RESEARCH ARTICLE

How GNNs Can Be Used in the Vehicle Industry

Felipe Macías Granado1 and Lama Alkhaled1,*

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 Graph Neural Network (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 Graph Neural Networks (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

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

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 (Al-Masri & Mokayed, 2021), surveillance and security (Mokayed et al., 2022a), natural language processing (Alkhaled et al., 2023), and numerous other fields. 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).

The main praxis of CV has been characterized by CNNs (Convolutional Neural Networks). 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. 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 computer vision 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 (Mokayed et al., 2014). Subsequently, the research extended its scope to present more intricate challenges, examining how traditional computer vision techniques could operate on images captured by drones to offer valuable insights into the automotive sector (Mokayed et al., 2022c).
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 (ViTs), whose main advantage over convolutional networks lies in their ability to capture long-range dependencies within an image, have begun to be used (Vaswani et al., 2017). Moreover, combined with CNNs (CMTs or Convolutional Meets Transformers), good performance in classification tasks is achieved (Guo et al., 2022). On the other hand, Self-attention-based architectures are widely used in areas such as NLP (Natural Language Processing). In the area of Computer Vision, CNNs are still much more dominant. Many works have tried to combine these architectures with self-attention (Wang et al., 2018; Carion et al., 2020). 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 (Dosovitskiy et al., 2020). Graph Neural Networks have emerged in response to the need to work with graph models. This concept has been introduced first by Gori et al. (2005). 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 Computer Vision and more specifically for the vehicle industry. Graph Neural Networks (GNNs) have emerged as a powerful tool in the automotive field, revolutionizing various 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 intelligent transportation systems. 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 (Wang et al., 2022). 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 data sets positions them as a key technology for addressing challenges and driving innovations in the dynamic and interconnected realm of automotive systems (Li et al., 2021). In the context of Computer Vision, 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 (Han et al., 2022).

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

2. GNN Models for Vision

The selection of Graph Neural Network (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 involves diverse and multi-perspective 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 is 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 is 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. Secondly, a FFN (Fast Forward Network) 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](image)

Comparing this model with previous GCNs (Graph Convolutional Network), 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 Eq. 1.1.

\[
Y = \sigma(\text{GraphConv}(XW_{in}))W_{out} + X
\]

Eq.1(Han et al., 2022)

where \(W_{in}\) and \(W_{out}\) are the weights of the layers and \(\sigma\) 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 (Mokayed et al., 2022c). 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.

![Figure 3: Example of a complex situation in an autonomous vehicle environment (Feng et al., 2020; Neuhold et al., 2017)](image)

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 (Shi & Rajkumar, 2020).

![Figure 4: Architecture structure followed by Point-GNN](image)

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 Eq. 2

\[
E = \{(p_i, p_j) \mid \|x_i - x_j\|_2 < r\}
\]

Eq.2 (Shi & Rajkumar, 2020)

Then, in order to finish building the graph, a cell list is used to find point pairs that are within a given cut-off 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 \(P^*\) 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 Eq.3):

\[ v_i^{t+1} = g_i^t(f_i^t((e_{ij}^t | (i,j) \in E), v_i^t)) \]

Eq.3 (Shi & Rajkumar, 2020)

\[ e_{ij}^t = f_i^t(v_i^t, v_j^t) \]

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, MLP (Multi-Layer Perceptrons) are used as shown in Eq. 4.

\[ x_i^t = MLP_h^t(s_i^t) \quad \text{Eq.1} \]

\[ e_{ij}^t = MLP_f^t([x_j - x_i + \Delta x_i^t, s_j^t]) \quad \text{Eq.4} \]

\[ s_i^{t+1} = MLP_p^t\left(\max(e_{ij}^t | (i,j) \in E)\right) + s_i^t \]

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) (Shi & Rajkumar, 2020).

2.3. Few-shot GNN

The concept of one-shot learning was initially presented by Li et al. (2006). 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 a graph convolutional block that computes an input feature matrix N × F0 feature matrix, \( V^{(k+1)} \) where N is the number of nodes and F0 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 a MLP stacked after the absolute difference between two vector nodes, as Eq. 5 indicates.

\[ \phi_\theta(x_i^{(k)}, x_j^{(k)}) = MLP_\theta(\text{abs}(x_i^{(k)}, x_j^{(k)})) \]

Eq.5 (Li et al., 2006)

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 (Garcia & Bruna, 2017).

3. Experimental Results

Different datasets related to vehicle industries are proposed to use to evaluate the performance of the various GNN models.

For the first proposed model ViG, ImageNet ILSVRC 2012 (Russakovsky et al., 2015) has been used, it has 120M training images and 50K validation images, with 1000 categories.

<table>
<thead>
<tr>
<th>Table 1 Result of ViG model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>ViG B</td>
</tr>
<tr>
<td>ViG S</td>
</tr>
<tr>
<td>ViG Ti</td>
</tr>
<tr>
<td>Ti</td>
</tr>
</tbody>
</table>
This conducted experiment aims to use the facilities provided by working with nodes and graphs in the context of Computer Vision. As we 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) (Geiger et al., 2012) 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.

### Table 2
**Result of Point-GNN model on KITTI**

<table>
<thead>
<tr>
<th>Models</th>
<th>Car</th>
<th>Easy</th>
<th>Moderate</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point-GNN</td>
<td>88.33</td>
<td>79.47</td>
<td></td>
<td>72.29</td>
</tr>
<tr>
<td>Voxel Net [18]</td>
<td>81.97</td>
<td>65.46</td>
<td></td>
<td>62.85</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Models</th>
<th>Pedestrian</th>
<th>Easy</th>
<th>Moderate</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point-GNN</td>
<td>51.92</td>
<td>43.77</td>
<td></td>
<td>40.14</td>
</tr>
<tr>
<td>Voxel Net [18]</td>
<td>57.86</td>
<td>53.42</td>
<td></td>
<td>48.87</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Models</th>
<th>Cyclist</th>
<th>Easy</th>
<th>Moderate</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point-GNN</td>
<td>78.6</td>
<td>63.48</td>
<td></td>
<td>57.07</td>
</tr>
<tr>
<td>Voxel Net [18]</td>
<td>67.16</td>
<td>47.65</td>
<td></td>
<td>45.11</td>
</tr>
</tbody>
</table>

### Table 3
**Result of few-shot GNN model**

<table>
<thead>
<tr>
<th>Method</th>
<th>Shot</th>
<th>Technique</th>
<th>Iterations</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-Way</td>
<td>1 shot</td>
<td></td>
<td>1000</td>
<td>92.65</td>
</tr>
<tr>
<td>5-Way</td>
<td>5-shot</td>
<td></td>
<td>1000</td>
<td>97.035</td>
</tr>
</tbody>
</table>

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 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 was used. This dataset 8 different classes.

These results indicate that, even with fewer iterations than with the Mini-ImageNet dataset (Vinyals et al., 2016), 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 computer vision 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 on 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's 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.

Supportive Materials

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

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

The data that support the findings of this study are openly available in [repository name e.g. “figshare”] at [http://doi.org/[doi]].

References


