International Journal on Cybernetics & Informatics (IJCI) Vol.13, No.5, October 2024

Multi-Classification of CAD Entities: Leveraging the Entity-as-Node Approach with Graph Neural Networks

Sheela Raju Kurupathi[1], Park Dongryul[1], Sebastian Bosse[1], and Peter Eisert[1,2]

[1]Fraunhofer Heinrich Hertz Institute (HHI)
[2]Humboldt University of Berlin

Abstract. The construction industry faces challenges in extracting and interpreting semantic information from CAD floor plans and related data. Graph Neural Networks (GNNs) have emerged as a potential solution, preserving the structural integrity of CAD drawings without rasterization. Accurate identification of structural symbols, such as walls, doors, and windows, is vital for generalizing floor plans. This paper investigates GNN methods to enhance the classification of these symbols in CAD floor plans, proposing an entity-as-node graph representation. We evaluate various preprocessing strategies and GNN architectures, including Graph Attention Networks (GAT), GATv2, Generalized Aggregation Networks (GANet), Principal Neighborhood Aggregation (PNA), and Unified Message Passing (UniMP) on the CubiCasa5K dataset. Our results show that these methods significantly outperform current state-of-the-art approaches, demonstrating their effectiveness in CAD floor plan entity classification.

Keywords: BIM, CAD, Floor Plans, GNN, Entity-as-Node, Multi-Classification.

1 Introduction

Floor plans are fundamental in architecture and civil engineering, providing essential structural and geometric details of buildings. They offer a comprehensive view of dimensions, spatial organization, and the arrangement of various elements within a single model. Computer-aided design (CAD) plans [1] are particularly valuable because they represent vertical sections derived from horizontal planes within a building. CAD drawings are crucial throughout the planning, construction, and maintenance phases. They present a clear depiction of design elements, facilitate effective communication among stakeholders, and ensure consistency with the design.

Despite their critical role, the manual processing and analysis of CAD drawings are often labor-intensive and fraught with inefficiencies. This limitation highlights a pressing need for more advanced and automated analysis methods. The automatic analysis of CAD drawings presents an opportunity to advance Building Information Modeling (BIM) [2]. BIM represents a significant evolution from traditional CAD by providing a more comprehensive and dynamic representation of buildings, encompassing detailed physical and functional characteristics. This enriched model enhances the accuracy of building representations and supports more effective management throughout the building's lifecycle, from construction to operation and maintenance. Automating the analysis of CAD drawings streamlines the BIM process and integrates data more seamlessly, leading to improved modeling precision and more efficient project management. This automation facilitates the development of more coherent workflows, minimizes errors, and supports informed decision-making. Consequently, it contributes to more successful and sustainable building practices by optimizing resource utilization and enhancing overall project outcomes.

CAD drawings are denoted through vector-based graphics, utilizing geometrical entities such as lines, polygons, arcs, circles, and ellipses to depict various objects and structures.

 DOI:10.5121/ijci.2024.130503

Recent advancements have seen Graph Neural Networks (GNNs) [3] emerge as a powerful tool for leveraging the structural properties inherent in these vector-based representations. GNNs can capture and analyze the relationships between different entities within the floor plan, thus offering a more profound and insightful understanding of the data.

In this study, we propose a novel approach to CAD floorplan analysis. We employ an entity-as-node paradigm within a GNN framework for classifying floorplan elements. Our approach introduces a new methodology for data manipulation utilizing the CubiCasa5K [4] floorplan dataset. This method emphasizes the importance of graph-specific information, focusing on edge-based entity relationships to enhance classification accuracy. Our key contributions include:

- **Efficient Vector Representation:** We developed an advanced vector representation of floorplan elements through intersection-based element splitting, which improved the granularity and detail of the representation.
- **Multi-Class Classification Framework:** We propose a robust multi-class classification framework that leverages GNNs to classify various floorplan elements accurately, thus advancing the state of the art in CAD floorplan analysis.
- **Enhanced Entity Classification:** Our approach improves entity classification by incorporating edge-based relations, providing a more nuanced understanding of the spatial and functional relationships between elements.

These advancements push the boundaries of CAD floorplan analysis and classification, contributing to more precise and efficient methods in the construction domain. By integrating these innovations, our study aims to set a new standard for automated analysis in architectural and engineering applications, driving theoretical and practical advancements in the field.

1.1 Related work

Automating the analysis of CAD floor plans is a complex task. It involves two primary methodologies: pixel-based and vector-based approaches. Pixel-based methods rely on rasterized images, which convert floor plans into grids of pixels. This transformation poses challenges in capturing precise geometric details due to the loss of structural clarity. On the other hand, vector-based methods leverage organized geometric and semantic information, allowing for more sophisticated computational processing and editing. This structured representation facilitates a more accurate and detailed analysis of CAD floor plans, preserving the integrity of the original design elements.

Graph Neural Networks (GNNs) have emerged as a powerful tool for analyzing CAD floor plans, offering a promising alternative to traditional methods. For example, Hu et al. [5] introduced Graph2Plan, a cutting-edge framework that integrates GNNs with Convolutional Neural Networks (CNNs). This innovative approach enhances floor plan analysis by combining the spatial features extracted by CNNs with a graph-based representation created by GNNs. Similarly, Paudel et al. [6] represented floor plans as undirected graphs, where rooms are nodes and adjacency relationships are edges. This method effectively captures the relational structure of floor plans, facilitating advanced tasks such as room segmentation and classification. Chen et al. [7] further advanced this field by proposing novel graph-based techniques to capture complex layout features, thereby improving classification accuracy and overall analysis performance.

Ahmed et al. [8] developed a method focused on automatic room detection and labeling within architectural floor plans. Their approach includes preprocessing steps for enhancing image quality and feature extraction techniques to differentiate between various room types and labels. This methodology significantly improves the efficiency and accuracy of room classification, which is crucial for automating large-scale floor plan datasets. Zhang et al. [9] explored a GNN-based approach for classifying architectural floor plans, using graph representations to capture intricate relationships within the data. Wang et al. [10] utilized hierarchical GNNs to capture multi-level relationships within floor plans, enhancing classification accuracy by integrating both local and global structural information.

Our research presents a novel approach by employing an entity-as-node methodology [11], diverging from the previously explored edge-oriented methods. This innovative approach improves the clarity and efficiency of floor plan representation, leading to superior performance in node classification tasks. By representing each entity in the floor plan as a distinct node in the graph, we enhance the precision of the analysis and address limitations found in prior GNN studies. We integrate this entity-as-node approach with comprehensive data preprocessing to refine CAD floor plan analysis.

We rigorously evaluate various GNN architectures for multi-label entity classification on floor plan graph nodes. These architectures include Graph Attention Networks (GAT) [12], GATv2 [13], Generalized Aggregation Networks (GANet) [14], Principal Neighborhood Aggregation (PNA) [15], and Unified Message Passing (UniMP) [16]. PNA enhances node representation by employing diverse aggregation strategies, while UniMP offers a unified framework for message passing in graphs. Our experiments reveal that UniMP [16] achieves the highest performance, demonstrating the exceptional effectiveness of our proposed approach in classifying entities within CAD floor plans. This work highlights the potential of advanced GNN methodologies in transforming the analysis and interpretation of complex architectural designs.

2 Proposed Approaches

The workflow and experimental pipeline of our approach are illustrated in Figure 1. The process initiates with the preprocessing of CAD floor plan elements, initially provided in SVG format. This preprocessing stage encompasses several critical steps: annotation, subdivision, and normalization of the elements. Additionally, some methods include splitting or merging elements based on intersection points or start-end coordinates to enhance the granularity of the representation. After preprocessing, the floor plan elements are represented as nodes within a graph. The connections between these nodes are established based on a predefined strategy, creating a comprehensive graph structure. Each node in the graph has features, including identifiers, class labels, fill and stroke attributes, style, opacity values, and stroke width, that provide a rich representation of the floor plan data. The graph features are then input into five distinct Graph Neural Networks (GNNs) for the task of node classification. Each GNN processes the graph to predict node labels, which may correspond to categories such as *Wall*, *Window*, *Stairs*, *Railing*, and *Misc*. The output semantic graph is labeled using these predicted labels. The final phase of the process involves visualizing the semantic floor plan according to the predicted labels. This visualization serves as a tool to interpret the effectiveness of the classification, providing insights into the accuracy and reliability of the GNN models. The results are meticulously analyzed to draw conclusions and evaluate the performance of the employed GNN architectures, thereby assessing their efficacy in the CAD floor plan analysis.

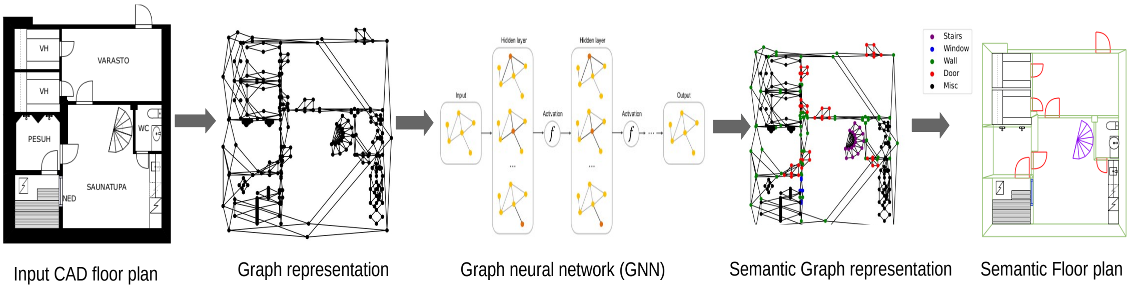


Fig. 1. Workflow of GNN-based CAD entity classification with the proposed approach of element intersection splits and leveraging of edge-based entity relations.

2.1 Cubicasa5K Dataset

For this study, we employed the CubiCasa5K dataset [4], which comprises 5000-floor plan images in SVG format. Each image in this dataset provides comprehensive shape attributes, including coordinates, color, and line thickness [17]. Sample images from the CubiCasa5K dataset are depicted in Figure 2.

The dataset is categorized into three subsets: 4200 floor plans for training, 400 for validation, and 400 for testing. The preprocessing pipeline follows the CubiCasa5K methodology [4], encompassing several critical steps:

– **Annotation:** Each SVG element is annotated to identify its class and relevant features.
– **Subdivision:** The elements are subdivided to improve resolution and detail for analysis.
– **Normalization:** Attributes such as coordinates, color, and thickness are normalized to ensure consistency across the dataset.

Upon completion of the preprocessing steps, the SVG elements are transformed into graph nodes. The connections between these nodes are established based on a predefined strategy, resulting in a structured graph representation of the floor plans. This structured graph is the foundation for subsequent analysis, enabling a detailed examination of the floor plan data.

Fig. 2. Sample images from Cubicasa 5k dataset. The images from left to right are original floorplan image and the SVG label. [4]

2.2 Approaches for CAD Entity Classification

This study evaluates eight distinct methods for classifying CAD floorplan entities using Graph Neural Networks (GNNs). The baseline method, "ENNet" (Entity-as-Node Network), represents each CAD floorplan element as an individual node in the GNN. This method serves as a state-of-the-art reference.

Seven additional approaches build on the ENNet framework by incorporating three distinct augmentation features, with suffixes indicating the specific features utilized:

- ***d*** **:** Utilization of edge features, which capture distance information between nodes.
- ***c*** **:** Establishment of node connections based on intersections between elements.
- ***s*** **:** Implementation of element sub-splitting based on crossing points.

For example, ENNet-cd integrates element intersection information for node connections and incorporates edge features into the graph representation but does not use sub-splitting of elements based on crossings.

As detailed in Table 1, the eight approaches range from ENNet to ENNet-scd, each representing a unique combination of these three experimental factors: element intersection splits, node connection methods, and edge feature utilization.

Approach	Intersection split	Node connection	Distance edge feature
ENNet	X	Start-end point	X
ENNet-d	X	Start-end point	O
ENNet-c	X	Intersection	X
ENNet-cd	X	Intersection	O
ENNet-s	O	Start-end point	X
ENNet-sd	O	Start-end point	O
ENNet-sc	O	Intersection	X
ENNet-scd	O	Intersection	O

Table 1. The state-of-the-art ENNet approach and 7 proposed approaches listed with differentiation based on three separate experimental features.

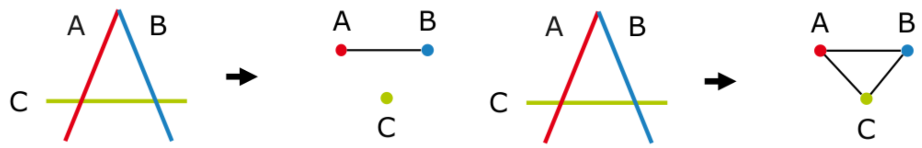
2.3 Methods for CAD Floorplan to Graph Conversion

We have developed three methods for forming links between entity nodes and specifying edge features at the graph level.

Start-End point-based connection The start-end point-based connection method is designed to form connections between entity nodes in a graph based on their start and end coordinates. Each CAD floor plan element is an individual node within the graph. The core principle is establishing edges between nodes if their start or end points are nearby or exactly match. Specifically, an edge is created between two nodes if either the start or end point of one node coincides with start or end point of another node. This method ensures the graph accurately reflects the spatial relationships between different entities in the floor plan. A rounding technique handles minor deviations in coordinate values and to ensure precise node connections. This technique adjusts the coordinates to a standard precision, reducing the impact of minute measurement inaccuracies that could otherwise lead to erroneous or missed connections.

For instance, consider the scenario illustrated in Figure 3 (a). Here, nodes A, B, and C represent separate entities within the graph. The entities A and B are connected by an edge because they share a common coordinate point, either as a start or end point, indicating a direct spatial relationship between them. This connection captures the relational information between nodes A and B, enhancing the overall connectivity and contextual understanding of the floor plan. Conversely, entity C does not share any start or end points with the other entities and, therefore, does not form any connections. As a result, entity C remains isolated within the graph, highlighting the importance of shared points in establishing meaningful connections between nodes.

This method is useful for creating a comprehensive and accurate graph representation of floor plans, as it effectively captures spatial relationships crucial for subsequent analysis and classification tasks. By focusing on the start and end points, the approach ensures that the graph accurately mirrors the layout and connectivity of elements within the CAD floor plan.



(a) Start-End point-based node connection.

(b) Intersection-based node connection.

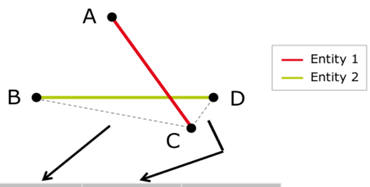
Fig. 3. Two different approaches to creating node connections.

Intersection-based connection In Figure 3 (b), the layout of independent graph entities A, B, and C remains consistent with Figure 3 (a), but the method used to define node connections has been altered. The intersection-based approach identifies and incorporates crossings between entities, offering a more nuanced method for establishing node connections. Unlike the start-end point-based method, which previously failed to recognize any relational connection between entity C and the other entities despite their spatial proximity, the intersection-based method significantly enhances adjacency detection by focusing on intersection points. This approach utilizes information diffusion mechanisms [18] within Graph Neural Networks (GNNs). In this context, node features are propagated and aggregated across multiple layers, allowing the network to effectively capture and utilize the relational information inherent in the graph structure. The intersection-based method employs a sweep line algorithm to detect segment intersections within the floor plan. This algorithm systematically processes the floor plan elements to identify points where different segments cross each other.

A rounding method is applied to smooth out minor coordinate deviations, similar to the technique used in the start-end point-based approach. This preprocessing step helps to mitigate the effects of minute measurement inaccuracies that could otherwise disrupt the identification of intersections. By detecting and incorporating segment intersections, the intersection-based approach enhances the ability of the GNN to recognize and model spatial relationships between entities. This improved adjacency detection allows for a more comprehensive and accurate representation of the floor plan. It contributes to better performance in subsequent analysis and classification tasks.

Distance edge feature This research investigates whether incorporating graph edge information, in addition to basic node features, improves model performance in CAD entity classification tasks. The focus is on evaluating whether edge features enhance the classification accuracy of methods such as ENNet-d, ENNet-cd, ENNet-sd, and ENNet-scd. These methods utilize edge features to capture the relationships between connected nodes in the graph. One crucial edge feature examined is cross-distance information, which quantifies the numerical distances between linked entities. Cross-distance information is valuable due to its direct representation of spatial relationships within the CAD floor plans. We use Euclidean distance as the edge feature to measure these spatial relationships. However, defining these edge features presents challenges due to the arbitrary nature of CAD floor plan elements and their varying orientations.

To address these challenges, we prioritize the features that are more indicative of the spatial relationships between entities. For instance, instead of using explicit start and end point positions—such as distances between start points or endpoints—features reflecting overall spatial relationships, like maximum, minimum, and mean distances, were selected. These features offer a more consistent and interpretable representation of the distances between nodes. Figure 4 illustrates these chosen features and their implementation, highlighting their effectiveness in capturing the independent relationships between connected entities. By evaluating these edge features, this research aims to determine their impact on enhancing the performance of graph-based models in classifying CAD entities. It contributes to more accurate and reliable floor plan analysis.



max_dist	min_dist	mean_dist
10.12	2.94	7.01

Fig. 4. 3-dimensional edge feature with distance information between entities. Distance values in the feature matrix are chosen arbitrarily.

3 Experiments and Quantitative Results

3.1 Balanced Accuracy Metric

Given the label imbalance in the dataset, this study employs balanced accuracy [19] to evaluate the performance of the multi-label classification model. Balanced accuracy is selected to address the bias when a model trained on an imbalanced dataset. Balanced accuracy provides the counts of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for each class. For a given class i, Sensitivity is calculated as:

$$\text{Sensitivity}_i = \frac{TP_i}{TP_i + FN_i} \tag{1}$$

where TP_i denotes the true positives for class i and FN_i represents the false negatives for class i. Sensitivity, also known as the true positive rate, measures the proportion of actual positives correctly identified by the model for class i.

To compute the overall balanced accuracy across all classes, we average the sensitivities of each class. The formula for balanced accuracy is:

$$\text{Balanced Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FN_i} \tag{2}$$

where N is the total number of classes. This metric calculates the mean sensitivity across all classes, ensuring that each class contributes equally to the evaluation.

Additionally, balanced accuracy can be seen as a measure that combines the true positive rate with the true negative rate. The true negative rate, or specificity, for class i is:

$$\text{Specificity}_i = \frac{TN_i}{TN_i + FP_i} \tag{3}$$

where TN_i denotes the true negatives for class i and FP_i represents the false positives for class i. While balanced accuracy primarily focuses on sensitivity, it is valuable to consider specificity to understand the model performance comprehensively.

By averaging the true positive rates across all classes, balanced accuracy offers a more accurate reflection of model performance. It ensures that each class performance is equally weighted, thus providing a balanced view of the model's ability to classify all classes correctly.s

3.2 Quantitative Results

Table 2 presents a summary of the experimental results across five different networks for CAD floor plan entity classification, evaluating eight distinct approaches. Balanced accuracy scores were obtained after training each model for 100 epochs, utilizing 50 hidden units and 50 hidden layers.

Approach /Network	GAT	GATv2	GANet	PNA	UniMP
ENNet	0.740	0.737	0.638	0.615	0.750
ENNet-d	0.733	0.751	0.697	0.683	0.672
ENNet-c	0.946	0.948	0.812	0.896	0.984
ENNet-cd	0.957	0.930	**0.981**	**0.987**	0.919
ENNet-s	0.921	0.978	0.937	0.778	0.960
ENNet-sd	0.944	**0.983**	**0.981**	0.971	0.832
ENNet-sc	**0.964**	0.947	0.936	0.850	**1.000**
ENNet-scd	0.948	0.969	0.912	0.974	0.848

Table 2. Comparison of balanced accuracy scores for 5 different GNNs tested on the CubiCasa5K dataset [4]. 8 different approaches are evaluated based on 3 separate experimental factors as seen in Table 1. For each network, the highest score reached from all approaches is highlighted in bold.

Among the evaluated methods, *ENNet-sc* achieved the highest performance, attaining a balanced accuracy score of 1.00 when paired with the UniMP network on the test set. This result significantly surpasses the baseline *ENNet* approach, which achieved a maximum balanced accuracy of 0.75 with the same network. The performance of both *ENNet* and *ENNet-d* approaches was comparatively lower, with balanced accuracy scores ranging from 0.615 to 0.751. Conversely, the approaches from *ENNet-c* through *ENNet-scd* generally exhibited higher balanced accuracy scores, with most methods surpassing 0.9. This indicates a clear distinction between the lower-performing *ENNet* and *ENNet-d* methods and the higher-performing *ENNet-c* to *ENNet-scd* methods, with no significant variation observed within each group.

These results underscore the effectiveness of incorporating element intersection information in enhancing classification performance. However, the influence of distance information on graph edges exhibits variability depending on the network architecture employed. The observed performance differences suggest that the impact of distance information can be either beneficial or detrimental to classification outcomes. This variability may arise from the common properties shared among various elements, such as walls and railings, which frequently connect and interact, influencing the classification results.

4 Conclusion and Limitations

This work presents innovative methods to enhance Graph Neural Network (GNN) based multi-label classification of floor plan entities by refining the input dataset. Significant improvements include the removal of duplicate CAD elements and the segmentation of elements based on their intersections. Experimental results, derived from eight distinct approaches and evaluated across five different GNN architectures, demonstrate that integrating element intersection information enhances classification performance, surpassing previous state-of-the-art methods. It underscores the crucial role of preprocessing in optimizing graph structure and leveraging the full potential of GNNs.

Despite these advances, several challenges remain. Label noise emerges from overlapping elements of different types, complicating accurate classification and potentially degrading model performance. Additionally, the sweep line technique [20], employed for element segmentation, has limitations as it primarily accommodates straight lines, which may not capture all relevant intersections. Fully connected nodes also prove to be computationally intensive, posing scalability issues.

5 Future Research Work

Future work should focus on several key areas to address these limitations and further advance the field. Firstly, exploring alternative methods for node connections could lead to more accurate and efficient graph representations. Techniques such as adaptive node connection algorithms or distance-based similarity measures may offer improvements. Secondly, investigating the incorporation of additional node features is crucial. Features that capture more detailed spatial and semantic information about the elements could enhance the model's ability to distinguish between entities. It might include integrating contextual information or hierarchical relationships between entities. Thirdly, developing more effective edge features could improve classification accuracy and computational efficiency. Research into dynamic edge features that adapt based on the context or interaction patterns between elements might yield promising results.

Additionally, addressing label noise through advanced data cleaning and augmentation techniques could help mitigate inaccuracies caused by overlapping elements. To improve the model's robustness, we can explore techniques like semi-supervised learning or noise-robust loss functions. Finally, expanding the experimental scope to include large and more diverse datasets could validate the generalizability of the proposed methods and assess their effectiveness in various real-world scenarios. By addressing these areas, future research will overcome present challenges and drive further advancements in floor plan entity classification.

Acknowledgements

This research is funded by BMWK in the second call for funding of the "Artificial Intelligence Innovation Competition" as part of the BIMKIT project (grant number 01MK21001H).

References

1. Spangler, S. C. (2024). A/E/C Computer-Aided Design (CAD) Standard: Release 6.2.
2. M. Urbieta et al. 2023, "Generating BIM model from structural and architectural plans using Artificial Intelligence" Journal of Building Engineering 2023. DOI: https://doi.org/10.1016/j.jobe.2023.107672.
3. S. K. Maurya, X. Liu, and T. Murata, "Simplifying approach to node classification in graph neural networks", arXiv preprint arXiv:2111.06748, 2021.
4. A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, and J. Kannala, Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis, 2019. arXiv: 1904.01920 [cs.CV].
5. R. Hu, Z. Huang, Y. Tang, O. V. Kaick, H. Zhang, and H. Huang, "Graph2plan", ACM Transactions on Graphics, vol. 39, no. 4, Aug. 2020. DOI: 10.1145/3386569.3392391.
6. A. Paudel, R. Dhakal, and S. Bhattarai, Room classification on floor plan graphs using graph neural networks, 2021. arXiv: 2108.05947 [cs.LG].
7. Chen, H., et al. "Graph Neural Networks for Building Layout Analysis and Classification", Proceedings of the European Conference on Computer Vision (ECCV), 2020.
8. S. Ahmed, M. Liwicki, M. Weber, and A. Dengel. "Automatic room detection and room labeling from architectural floor plans", In 2012 10th IAPR international workshop on document analysis systems. IEEE, 339–343.
9. Zhang, Y., et al. "A Graph Neural Network Approach to Classifying Architectural Floor Plans", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
10. Wang, T., et al. "Hierarchical Graph Neural Networks for Floor Plan Classification", IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 2022.
11. C. Simonsen, F. Thiesson, M. Philipsen, and T. Moeslund, "Generalizing floor plans using graph neural networks", Dansk, in 2021 IEEE International Conference on Image Processing (ICIP), USA: IEEE, 2021, pp. 654–658, ISBN: :978-1-6654-3102-6
12. P. Veli˘ckovi´c, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, Graph attention networks, 2018. arXiv: 1710.10903 [stat.ML].
13. S. Brody, U. Alon, and E. Yahav, How attentive are graph attention networks?, 2022. arXiv: 2105.14491 [cs.LG].
14. G. Li, C. Xiong, A. Thabet, and B. Ghanem, Deepergcn: All you need to train deeper gcns, 2020. arXiv: 2006.07739 [cs.LG].
15. G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veli˘ckovi´c, "Principal neighbourhood aggrega- tion for graph nets", in Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 13 260–13 271.
16. Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, Masked label prediction: Unified message passing model for semi-supervised classification, 2021. arXiv: 2009.03509 [cs.LG].
17. A. Watt, SVG Unleashed (Unleashed Series). Sams, 2002, ISBN: 9780672324291. [Online]. Available: https://books.google.de/books?id=Sgqqm6yDhigC.
18. F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model", IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61–80, 2009. DOI: 10.1109/TNN.2008.2005605.
19. K. H. Brodersen, C. S. Ong, and Stephan, "The balanced accuracy and its posterior distribution", in 2010 20th International Conference on Pattern Recognition, 2010, pp. 3121–3124. DOI: 10.1109/ICPR.2010.764.
20. J. L. Bentley and T. Ottmann, "Algorithms for reporting and counting geometric intersections", IEEE Transactions on Computers, vol. C-28, pp. 643–647, 1979.

Authors

Sheela Raju Kurupathi received her Master's degree in Computer Science with a specialization in Intelligent Systems from the Technical University of Kaiserslautern, Germany. Currently, she is employed at Fraunhofer, where she focuses on Artificial Intelligence and Machine Learning applications in Laser and Diamond technologies. Her research interests encompass the development and application of Machine Learning and AI-based systems.

Park Dongryul received his Master's degree from Humboldt University of Berlin. His research interests focus on the development of computer vision, human-machine interaction, and AI-based systems.

Sebastian Bosse received his Dr.-Ing. degree in Computer Science from the Technical University of Berlin in 2018. He is currently the head of the Interactive Cognitive Systems group at the Fraunhofer Heinrich Hertz Institute (HHI) in Berlin, Germany. His research focuses on computational perception and cognition, machine learning, computer vision, and human-machine interaction.

Peter Eisert received his Dr.-Ing. degree from the University of Erlangen-Nuremberg, Germany, in 2000. He is currently a Professor of Visual Computing at Humboldt University of Berlin and heads the Vision Imaging Technologies Department at the Fraunhofer Institute for Telecommunications – Heinrich Hertz Institute in Berlin, Germany. His research interests encompass 3D image and video analysis and synthesis, face and body processing, machine learning, computer vision, and computer graphics.