

Detecting Spatial Ordering of Nanoparticles with Geometric Deep Learning

Jan Krupiński
0009-0001-0267-7387

Cracow University of Technology
Faculty of Electrical and Computer Engineering
ul. Warszawska 24, 31-155 Kraków, Poland
Email: jan.krupinski@pk.edu.pl

Kazimierz Kielkowicz
0000-0001-5791-6069

Cracow University of Technology
Faculty of Electrical and Computer Engineering
ul. Warszawska 24, 31-155 Kraków, Poland
Email: kazimierz.kielkowicz@pk.edu.pl

Abstract—Nanoparticle dispersion in heterogeneous catalysts plays a critical role in catalytic performance. We propose a robust and generalizable graph neural network (GNN) approach that combines the edge convolutional operator (EdgeConv) with a graph attention (GAT) layer to classify dispersion patterns in palladium on carbon (Pd/C) catalysts. Our method leverages GNNs to operate directly on particle location data extracted from scanning electron microscopy (SEM) images, thereby avoiding reliance on image features that may introduce bias. The proposed method offers an advantage over traditional image-based approaches, which risk overfitting to visual characteristics of the image that are unrelated to the spatial distribution of the nanoparticles. We validate the performance of our GNN architecture on multiple Pd/C samples with distinct carbon support types, achieving an accuracy of 89.84%, and demonstrate that our approach can reliably identify dispersion defects. The results highlight the potential of GNNs as a promising alternative for structure-based analysis and quality assessment of nanomaterial-based catalysts.

Index Terms—Heterogeneous Catalysts, Graph Neural Networks, Deep Learning, Scanning Electron Microscopy

I. INTRODUCTION

NANOMATERIAL-BASED catalysts are a major class of heterogeneous catalysts, widely used in chemistry, industry and medicine [1], [2]. They primarily consist of metal nanoparticles dispersed on a solid support, forming active sites that facilitate organic chemical reactions. Differentiated by the type of metal or metal alloy nanoparticle, as well as by the nature of the support material (ranging from carbon to various oxides such as silica), they exhibit tunable catalytic properties that can be optimized and tailored for specific reactions.

Here, we focus on palladium metal on carbon support (Pd/C) catalysts, which are primarily used in carbon-carbon coupling reactions (C-C coupling) and hydrogenation processes to efficiently synthesize a wide range of organic compounds [3], [4]. There exists a great variability among Pd/C catalysts, depending on both the metal and the characteristics of activated charcoal support used. Such factors as palladium oxidation, dispersion of the nanoparticles, water content and the support structure play a great role in the reactivity of the catalyst [5]. Optimization of these parameters can lead to improved reaction efficiency, greater selectivity, and enhanced catalyst stability across a great range of synthetic applications.

The dispersion of nanoparticles on the carbon support can be analyzed using scanning electron microscopy (SEM) [6]. The Pd/C morphology may range from a mostly uniform distribution of palladium nanoparticles to the formation of ordered structures arising from imperfections in the support [7]. In such cases, nanoparticles can, for example, nucleate along grain boundaries or pore edges, leading to non-uniform distributions of active sites. These structural irregularities can negatively impact the catalytic efficiency and consistency.

This work proposes a generalizable and robust deep learning approach for detecting nanoparticle dispersion defects in Pd/C catalysts based on a novel graph neural network with edge convolutional operator (EdgeConv) [8] and a graph attention (GAT) layer [9]. We address limitations of existing techniques by leveraging graph neural networks to classify catalysts based on nanoparticle spatial arrangements and geometrical patterns. By working solely with particle location data, we aim to eliminate potential biases introduced during image acquisition or sample preparation. Additionally, we evaluate our model on Pd/C samples with an alternative carbon support type to demonstrate its effectiveness under real-world variations in material composition.

This paper is structured as follows. Section II reviews related research involving deep neural networks. Section III introduces the dataset and data preparation process, followed by a description of the graph neural network (GNN) architectures and the training scheme. In Section IV, we outline the evaluation metrics, present model performance, and compare our approach with alternative methods. Section V concludes our paper and provides a brief discussion.

II. RELATED WORK

A. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have been successfully applied to problems with an underlying grid-like (i.e., Euclidean) data structure, particularly in image processing, speech recognition, classification, and image segmentation [10]. Recent applications of CNNs include bone age evaluation from X-ray images to support diagnosis and treatment planning for various disorders [11]; olive disease classification using an adaptive ensemble of two EfficientNet-B0 models,

which improves state-of-the-art accuracy on a publicly available dataset [12]; and semantic segmentation of complex urban street scenes for autonomous driving, where models such as MobileNet and ResNet50 are used as encoders in the U-Net architecture [13].

In the field of heterogeneous catalysis, CNNs have been applied to nanoparticle segmentation and tracking under reactive conditions [14]. CNN architectures such as U-Net [15], which consists of two paths (a contracting path and an expansive path) and employs a training strategy that heavily relies on data augmentation, have been shown to make more efficient use of limited annotated samples. U-Net has been used to analyze transmission electron microscopy (TEM) images and videos, along with other architectures [16], [17]. Additionally CNNs have been used for automated analysis to identify the number of defects and to define anchoring and segmentation in the study of high-entropy metal nanoparticles [16]. More recent state-of-the-art models, such as Segment Anything Model (SAM) [18], have also been employed to aid in the quantification and analysis of nanoparticles [19].

CNNs have been further used to analyze SEM images of Pd/C catalysts for the purpose of classification of nanoparticle dispersion defects, distinguishing between defective and non-defective morphology [20]. While high classification accuracy ($\geq 90\%$) was reported, the dataset was limited in size, and the models were not evaluated on independent samples. As a result, the models were shown to differentiate between specific sample identities rather than between dispersion patterns themselves. In this work, we aim to overcome these limitations by developing a more generalizable method.

B. Graph Neural Networks

For problems involving data that does not exhibit a regular grid-like structure, such as point clouds or molecular structures, the data can instead be modeled as graphs [21]. Graph Neural Networks (GNNs) represent each data point as a vertex in a graph and construct edges based on neighborhood relationships. Spatial GNNs define message-passing and aggregation operations directly over a node's neighborhood in the input or feature space.

In the message-passing mechanism, each node updates its representation by receiving and aggregating information (or "messages") from its neighboring nodes, often using learnable functions such as multilayer perceptrons (MLPs). This process allows nodes to iteratively encode both local geometric structure and other features from their spatial context. Pooling is often used to coarsen the graph or summarize neighborhood information, followed by task-specific layers - such as fully connected layers for classification or regression tasks [22].

Simonovsky and Komodakis [21] generalized the convolution operation from traditional CNNs on regular grids to arbitrary graphs. Their approach constructs deep neural networks for graph classification by treating each point as a vertex and connecting it to its neighbors via directed edges.

Building on this idea, Wang et al. [8] proposed the Dynamic Graph CNN (DGCNN) architecture, where the graph

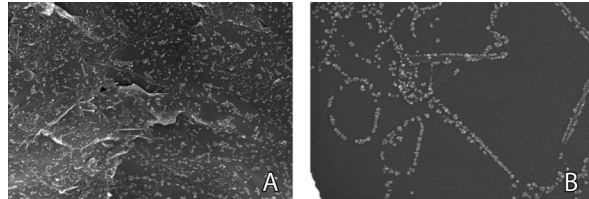


Fig. 1. Image A (on the left) shows an example of a disordered nanoparticle distribution, while Image B (on the right) illustrates nanoparticles forming ordered patterns.

is constructed in the feature space and dynamically updated at each network layer. At its core, DGCNN uses the edge convolutional operator (EdgeConv), which applies an MLP as a feature learning function over edges, followed by an aggregation function that combines information from each point's neighbors.

While GNNs are primarily used for analyzing point clouds [22], they have also found broader application in fields such as molecular modeling and physical system prediction [23]. GNNs have recently been applied to heterogeneous catalysis, including reaction modeling [24] and catalyst screening via binding energy prediction [25]. GNNs have also been applied to graph classification tasks, particularly for predicting the overall toxicity of molecular structures. To improve generalization with limited labeled data, Few-Shot Learning techniques have been incorporated alongside models such as Graph Isomorphism Networks (GINs) and multi-headed attention mechanisms [26].

III. MATERIALS AND METHODS

This section describes and analyzes the dataset, and introduces the proposed graph neural network (GNN) architecture used in this work.

A. Dataset

The dataset used in this study [7] consists of 1000 scanning electron microscopy (SEM) images collected from five different Pd/C catalyst samples. Despite its limited sample size, the dataset encompasses a range of support materials, imaging magnifications, and spatial sampling regions, aiming to provide representative variability across Pd/C catalysts. The dataset includes three types of carbon supports: graphite powder, nanoglobular carbon, and pressed graphite bars. Each image was labeled as either containing ordered or disordered nanoparticle distributions. An example of both distributions is presented on Figure 1. A summary of the images per sample in the dataset is shown in Table I.

TABLE I
OVERVIEW OF THE Pd/C SAMPLES IN THE DATASET [7].

Sample	Images	Ordered	Support	Subset
S1	687	Yes	Graphite powder	Train. / Val.
S2	63	Yes	Graphite powder	Testing
S3	185	No	Nanoglobular C	Train. / Val.
S4	50	No	Graphite bars	Testing
S5	15	No	Graphite bars	Testing

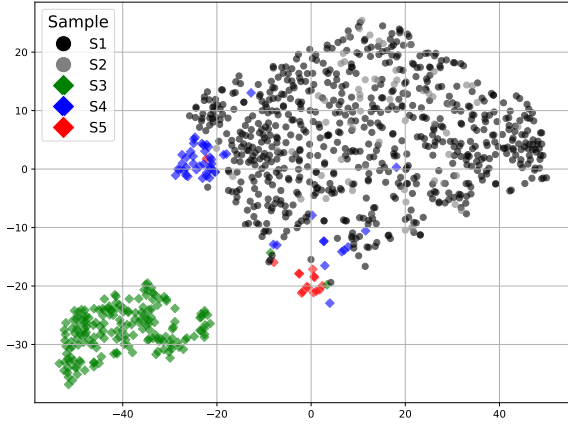


Fig. 2. T-SNE visualization of features extracted by ViT-L from the images in the dataset. Samples marked with dots contain spatially ordered nanoparticles, while samples marked with diamonds are disordered. While most samples are relatively similar visually, S3 is a clear outlier.

To explore visual similarities within the dataset, we extract image features using a deep learning model pretrained on a large-scale image dataset. These features are then projected into a lower-dimensional space using t-distributed stochastic neighbor embedding (t-SNE), a non-linear dimensionality reduction technique commonly used for visualizing high-dimensional data [27]. For this purpose, we use a Vision Transformer (ViT) [28] pretrained using the self-supervised DINOv2 method on LVD-142M [29]. DINOv2 trains vision models without the need for labeled data by encouraging consistency between different augmented views of the same image, enabling the model to learn embeddings that capture semantic and structural information in its latent feature space. This approach is particularly suitable for our task, as it enables meaningful feature extraction without requiring task-specific fine-tuning.

The resulting t-SNE visualization of the dataset is shown in Figure 2. The images are clustered primarily based on their visual characteristics, rather than the spatial arrangement of the nanoparticles. These visual characteristics are primarily determined by the type of sample support (background). At this scale, graphite powder (samples S1 and S2) appears visually similar to graphite bars (samples S4 and S5), while nanoglobular carbon (sample S3) looks distinctly different and forms a separate cluster. This highlights how deep learning models trained on the dataset can be influenced by the visual features introduced by the support material, rather than focusing solely on the spatial distribution of nanoparticles.

B. GNN Architecture

Since our data consists of two-dimensional projections of nanoparticle positions extracted from SEM images, we base our GNN models on architectures commonly used for point cloud analysis. Although the actual particle arrangements are three-dimensional, their projections still encode meaningful spatial structure.

The first step in such models is the construction of a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from the given positions. Here $\mathcal{V} = \{1, \dots, n\}$ denotes the n nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges. We construct a k -nearest neighbors (k-NN) graph to accomplish this task, setting $k = 6$ to balance graph complexity with computational efficiency.

Our base model employs the edge convolutional operator (EdgeConv) [8], which computes edge features for each node $\mathbf{x}_i$ by aggregating information from its neighborhood $\mathcal{N}(i)$:

$$\mathbf{x}'_i = \sum_{j \in \mathcal{N}(i)} h_{\Theta}(\mathbf{x}_i \parallel \mathbf{x}_j - \mathbf{x}_i), \quad (1)$$

Here the $\mathbf{x}$ variables are the 2D spatial coordinates of particles and h_{Θ} is a learnable function implemented as a multilayer perceptron (MLP). Max pooling is used to aggregate information across neighbors. Following the Dynamic Graph CNN (DGCNN) architecture [8], multiple EdgeConv layers can be stacked to learn hierarchical features. However, to avoid problems caused by excessive Laplacian smoothing [30], we use a shallow model with only two EdgeConv layers. As more layers are added, the features of neighboring nodes become increasingly similar, and the representation across the entire graph converges to a single value. This erases important local differences, harming performance. After the EdgeConv layers, the outputs are concatenated and pooled globally before passing through a final MLP for binary classification.

To mitigate the effects of Laplacian smoothing in deeper graph architectures, we also propose a hybrid model architecture where the deeper EdgeConv layer is replaced by a graph attention (GAT) layer [9]:

$$\mathbf{x}'_i = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{i,j} \Theta_t \mathbf{x}_j, \quad (2)$$

In this operator, the transformed features $\Theta_t \mathbf{x}_j$ of neighboring nodes are weighted by attention coefficients $\alpha_{i,j}$, which are learned via an additive attention mechanism. In our architecture, the first EdgeConv layer captures local geometric relationships by operating on an initial k-NN graph. A new k-NN graph is then reconstructed based on the learned features. The subsequent GAT layer models higher-order dependencies on this updated graph by assigning learnable, context-aware weights to neighboring nodes. This dynamic attention mechanism reduces over-smoothing and improves the model's ability to focus on the most informative neighbors. The features are then concatenated and pooled as before. An overview of both models is provided in Table II.

IV. EXPERIMENTAL EVALUATION

A. Data Preprocessing

While both the image resolution (1280×890) and overall dataset size were sufficient for training, validation, and testing of deep learning models, the number of distinct samples was relatively limited. To ensure that the models learn to recognize nanoparticle ordering (rather than relying on sample-specific characteristics or substrate structure) we partitioned the data by

TABLE II
PROPOSED GNN ARCHITECTURES, WITH LAYER SHAPE DESCRIBING THE NUMBER OF NEURONS PER LAYER.

DGCNN Model		EdgeConv + GAT Model	
Layer Type	Layer Shape	Layer Type	Layer Shape
EdgeConv	4, 64, 64, 128	EdgeConv	4, 64, 128, 256
EdgeConv	256, 128, 128, 256	GAT	256, 64 (4 heads)
Concat. + Pooling	-	Concat. + Pooling	-
Global MLP	384, 256, 128	Global MLP	512, 256, 128
Dense (MLP)	128, 64, 2	Dense (MLP)	128, 64, 2

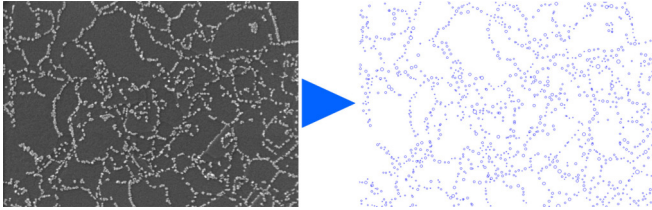


Fig. 3. Example SEM image (left) from sample S1 along with the detected particle locations (right). The nanoparticles form ordered structures.

sample. Samples S1 and S3, which contain the most images, were used for training and validation in an 80/20 split, while the remaining samples (S2, S4, and S5) were reserved for testing. This ensures a more robust assessment of model, and prevents classification based on the sample features alone.

To prepare our data for GNNs, we extracted nanoparticle coordinates from each image. While CNN-based methods have previously been used for nanoparticle segmentation [14], [16], [17], [19], classical methods have been shown to produce acceptable results in nanoparticle detection [31]. We opted for a more computationally efficient classical approach based on the Simple Blob Detection algorithm [32]. This method uses intensity thresholding and contour filtering to identify particles. Contours were filtered by size and brightness to isolate small, bright features corresponding to individual nanoparticles. An example SEM image and the corresponding extracted particle locations are shown in Figure 3.

B. Model Training

The models were trained using the cross-entropy loss function, optimized via the Adam algorithm [33]. A learning rate of 1×10^{-5} was chosen to ensure stable convergence. To prevent overfitting, we employed early stopping, halting training if no improvement was observed in the validation loss for 10 consecutive epochs. Training was performed with a batch size of 64, where each input sample consisted of 512 particle locations, randomly selected from the full set of detected particles in a given image. Batch normalization and dropout were applied throughout the network to improve generalization.

As previously noted, samples S1 and S3 were used for training and validation. The remaining samples were held out to evaluate the model on previously unseen data, ensuring robustness to variations in material and avoiding bias. Nevertheless, the training dataset introduced challenges related to both class imbalance and structural bias. Firstly, the dataset

contained significantly more ordered distributions (687) than disordered ones (185), making it imbalanced. Additionally, sample S3 had a unique, globular structure, not present in the other samples. This raised the risk that the model might learn to associate specific support characteristics with disorder, rather than focusing on the actual nanoparticle arrangement. To address these problems, we applied several data augmentation techniques:

- **Disordered data generation** - additional disordered particle distributions were synthetically generated, to address the class imbalance. Particle positions were initialized on a regular grid and then perturbed by adding noise drawn from a uniform distribution.
- **Geometric transformations** - particle coordinates were flipped horizontally and vertically, rotated.
- **Jittering** - small random perturbations were added to particle positions with a fixed probability.

These augmentation strategies aimed to diversify the training data and reduce overfitting to specific samples or support types, improving the model's ability to generalize to new Pd/C samples.

C. Evaluation Metrics

To assess the performance of our models, we have used typical binary classification metrics such as accuracy, recall, precision and F1 Score (harmonic mean of recall and precision). These metrics can offer complimentary insights into the performance of the model. In our experiments, ordered distributions were treated as the positive class, and disordered distributions as the negative class. In our context, both false positives and false negatives can be problematic, therefore special attention is given to the balance between precision and recall. The results can also be summarized on a confusion matrix.

D. Model Performance

Following training, both architectures described in Table II were evaluated on the held-out test samples: S2, S4, and S5. The results presented in Table III show that both models perform well on new samples and support types, achieving strong performance in classifying both ordered and disordered distributions. The DGCNN model, utilizing only EdgeConv layers, reached an accuracy of 85.16%. Our proposed hybrid model, which combines EdgeConv and GAT layers, outperformed DGCNN with an accuracy of 89.84%. The corresponding confusion matrices are shown in Figure 4, highlighting the

TABLE III
GNN RESULTS ON THE TEST SET.

	DGCNN	EdgeConv + GAT
Accuracy	85.16%	89.84%
Precision	84.38%	90.32%
Recall	85.71%	88.89%
F1 Score	85.04%	89.60%

	DGCNN		EdgeConv + GAT	
Actual \ Predicted	Disordered	Ordered	Disordered	Ordered
Disordered	55	10	59	6
Ordered	9	54	7	56

Fig. 4. Confusion matrices for the GNN results on the test set. The performance for both classes is very similar in both models. Our proposed architecture outperforms DGCNN.

models' balanced performance across both classes, as well as the improvement achieved by our hybrid model.

E. Comparison With CNNs

Recent studies have shown that CNNs can achieve high accuracy when applied to the classification of SEM images of Pd/C catalysts [20]. However, we have raised concerns that CNNs might learn visual sample and support structure characteristics, rather than nanoparticle ordering. To test this claim we have trained two CNN architectures on the dataset: ResNet34 [34] and ConvNeXt [35]. Both models were trained using a similar scheme as described for GNNs, however with starting weights pretrained on the ImageNet [36] dataset with a 224x224 input size.

The results on the training, validation and testing sets are presented in Table IV. Although both CNNs achieved excellent performance on the training and validation data, they failed to generalize to new samples and carbon support types, with test accuracy dropping to 49.22%. As such, their use in real-world scenarios may be limited. CNNs classified all testing images as containing ordered nanoparticle patterns, which can be explained as being due to the visual characteristics of the carbon support in the testing images. Samples S4 and S5 use graphite bars as the support, which in SEM images is more visually similar to graphite powder (ordered samples S1, S2) than to nanoglobular carbon present in the sample S3. This conclusion is based on our previous t-SNE visualization of the dataset, presented on Figure 2.

V. CONCLUSION

In this paper, we propose a novel geometric deep learning approach for classifying dispersion patterns in palladium on carbon (Pd/C) catalysts. Our method is based on graph neural networks (GNNs) and operates directly on particle location

TABLE IV
CNN ACCURACY ON THE TRAINING, VALIDATION AND TESTING IMAGES.

	ResNet34	ConvNeXt
Train.	90.15%	94.22%
Val.	94.53%	98.44%
Test.	49.22%	49.22%

data extracted from scanning electron microscopy (SEM) images. This approach enables classification of catalysts based on the spatial arrangement and geometrical patterns of nanoparticles. As a result, it offers significant advantages over traditional image-based methods, which are prone to overfitting due to irrelevant visual features unrelated to nanoparticle distribution.

First, we present a Dynamic Graph CNN (DGCNN) architecture [8] applied to the classification of dispersion patterns in Pd/C catalysts. Second, in order to mitigate the effects of Laplacian smoothing in deeper graph architectures, we introduce a hybrid deep learning model that incorporates a Graph Attention (GAT) layer stacked on top of a EdgeConv layer. These architectures are compared with standard convolutional neural networks (CNNs), specifically ResNet34 [34] and ConvNeXt [35].

The dataset we used in our study consists of 1000 scanned electron microscopy (SEM) images collected from five different Pd/C catalyst samples [7]. We tested our methods on multiple Pd/C samples with distinct carbon support types to the ones used in training, demonstrating that our proposed methods can reliably detect dispersion defects under real-world variations in material composition.

To assess the performance of the proposed deep learning architectures (DGCNN [8], EdgeConv + GAT, ResNet34 [34], and ConvNeXt [35]), we used standard binary classification metrics, including accuracy, recall, precision, and F1 score. Our results show that the hybrid model combining EdgeConv and GAT layers outperforms the DGCNN, achieving an accuracy of 89.84% compared to 85.16% (see Figure 4 for details). In contrast, the image-based CNNs, ResNet34 and ConvNeXt, both achieved significantly lower accuracy scores of 49.22% (see Table IV) on testing data.

These results demonstrate that both the DGCNN and our proposed EdgeConv + GAT model outperform traditional CNN architectures, with the hybrid model achieving the highest accuracy among all tested methods. Overall, our findings highlight the potential of graph neural networks as a powerful alternative to image-based methods for structure-aware analysis and quality assessment of nanomaterial-based catalysts.

In future work, our GNN architecture could be trained and evaluated on a more diverse set of catalyst types. Depending on the characteristics of the SEM images, this may also require incorporating deep learning-based nanoparticle detection or segmentation methods. Additionally, further research could explore strategies for deepening the GNN architecture while maintaining training stability and enhancing performance.

REFERENCES

- [1] Hutchings, G. J. 2009. Heterogeneous catalysts—discovery and design. *J. MaterChem.* 19(9), 1222–1235. <https://doi.org/10.1039/B812300B>
- [2] Tao, F. (Ed.). 2014. *Metal Nanoparticles for Catalysis: Advances and Applications*. Catalysis Series. Royal Society of Chemistry. <https://doi.org/10.1039/9781782621034>
- [3] Lunxiang, L. and Liebscher, J. 2007. Carbon–Carbon Coupling Reactions Catalyzed by Heterogeneous Palladium Catalysts. *Chem. Rev.* 107(1), 133–173. <https://doi.org/10.1021/cr0505674>
- [4] Mao, Z., Gu, H. and Lin, X. 2021. Recent Advances of Pd/C-Catalyzed Reactions. *Catalysts* 11(9), 1078. <https://doi.org/10.3390/catal11091078>
- [5] Felpin, F.-X. 2014. Ten Years of Adventures with Pd/C Catalysts: From Reductive Processes to Coupling Reactions. *Synlett* 25(08), 1055–1067. <https://doi.org/10.1055/s-0033-1340668>
- [6] Suga, M. *et al.* 2014. Recent progress in scanning electron microscopy for the characterization of fine structural details of nanomaterials. *Prog. Solid State Chem.* 42(1–2), 1–21. <https://doi.org/10.1016/j.progsolidstchem.2014.02.001>
- [7] Boiko, D. A., Pentsak, E. O., Cherepanova, V. A. *et al.* 2020. Electron microscopy dataset for the recognition of nanoscale ordering effects and location of nanoparticles. *Sci. Data* 7, 101. <https://doi.org/10.1038/s41597-020-0439-1>
- [8] Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M. and Solomon, J. M. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* 38(5), Article 146, 12 pages. <https://doi.org/10.1145/3326362>
- [9] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. and Bengio, Y. 2018. Graph Attention Networks. In: *Int. Conf. on Learning Representations (ICLR)*. <https://openreview.net/forum?id=rJXMpikCZ>
- [10] LeCun, Y., Bengio, Y. and Hinton, G. 2015. Deep learning. *Nature* 521, 436–444. <https://doi.org/10.1038/nature14539>
- [11] Fahim, S. F., Tasnim, N., Kibria, G., Morshed, M. S., Nishat, Z. T., Azad, S. B., Nath, S., Dey, A. L., Shuddho, M. A. and Niloy, N. T. 2024. A Proficient Convolutional Neural Network for Classification of Bone Age from X-Ray Images. In: *Proc. 9th Int. Conf. on Research in Intelligent Computing in Engineering, Ann. Comput. Sci. Inf. Syst.* 42, 17–21. <http://dx.doi.org/10.15439/2024R60>
- [12] Bruno, A., Moroni, D. and Martinelli, M. 2023. Efficient Deep Learning Approach for Olive Disease Classification. In: *Proc. 18th Conf. on Computer Science and Intelligence Systems*, eds. Ganzha, M., Maciaszek, L., Paprzycki, M., Ślęzak, D., *ACSIS* 35, 889–894. <http://dx.doi.org/10.15439/2023F4794>
- [13] Ciecholewski, M. 2023. Urban Scene Semantic Segmentation Using the U-Net Model. In: *Proc. 18th Conf. on Computer Science and Intelligence Systems*, eds. Ganzha, M., Maciaszek, L., Paprzycki, M., Ślęzak, D., *ACSIS* 35, 907–912. <http://dx.doi.org/10.15439/2023F3686>
- [14] Faraz, K., Grenier, T., Ducottet, C. *et al.* 2022. Deep Learning Detection of Nanoparticles and Multiple Object Tracking of Their Dynamic Evolution During In Situ ETEM Studies. *Sci. Rep.* 12, 2484. <https://doi.org/10.1038/s41598-022-06308-2>
- [15] Ronneberger, O., Fischer, P. and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W. and Frangi, A. (eds), *Med. Image Comput. Comput.-Assist. Interv. – MICCAI 2015, Lect. Notes Comput. Sci.* 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28
- [16] Alnaasan, M., Al Zoubi, W., Alhammadi, S., Kang, J.-H., Kim, S. and Ko, Y. G. 2024. Well-Defined High Entropy-Metal Nanoparticles: Detection of the Multi-Element Particles by Deep Learning. *J. Energy Chem.* 98, 262–273. <https://doi.org/10.1016/j.jechem.2024.06.038>
- [17] Mohsin, A. S. M. and Choudhury, S. H. 2024. Label-Free Quantification of Gold Nanoparticles at the Single-Cell Level Using a Multi-Column Convolutional Neural Network (MC-CNN). *Analyst* 149(8), 2412–2419. <https://doi.org/10.1039/D3AN01982A>
- [18] Kirillov, A., *et al.*, 2023. Segment Anything. In: *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Paris, France, pp. 3992–4003. <https://doi.org/10.1109/ICCV51070.2023.00371>
- [19] Monteiro, G. A. A., Monteiro, B. A. A., dos Santos, J. A., *et al.*, 2025. Pre-trained artificial intelligence-aided analysis of nanoparticles using the segment anything model. *Sci. Rep.* 15, 2341. <https://doi.org/10.1038/s41598-025-86327-x>
- [20] Boiko, D. A., Pentsak, E. O., Cherepanova, V. A., Gordeev, E. G., and Ananikov, V. P., 2021. Deep neural network analysis of nanoparticle ordering to identify defects in layered carbon materials. *Chem. Sci.* 12, 7428–7441. <https://doi.org/10.1039/D0SC05696K>
- [21] Simonovsky, M., and Komodakis, N., 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, pp. 29–38. <https://doi.org/10.1109/CVPR.2017.11>
- [22] Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., and Bennamoun, M., 2021. Deep learning for 3D point clouds: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 43(12), 4338–4364. <https://doi.org/10.1109/TPAMI.2020.3005434>
- [23] Zhang, S., Tong, H., Xu, J., and Maciejewski, R., 2019. Graph convolutional networks: a comprehensive review. *Comput. Soc. Netw.* 6, 11. <https://doi.org/10.1186/s40649-019-0069-y>
- [24] Jiao, Z., Liu, Y., and Wang, Z., 2024. Application of graph neural network in computational heterogeneous catalysis. *J. Chem. Phys.* 161(17), 171001. <https://doi.org/10.1063/5.0227821>
- [25] Gu, G. H., Noh, J., Kim, S., Back, S., Ulissi, Z., and Jung, Y., 2020. Active learning across intermetallics to guide discovery of electrocatalysts for CO₂ reduction and H₂ evolution. *J. Phys. Chem. Lett.* 11(9), 3185–3191. <https://doi.org/10.1021/acs.jpclett.0c00634>
- [26] Mehta, B., Kothari, K., Nambiar, R., and Shrawne, S., 2024. Toxic molecule classification using graph neural networks and few-shot learning. In: *Proc. 19th Conf. Comput. Sci. Intell. Syst. (FedCSIS)*, ACSIS, vol. 41, pp. 105–110. <http://dx.doi.org/10.15439/2024F3810>
- [27] Hinton, G. E., and Roweis, S., 2002. Stochastic neighbor embedding. *Adv. Neural Inf. Process. Syst.* 15. <https://dl.acm.org/doi/10.5555/2968618.2968725>
- [28] Dosovitskiy, A., *et al.*, 2020. An image is worth 16×16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. <https://doi.org/10.48550/arXiv.2010.11929>
- [29] Oquab, M., *et al.*, 2023. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*. <https://doi.org/10.48550/arXiv.2304.07193>
- [30] Li, Q., Han, Z., and Wu, X.-M., 2018. Deeper insights into graph convolutional networks for semi-supervised learning. *Proc. AAAI Conf. Artif. Intell.* 32(1). <https://doi.org/10.1609/aaai.v32i1.11604>
- [31] Boiko, D. A., Sulimova, V. V., Kurbakov, M. Y., Kopylov, A. V., Seredin, O. S., Cherepanova, V. A., Pentsak, E. O., and Ananikov, V. P., 2022. Automated recognition of nanoparticles in electron microscopy images of nanoscale palladium catalysts. *Nanomaterials* 12(21), 3914. <https://doi.org/10.3390/nano12213914>
- [32] OpenCV team, 2025. OpenCV: Open Source Computer Vision Library. Accessed: 2025-05-25. https://docs.opencv.org/4.x/d0/d7a/classcv_1_1SimpleBlobDetector.html
- [33] Kingma, D. P., and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. <https://doi.org/10.48550/arXiv.1412.6980>
- [34] He, K., Zhang, X., Ren, S., and Sun, J., 2016. Deep residual learning for image recognition. In: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- [35] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S., 2022. A ConvNet for the 2020s. In: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 11976–11986. <https://doi.org/10.1109/CVPR52688.2022.01167>
- [36] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. In: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, pp. 248–255. <https://doi.org/10.1109/CVPR.2009.5206848>