[x_j - x_i + \Delta x_i^t, s_j^t\right]\right) \\ s_i^{t+1} &= MLP_g^t\left(Max\left(e_{ij}|(i,j) \in E\right)\right) + s_i^t \end{aligned}\right\} \tag{4}$$

2.2.3. Box merging and scoring

To ensure that the bounding boxes are outputted correctly, you must merge the different outputs of the network, since multiple vertices can be in the same object. Besides this, it is necessary to assign a confidence score. This score is computed as the sum of the classification scores weighted by the intersection-of-union (IoU) [23].

2.3. Few-shot GNN

The concept of one-shot learning was initially presented by Li et al. [24]. In their work, they posited that leveraging knowledge from existing classes could aid in predicting outcomes for new classes, even when only one or a few labels are accessible. In this work, the model is based on a GNN, made of blocks-k (see Figures 5 and 6). These blocks are mainly composed by two different structures: one that computes an N × N matrix representation of the graph structure characterized as the adjacency matrix A(k) and

Figure 4
Architecture structure followed by point-GNN

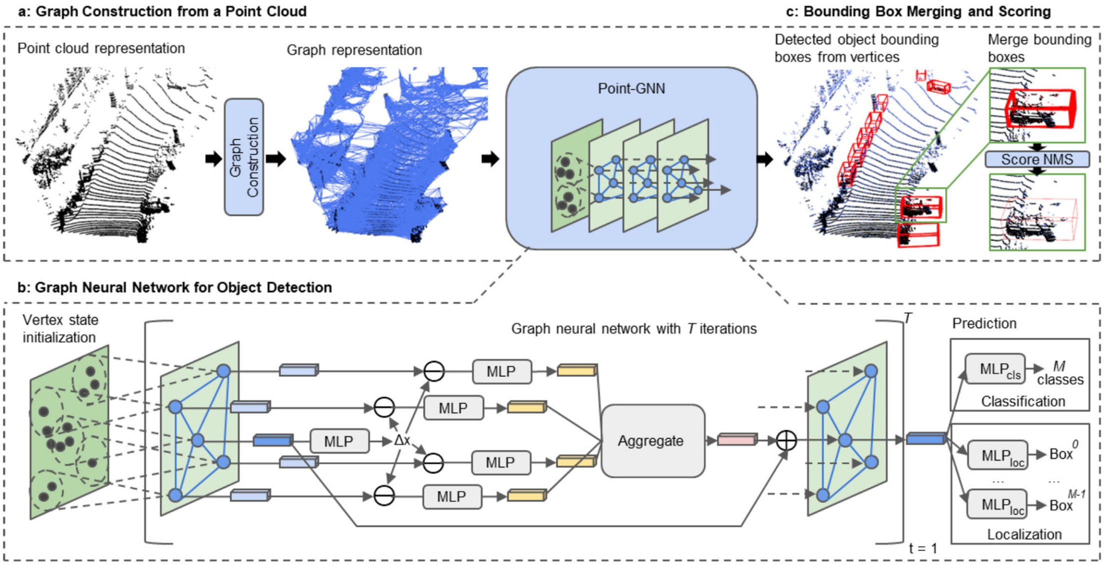
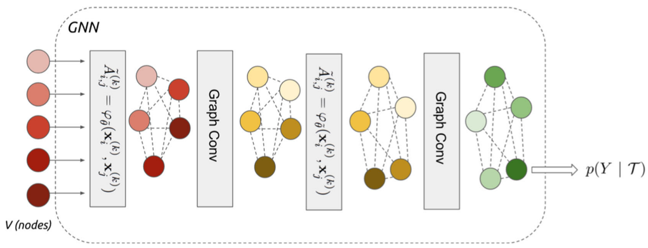


Table 3
Result of few-shot GNN model

Methods	Technique	Iterations	Accuracy
5-way	1 shot	1000	92.65
5-way	5-shot	1000	97.035

Figure 5
GNN illustration

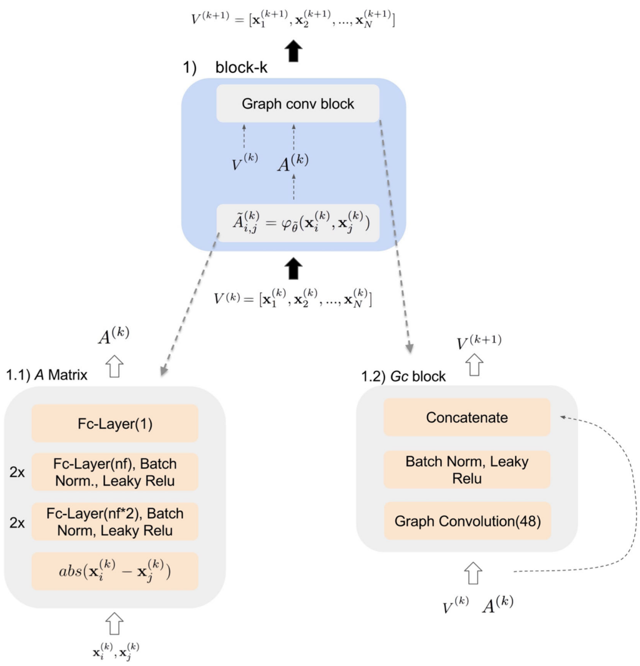


a graph convolutional block that computes an input feature matrix N* F^0 feature matrix, $V^{(K+1)}$ where N is the number of nodes and F^0 is the number of input features for each node. To compute the adjacency matrix, the input features pass through a set of fully connected layers. In these terms, it is considered an MLP stacked after the absolute difference between two vector nodes, as Equation (5) indicates.

$$\phi_{\tilde{\theta}}\left(x_i^{(k)}, x_j^{(k)}\right) = MLP_{\tilde{\theta}}\left(abs\left(x_i^{(k)}, x_j^{(k)}\right)\right) \tag{5}$$

(Li et al., [24])
In the graph convolution block, the input feature matrix $V^{(K+1)}$ is obtained after giving to it the adjacency matrix A^k and the input feature matrix V^k as inputs [25].

Figure 6
GNN model. Three blue blocks are used for Omniglot and Mini-Imagenet (*nf* = 96)



3. Experimental Results

Different datasets related to vehicle industries are proposed to use to evaluate the performance of the various GNN modelsas shown in Table 1.

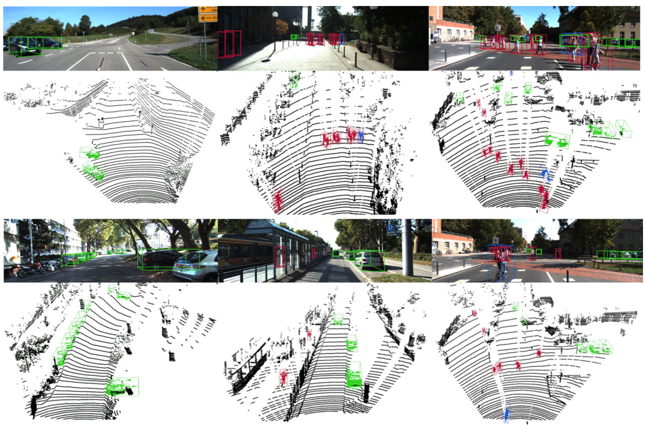
For the first proposed model ViG, ImageNet ILSVRC 2012 [26] has been used, it has 120M training images and 50K validation images, with 1000 categories.

This conducted experiment aims to use the facilities provided by working with nodes and graphs in the context of CV. As we

Table 4
Result of different GNNs' models on KITTI

	Car		
Models	Easy	Moderate	Hard
Point-GNN	88.33	79.47	72.29
ViG	76.12	65.23	52.6
Few-shot GNN	91.23	86.9	78.4
Voxel net [18]	81.97	65.46	62.85
	Pedestrian		
Models	Easy	Moderate	Hard
Point-GNN	51.92	43.77	40.14
ViG	40.3	22.6	16.8
Few-shot GNN	76.5	66.8	58.9
	Cyclist		
Models	Easy	Moderate	Hard
Point-GNN	78.6	63.48	57.07
ViG	67.16	47.65	45.11
Few-shot GNN	89.3	78.9	70.5

Figure 7
Qualitative results of point-GNN performance with the KITTI dataset



have already mentioned, working with graphs qualitatively improves the performance of the systems. In our context, using graph convolution directly on the generated graph results in low performance. For this reason, more feature transformation has been introduced to improve the diversity of information. This architecture demonstrates a clear superiority over the other architectures, so we can conclude that ViG architecture can serve as a basis for future projects.

On testing the next model, KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) [27] as one of the known datasets related to vehicles will be used to evaluate the performance. The KITTI dataset is one of the most widely used and popular datasets within the area of autonomous driving. It consists of 7481 training samples and 7518 testing samples. Each sample has an image and an associated point cloud as shown in Table 2 and Figure 7.

In this experiment, GNN proves to work better for the purpose of detecting 3D objects from a graph representation of the point clouds obtained by a LiDAR sensor as shown in Table 3. As we have previously observed in the table of results, this network is generally superior to the others in terms of object recognition in scene understanding for autonomous driving. This demonstrates, among other things, the superiority of GNNs over CNNs in this area. This superiority will translate into a proliferation in the implementation of these techniques in the future. We will see how these techniques evolve and combine with others, making the world of autonomous driving more and more developed, generating better, safer, and more accessible models for the general population. In the following experiment, few-shot model will be used over KITTI dataset. The model has been conducted in a virtual environment with Ubuntu 22.04, Python 3.8, and Pytorch 1.12. In this case, only the image sequences with their respective labels have been used, unlike the second experiment, where the data corresponding to those captured by the LiDAR sensor were used. This dataset has 8 different classes as described in Table 4.

These results indicate that, even with fewer iterations than with the Mini-ImageNet dataset [28], we have obtained a better object recognition accuracy. The main difference between ImageNet and Mini-ImageNet is that Mini-ImageNet typically includes a reduced number of classes compared to the full ImageNet dataset.

The analysis of three distinct GNN models reveals that employing the few-shot learning technique results in higher accuracy compared to the other two models. This is particularly valuable in scenarios where we have a shortage of labeled data inputs. In our specific field, although we possess extensive datasets with labeled inputs, it remains intriguing to apply these techniques within the realm of autonomous driving. Exploring these methodologies could potentially pave the way for novel research domains and innovative approaches that build upon the aforementioned concepts.

4. Conclusion

The integration of GNN models in the automotive industry holds the promise of significantly improving task outcomes and effectively tackling various challenges. Generally, harnessing the capabilities provided by node and graph operations in CV enhances performance across diverse systems. GNNs exhibit notable effectiveness, especially in the realm of 3D object detection using graph representations derived from LiDAR sensor point clouds. Additionally, few-shot GNN demonstrates superior object recognition accuracy with fewer iterations, as demonstrated in the Mini-Imagenet dataset. Testing these techniques with limited data is valuable in scenarios where abundant labeled inputs are scarce. While our specific domain benefits from extensive datasets, exploring these techniques in the context of autonomous driving could uncover new research areas and innovative methods. It is important to note that not all issues necessitate complex GAN models, and careful consideration of specific task requirements, data characteristics, and constraints should guide model selection. The nature of the data, available computational resources, and the desired balance between accuracy and efficiency in real-time applications should influence model choices, with attention to the automotive field's demands for robustness, interpretability, and safety.

5. Future Work

Our next focus will be on constructing a predictive model for animal trajectory movements within the context of autonomous driving. This model would anticipate the movements of animals, drawing from their typical biological behaviors observed on our roadways. Such an initiative would play a crucial role in accident prevention, enabling vehicles to predict animal trajectories with varying degrees of precision. Consequently, vehicles could make more informed decisions in these scenarios, contributing to enhanced safety measures.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Author Contribution Statement

Felipe Macías Granado: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Lama Alkhaled:** Writing – review & editing, Visualization.

Supportive Materials

For those interested, the code associated with this study is available on the following GitHub repository: https://github.com/felipemagr/few-shot-KITTI

References

[1] Al-Masri, A. N. A., & Mokayed, H. (2021). Intelligent fault diagnosis of gears based on deep learning feature extraction and particle swarm support vector machine state recognition. *Journal of Intelligent Systems and Internet of Things*, *4*(1), 26–40. https://doi.org/10.54216/JISIoT.040102

[2] Mokayed, H., Quan, T. Z., Alkhaled, L., & Sivakumar, V. (2022). Real-time human detection and counting system using deep learning computer vision techniques. *Artificial Intelligence and Applications*, *1*(4), 205–213. https://doi.org/10.47852/bonviewAIA2202391

[3] Alkhaled, L., Adewumi, T., & Sabry, S. S. (2023). Bipol: A novel multi-axes bias evaluation metric with explainability for nlp. *Natural Language Processing Journal*, *4*, 100030. https://doi.org/10.1016/j.nlp.2023.100030

[4] Elyan, E., Vuttipittayamongkol, P., Johnston, P., Martin, K., McPherson, K., Moreno-García, C. F., . . . , & Sarker, M. M. K. (2022). Computer vision and machine learning for medical image analysis: Recent advances, challenges, and way forward. *Artificial Intelligence Surgery*, *2*(1), 24–45. https://doi.org/10.20517/ais.2021.15

[5] Dilek, E., & Dener, M. (2023). Computer vision applications in intelligent transportation systems: A survey. *Sensors*, *23*(6), 2938. https://doi.org/10.3390/s23062938

[6] Jang, H. (2021). *Exploring the robust nature of human visual object recognition through comparisons with convolutional neural networks*. PhD Thesis, Vanderbilt University.

[7] Mokayed, H., Meng, L. K., Woon, H. H., & Sin, N. H. (2014). Car plate detection engine based on conventional edge detection technique. In *Proceedings of the International Conference on Computer Graphics, Multimedia and Image Processing*, 101–106.

[8] Mokayed, H., Palaiahnakote, S., Alkhaled, L., & AL-Masri, A. N. (2022). License plate number detection in drone images. *Artificial Intelligence and Applications*, *3*(1), 1–9. https://doi.org/10.47852/bonviewAIA2202421

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . , & Polosukhin, I. (2017). Attention is all you need. In *31st Conference on Neural Information Processing Systems.*

[10] Guo, J., Han, K., Wu, H., Tang, Y., Chen, X., Wang, Y., & Xu, C. (2022). Cmt: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12175–12185.

[11] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *Computer Vision – ECCV: Proceedings of the 16th European Conference*, 213–229. https://doi.org/10.1007/978-3-030-58452-8_13

[12] Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7794–7803.

[13] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., . . . , & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv Preprint:2010.11929*.

[14] Gori, M., Monfardini, G., & Scarselli, F. (2005). A new model for learning in graph domains. In *Proceedings of IEEE International Joint Conference on Neural Networks*, *2*, 729–734. https://doi.org/10.1109/IJCNN.2005.1555942

[15] Wang, R., Zhang, Y., Fortino, G., Guan, Q., Liu, J., & Song, J. (2022). Software escalation prediction based on deep learning in the cognitive internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *23*(12), 25408–25418. https://doi.org/10.1109/TITS.2022.3140903

[16] Li, Z., Gong, J., Lu, C., & Yi, Y. (2021). Interactive behavior prediction for heterogeneous traffic participants in the urban road: A graph-neural-network-based multitask learning framework. *IEEE/ASME Transactions on Mechatronics*, *26*(3), 1339–1349. https://doi.org/10.1109/TMECH.2021.3073736

[17] Han, K., Wang, Y., Guo, J., Tang, Y., & Wu, E. (2022). Vision GNN: An image is worth graph of nodes. In *Proceedings of 36th Conference on Neural Information Processing Systems*, *35*, 8291–8303.

[18] Zhou, Y., & Tuzel, O. (2018). Voxelnet: End-to-end learning for point cloud based 3D object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4490–4499.

[19] Mokayed, H., Clark, T., Alkhaled, L., Marashli, M. A., & Chai, H. Y. (2022). On restricted computational systems, real-time multi-tracking and object recognition tasks are possible. In *IEEE International Conference on Industrial Engineering and Engineering Management*, 1523–1528. https://doi.org/10.1109/IEEM55944.2022.9989755

[20] Mehta, A. A., Padaria, A. A., Bavisi, D. J., Ukani, V., Thakkar, P., Geddam, R., . . . , & Abraham, A. (2023). Securing the future: A comprehensive review of security

challenges and solutions in advanced driver assistance systems. *IEEE Access*, *12*, 643–678. https://doi.org/10.1109/ACCESS.2023.3347200

[21] Feng, D., Haase-Schütz, C., Rosenbaum, L., Hertlein, H., Glaeser, C., Timm, F., . . . , & Dietmayer, K. (2020). Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, *22*(3), 1341–1360. https://doi.org/10.1109/TITS.2020.2972974

[22] Neuhold, G., Ollmann, T., Rota Bulo, S., & Kontschieder, P. (2017). The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, 4990–4999.

[23] Shi, W., & Rajkumar, R. (2020). Point-GNN: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1711–1719.

[24] Li, F. F., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(4), 594–611. https://doi.org/10.1109/TPAMI.2006.79

[25] Garcia, V., & Bruna, J. (2017). Few-shot learning with graph neural networks. *arXiv Preprint:1711.04043*. https://doi.org/10.48550/arXiv.1711.04043

[26] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., . . . , & Li, F. F. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[27] Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361. https://doi.org/10.1109/CVPR.2012.6248074

[28] Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. In *30th Conference on Neural Information Processing Systems*.

How to Cite: Granado, F. M., & Alkhaled, L. (2024). How GNNs Can Be Used in the Vehicle Industry. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA42021556