Social Network Images Vulnerability Detection Using Graph Neural Network (GNN)

MD Rakibul Hassan[1*], MD Rakibul Hassan[1], Jharka Akter Jhumur[1], Ayesha Siddika[1], Kazi M. Amanat Kamal[2], Md. Nazmus Sakib[1]

[1]Department of Computer Science & Engineering, World University of Bangladesh
[2]School of Computing and Mathematics, Charles Sturt University, Wagga Wagga, NSW, 2678

*Corresponding author
doi: https://doi.org/10.21467/proceedings.123.9

ABSTRACT

The empirical study involves detecting vulnerable images in social media. Use a vulnerable image detection dataset with a 78% data accuracy rate using a graph neural network. The study investigates that sharing vulnerable pictures on social networks might cause cyberbullying or stalking victims leads to many stressful events. The inexcusable behavior by many people is encompassing in the digital world. This phenomenon leads to threats of violence, slander, or even leakage of personal images may be included. The study associates vulnerability detection with the pictures posted on social media. In this way, the people of the society will be able to identify the vulnerable images and help to build social awareness. The ability of GNN to model the dependence between nodes in a network allows for a breakthrough in graph analysis research. A graph autoencoder framework can help GNNs solve the network embedding challenge. The approach associate's construction of a custom dataset with two classes, involve Vulnerable and Nonvulnerable images. We declare image size and batch size. After creating the GNN model data is trained using 10 epochs and then plot the accuracy and loss curve. The significance of the accuracy rate reached a satisfactory level. For the statistical approach, a confusion matrix is constructed to compare the actual and prediction data this helps to understand the performance of the proposed model. Graph neural network is very effective to classify images and other things. The findings constructed a Graph Neural Networks (GNNs) is an effective framework for representation learning of graphs.

Keywords: Graph Neural Network, Vulnerability Detection, Social Network Images, Deep Learning.

1 Introduction

Social media platforms are extensively conveyed visual information to various populations. However, these platforms are uses to spread rumors and vulnerable images to a big audience in a short amount of time, potentially causing panic, anxiety, and financial loss to society. As a result, it is critical to recognize and control harmful rumors before they spread to the public. The study assesses the pictures that are assumed to be vulnerable are incorporated in the dataset. To quickly determine which ones are vulnerable. Occasionally people share the information with photographs that may, for instance, depict one man killing another with a gun or knife. As a result, a Graph Neural Network algorithm requires to determine the images are vulnerable on social media. Hence, a vulnerable image detection dataset is very important as it helps to differentiate between vulnerable and non-vulnerable images. We examine the results of the Graph Convolutional Network (GCN), a type of GNN, for

Proceedings DOI: 10.21467/proceedings.123; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-954993-5-9

detecting images data. According to a recent study, human judgment is far superior to machine learning. As a result, research activities in this industry have decreased. Detecting computer graphics images based on statistics collected from their wavelet decomposition, or residual images has been proposed in some works. Deep learning was only recently employed for this challenge and shown to outperform previous algorithms. When harmful images can be spotted readily by people, the necessity for specialized detectors becomes less pressing. However,computer graphics technology is rapidly evolving, making it increasingly difficult for onlookers to discern between computer-generated deep-fakes and actual photos. Indeed, new forms of imagemanipulation based on computer graphics have been recently devised, characterized by a much higher level of photorealism. By using converting images to graph data, more accurate results could be achieved. The study investigates the proposed graph neural network as a trained model. Here, two types of data that might are excluded include blur photographs and text. The study includes high-resolution photographs and node points of the photographs that are converted to CV files as graphs can show more about their symmetries and structure. Image classification is a basic computer vision task.

2 Literature Review

Graph Neural Networks was first proposed in 2005, but they have only recently gained traction[1]. GNNs can generate a numerical representation of the relationship between nodes in a graph by modelling it[2]. GNNs are extremely important because there are so much real-world data that can be represented as a graph[3]. Social networks, chemical molecules, maps, and transportation systems are just a few examples. Graphs are a type of data structure that depicts a collection of items (nodes) and their relationships (edges)[4]. Because of the expressive power of graphs, which can be used to denote a wide range of systems in fields such as social science, natural science (physical systems and protein-protein interaction networks), knowledge graphs, and a variety of other research areas, research on analyzing graphs with machine learning has recently received a lot of attention[5]. Graph analysis is a unique non-Euclidean data structure for machine learning that focuses on node classification, link prediction, and clustering. GNNs (graph neural networks) are deep learning-based approaches that operate on graphs[6]. Because of its superior performance, GNN has recently become a popular graph analysis tool[7]. Because graph data is so complex, it presents numerous challenges to existing machine learning algorithms[8]. The reason for this is that traditional Machine Learning and Deep Learning tools are limited to simple data types[9, 10]. As an example, consider images with the same structure and size as fixed-size grid graphs. The concept of Node Embedding is used in graph theory[11]. It entails mapping nodes to a two-dimensional embedding space (a low-dimensional space rather than the actual dimension of the graph) so that similar nodes in the graph are embedded close to each other[12]. Image classification is a fundamental task in computer vision[13]. When given a large training set of labelled classes, the majority of the models produce appealing results. The goal now is to improve these models' performance on zero-shot and few-shot learning tasks. GNN appears to be a good fit for this. Knowledge graphs can provide the information needed to guide a ZSL (zero-shot learning) task[14]. Other computer vision applications include object detection, interaction detection, and region classification[15]. guns are used to calculate RoI features in object detection; in interaction detection, GNNs is message-passing tools between humans and objects; and in region classification, GNNs perform reasoning on graphs that connect regions and classes[16].

3 Materials and Methods

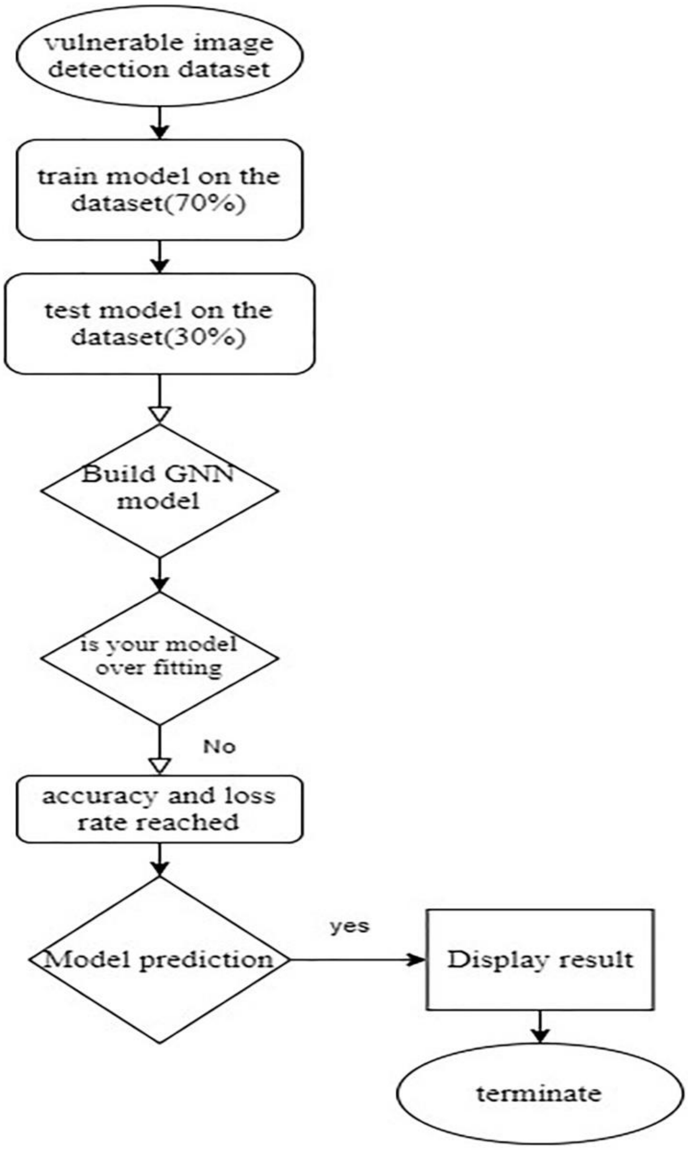

Figure 1: Working Procedure

At first, select a dataset for the study which is a vulnerable image detection dataset, and then split it into train and test sets. The study associates 70% data for the test set and 30% data for the train set in the proposed GNN model. Dataset is deployed Google Collab. This is also ensured that much data has been feeded to the proposed GNN model. This also associates the selection of some dangerous objects for object detection this includes both knives and guns. The investigation provides the machine to train and detect dangerous images from the social networks dataset. If the machine detects a gun or knife then it shows the image is dangerous. For this investigation, python and several excellent python libraries are introduced. Then it shows the final validation loss of this study is 1.50. Then add proposed algorithm which is the Graph Neural Network algorithm. For the betterment of the result statistical investigation is carried out based on the accuracy rate that reached approximately 78%. Accuracy is defined as

Proceedings DOI: 10.21467/proceedings.123
ISBN: 978-81-954993-5-9

Series: AIJR Proceedings
ISSN: 2582-3922

the percentage of correct predictions for the test data. Then finally visualization of the data result is projected.

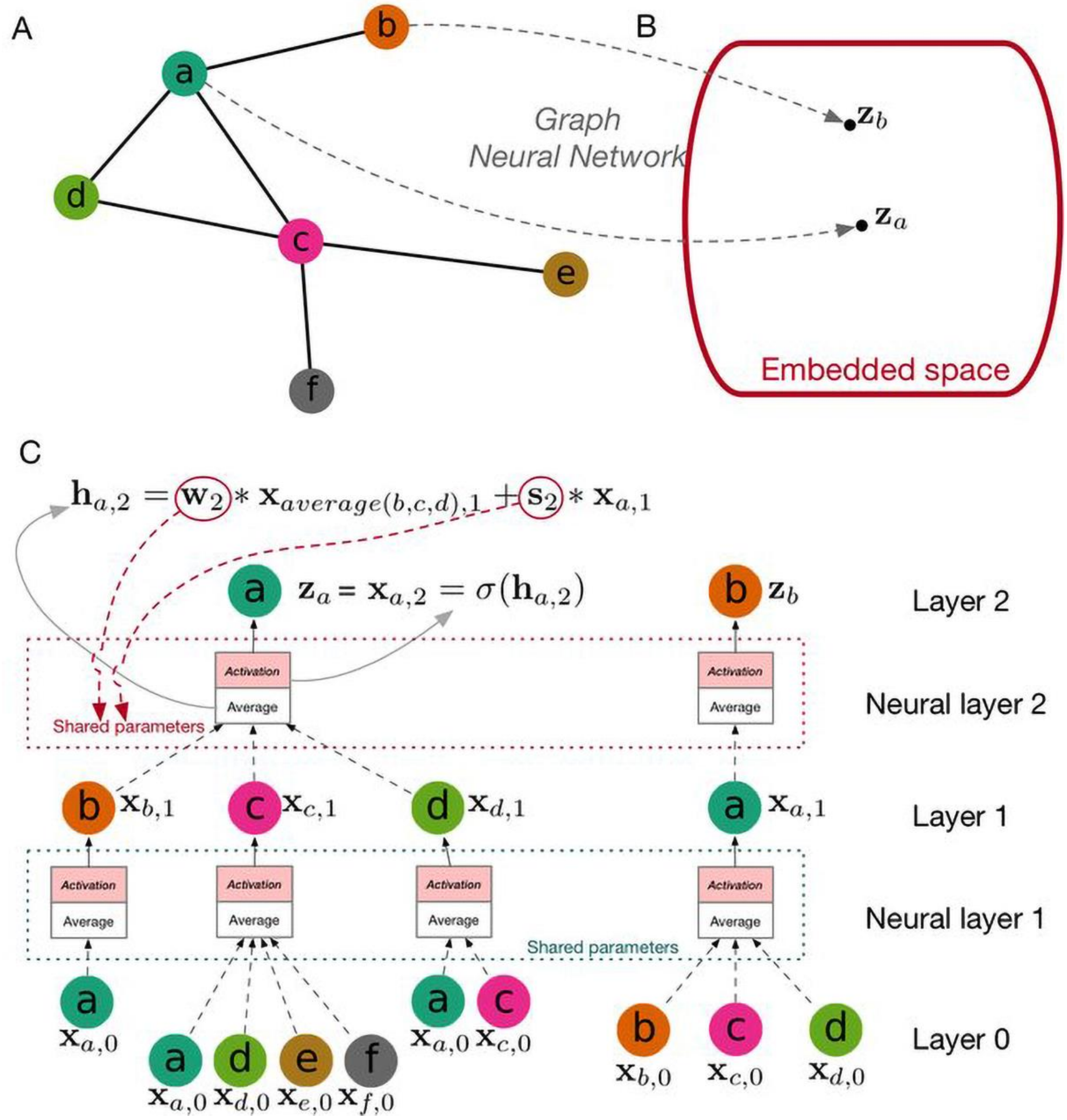

Figure 2: Illustration of a graph neural network[17].

(A) A typical graph data example. The embedding space (B). Each data point in this embedding space is represented by a vector, and the original topological information in (A) is preserved in that vector. (C) A graph neural network for embedding a network in a graph (A). As examples, consider nodes a and b. The original representations are the internal properties of each node. The nodes in each layer aggregate information from their neighbors and update the representations using an averaging and activation function. In this example, the embedding result is the output of layer 2. Notice that the parameters within the same layer between different trees are shared so this method can be generalized to the previously unseen graph of the same type.

4 Result and Discussion

In Figure 3 it is demonstrated that the qualitative dataset provides the significance of trained data using Graphical neural networks. The study provides the analysis of results more efficiently shown in figure 4.

Figure 3: output result shows accuracy rate

Figure 3 shows the accuracy of the object detection and also shows images dangerous/vulnerable or not. Two objects for object detection one is a gun and another one is a knife. The study implies more than five thousand data to train the model and after complete the train, the test phenomenon has been conducted on the proposed model. An accuracy rate was reached with a little bit of validation and training loss.

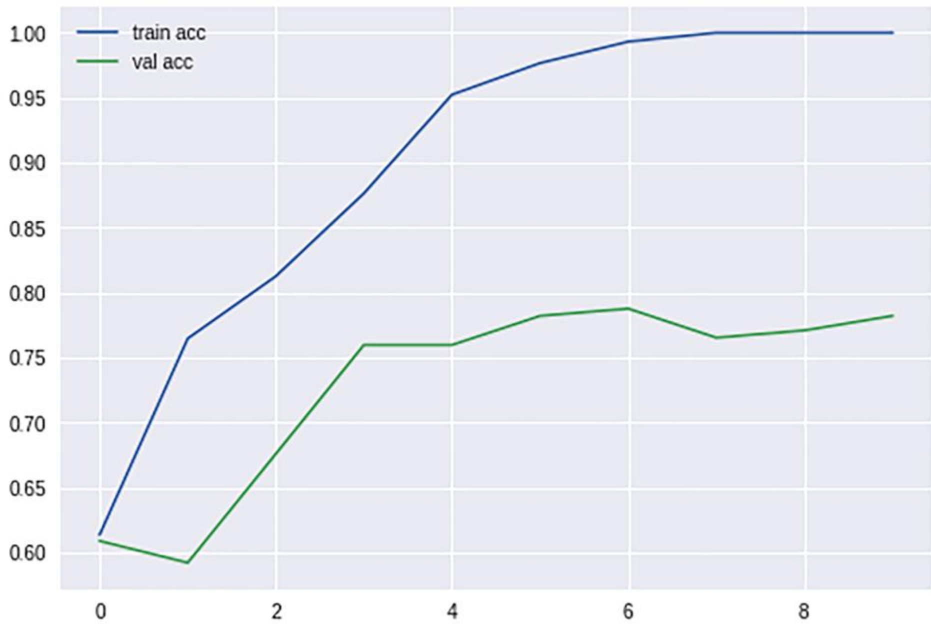

Figure 4: Train and valid accuracy

Figure 4 illustrates the train and valid accuracy results gained from proposed train model. Figure 4, also demonstrates that the training accuracy reached the benchmark ranging from 0.63 to 0.99. Validation accuracy first time 0.61 then increase to 0.76 then continuously increased in this figure. The final accuracy rate for valid is 0.7821.

Figure 5: Train and valid loss

Figure 5 shows that the train and valid loss after train the model is deployed. In this figure 5, it is denoted that the training loss decreased from 12 to 1.50. Validation loss first time 6.28 then decrease to 3.32. This provides detailed information about the result achieved with a validation loss of about 1.50.

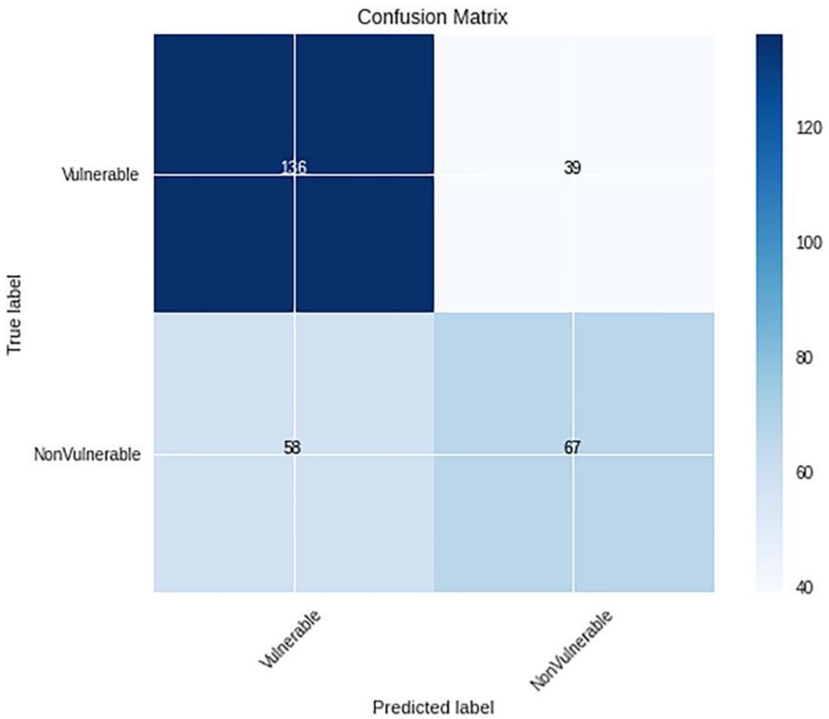

Figure 6: Confusion Matrix

The test data set includes 300 images. After that, the classifier will predict whether or not the images are vulnerable. The actual outcomes contain information about the extracted results that has been projected using confusion matrix in figure 6, whereas the predicted outcomes are the classification model's predictions. The confusion matrix provides the following information:

The model correctly predicted that the vulnerable was present in 136 of the images. The model incorrectly predicted that 39 images lacked the vulnerable. The model correctly classified 67 images as Nonvulnerable. The model incorrectly identified 58 images as vulnerable. The model predicted that the Investigation reached the vulnerable rate presented as 194 (136+58) images. According to proposed model, 106 (39+67) images did not have the vulnerable. In reality, 175 (136+39) images contained the vulnerable, while 125 (58+67) images did not.

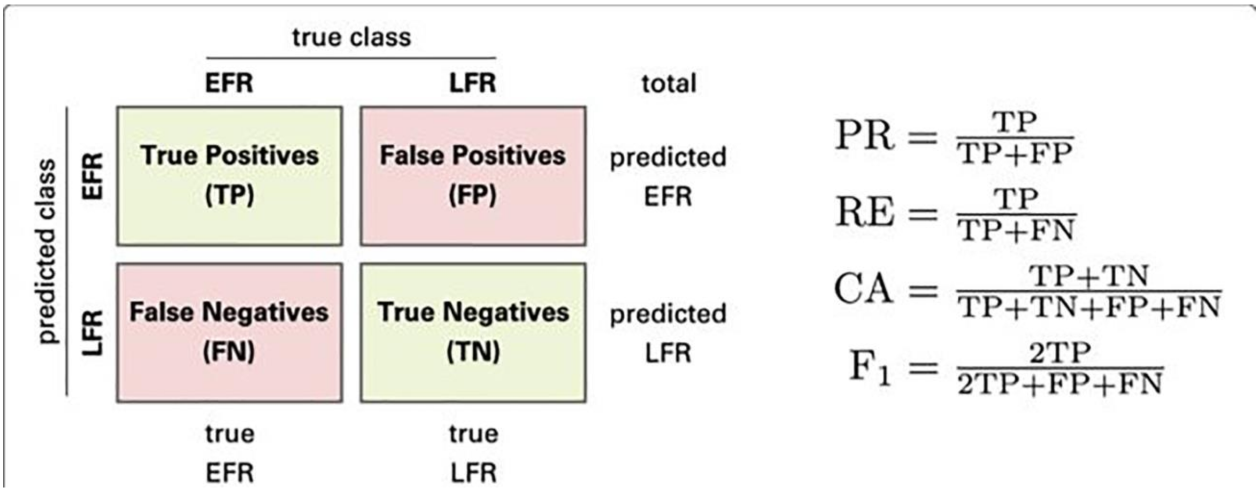

Figure 7: Confusion matrix. Exemplified CM with the formulas of precision (PR), recall (RE), accuracy (CA), and F 1-measure[18].

Now Figure (6) and Figure (7), the detailed statistical calculation is shown. Here, the model accuracy, precision, recall, and f1 score using the mathematical formula. In Figure (6) TP is 136, FP is 39, FN is 58, and TN is 67.

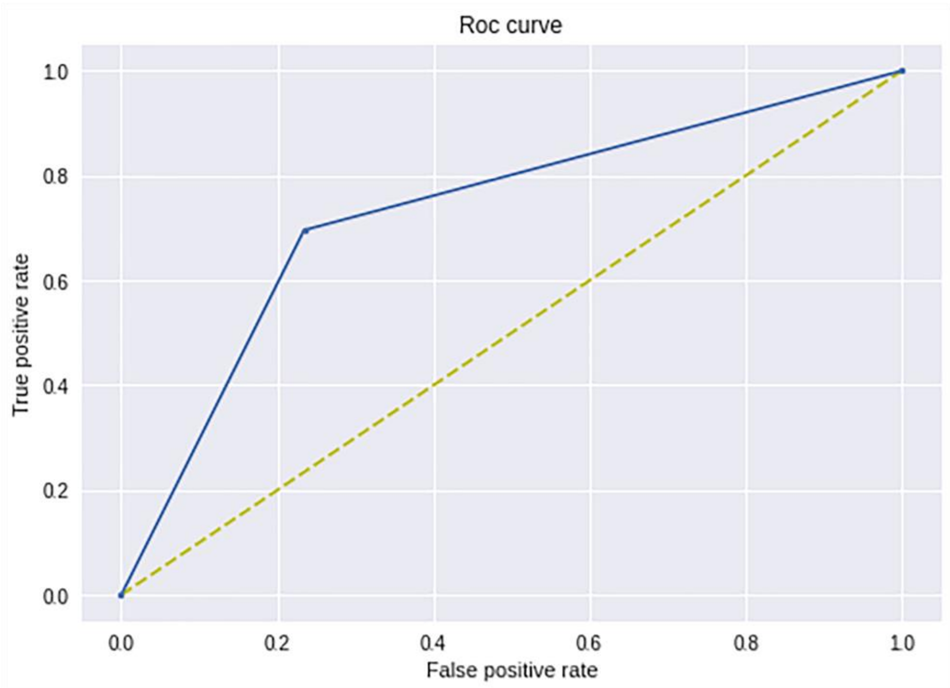

Figure 8: ROC curve

True Positive Rate (TPR) = TP/TP+FN….. (1)

False Positive Rate (FPR) = FP/FP+TN….. (2)

To plot the ROC curve, the TPR and FPR must be calculated for a variety of thresholds. For each threshold, plot the FPR value in the x-axis and the TPR value in the y-axis. Then join the dots with a line.

5 Limitations

The investigation excludes images captured from webcam or videos. The study did not analyzed the use of implementing graphical neural network (GNN) on such images transmitted from digital cameras or videos that might contain blur images.

6 Conclusion

The investigation represents the detection of dangerous objects mostly knives and guns detection method that blends a Normal-map (the surface normal computed from a point cloud) with other hand-crafted images in this paper. The proposed method enhances the input data's spatial shape information. This approach detects objects more accurately. It is less impacted by sparse point clouds than traditional approaches. Furthermore, it improves yaw. Prediction of the angle It also has outstanding anti-interference properties for unstable object surfaces. Data on the intensity of reflections. The technique has the potential to be applied to the virtual world dataset, allowing for more autonomous driving studies. In future, the investigation would endeavor to incorporate a new normal estimation technique into process to improve Normal-map accuracy. This can also incorporate a strategical decision to expand it so that more object classes may be recognized and done faster.

References

[1] M. Adnan, S. Kalra, and H. R. Tizhoosh, "Representation learning of histopathology images using graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 988-989.

[2] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi, "Graph neural networks with convolutional arma filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2021.

[3] N. K. Bose and P. Liang, *Neural* network fundamentals with *graphs,* algorithms*, and applications*. McGraw-Hill, Inc., 1996.

[4] T. Cai, S. Luo, K. Xu, D. He, T.-y. Liu, and L. Wang, "Graphnorm: A principled approach to accelerating graph neural network training," in *International Conference on Machine Learning*, 2021: PMLR, pp. 1204-1215.

[5] D. Cao *et al.*, "Spectral temporal graph neural network for multivariate time-series forecasting," *arXiv preprint arXiv:2103.07719,* 2021.

[6] F. Errica, M. Podda, D. Bacciu, and A. Micheli, "A fair comparison of graph neural networks for graph classification," *arXiv preprint arXiv:1912.09893,* 2019.

[7] F. F. Wu, Y. Ni, and P. Wei, "Power transfer allocation for open access using graph theory-fundamentals and applications in systems without loopflow," *IEEE transactions on power systems,* vol. 15, no. 3, pp. 923-929, 2000.

[8] W. Fan *et al.*, "Graph neural networks for social recommendation," in *The World Wide Web Conference*, 2019, pp. 417-426.

[9] I. Fukunaga, R. Sawada, T. Shibata, K. Kaitoh, Y. Sakai, and Y. Yamanishi, "Prediction of the Health Effects of Food Peptides and Elucidation of the Mode-of-action Using Multi-task Graph Convolutional Neural Network," *Molecular informatics,* vol. 39, no. 1-2, p. 1900134, 2020.

[10] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," *arXiv preprint arXiv:1711.04043,* 2017.

[11] L. Gong and Q. Cheng, "Exploiting edge features for graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9211-9219.

[12] D. Grattarola and C. Alippi, "Graph neural networks in tensorflow and keras with spektral," *arXiv preprint arXiv:2006.12138,* 2020.

[13] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra, "Beyond homophily in graph neural networks: Current limitations and effective designs," *arXiv preprint arXiv:2006.11468,* 2020.

[14] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open,* vol. 1, pp. 57-81, 2020.

[15] Q. Zhao, Z. Ye, C. Chen, and Y. Wang, "Persistence enhanced graph neural network," in *International Conference on Artificial Intelligence and Statistics*, 2020: PMLR, pp. 2896-2906.

[16] M. Yang, Y. Shen, H. Qi, and B. Yin, "Soft-mask: Adaptive Substructure Extractions for Graph Neural Networks," in *Proceedings of the Web Conference 2021*, 2021, pp. 2058-2068.

[17] Y. Li, C. Huang, L. Ding, Z. Li, Y. Pan, and X. Gao, "Deep learning in bioinformatics: Introduction, application, and perspective in the big data era," *Methods,* vol. 166, pp. 4-21, 2019.

[18] S. Bittrich, M. Kaden, C. Leberecht, F. Kaiser, T. Villmann, and D. Labudde, "Application of an interpretable classification model on early folding residues during protein folding," *BioData mining,* vol. 12, no. 1, pp. 1-16, 2019.