Review of Graph Neural Networks for Medical Image Denoising

Article *in* EAI Endorsed Transactions on AI and Robotics · December 2023

DOI: 10.4108/eetel.4358

1 author:

Jing Wang
Henan Polytechnic University
3 PUBLICATIONS **1** CITATION

EAI Endorsed Transactions
on _____ Research Article EAI.EU

Review of Graph Neural Networks for Medical Image Denoising

Jing Wang[1,*]

[1]1st School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan 454000, P R China

Abstract

As deep learning continues to evolve, more and more applications are generating data from non-Euclidean domains and representing them as graphs with complex relationships and interdependencies between objects. This poses a significant challenge to deep learning algorithms. Because, due to the uniqueness of graphs, applying deep learning to ubiquitous graph data is not an easy task. To solve the problem in non-Euclidean domains, graph Neural Networks (GNNs) have emerged. A graph neural network (GNN) is a neural model that captures dependencies between graphs by passing messages between graph nodes. With the continuous development of medical image technology, medical image diagnosis plays a crucial role in clinical practice. However, in practice, medical images are often affected by noise, artifacts, and other interfering factors, which may lead to inaccurate and unstable diagnostic results. Therefore, image-denoising techniques become especially critical in medical image processing. Therefore, researchers have proposed innovative methods based on graph neural networks for effective noise removal, preserving the key features of the image and improving the quality and usability of medical images. This paper reviews the research progress of graph neural networks in the field of medical image denoising. It also summarises the problems and challenges of current research and looks at the future direction of medical image-denoising research.

Keywords: graph neural networks; deep learning; graph data; medical images; denoising.

Received on DD MM YYYY, accepted on DD MM YYYY, published on DD MM YYYY

doi: 10.4108/_____

1. Introduction

With the rapid development of neural networks in recent years, deep learning has become a "jewel" in the field of artificial intelligence and machine learning [1]. Many machine learning tasks [2, 3] that used to rely on manual methods to extract feature information (e.g., image recognition, machine translation) have been replaced by a variety of more advanced deep learning methods. Of course, the success of deep learning in image classification [4], video processing [5], speech recognition [6], natural language understanding [7] is no accident. This is due not only to big data [8] and high-performance computing power [9] but also to the effectiveness of deep learning itself in extracting potential representations from Euclidean data [10].

For graphs can be regular or irregular. A graph may simultaneously have unordered nodes of different sizes, nodes from the same graph may have different numbers of neighbors as well as each node neighborhood in the graph may be different. This leads to the fact that some operations of deep learning algorithms (e.g., convolution operations [11]) can achieve good results in the Euclidean domain, but are difficult to apply to the graph domain.

Graphs are ubiquitous and widely used in the real world to represent objects and their relationships in various domains. Examples include large-scale social networks[12], traffic networks [13], chemical molecule analysis[14], recommender systems [15], ecosystems, and so on. More and more applications rely on representing data generated in non-Euclidean [16] domains as graphs with complex

*Corresponding author. Email: wangjing@home.hpu.edu.cn

relationships and interdependencies between objects. The complexity of graph structures poses significant challenges to existing deep-learning algorithms. In recent years, there has been a strong interest in deep learning methods for extending graph data, and graph neural networks (GNNs) [17] have emerged, driven by deep learning algorithms such as convolutional neural networks (CNNs) [18] and recurrent neural networks (RNNs) [19]. The emergence of graph neural networks has made it possible to apply deep learning algorithms to solve graph problems in non-Euclidean domains.

Graph Neural Networks (GNN) is a deep learning algorithm based on graph structure that learns the representation of nodes and edges in a graph and achieves tasks such as classification, clustering, and prediction of the graph as a whole. Unlike traditional machine learning algorithms that need to transform the graph into vectors or matrices, graph neural networks improve the representation of graph data by performing computations directly on the graph, using the relationships between nodes. For example, in social network analysis, graph neural networks can help us with tasks such as discovering community structure and predicting user interests and behaviors. In chemical molecule analysis, graph neural networks can help us with tasks such as classification, clustering, and prediction of molecules. In recommender systems, graph neural networks can leverage relationships between users to improve recommendations.

Early research on Graph Neural Networks (GNN) falls under the category of Recurrent Neural Networks (RecGNN) with high overhead. Ref. [20] introduced neural networks to directed acyclic graphs and promoted the study of GNNs. Gori, et al. [21] were the first to introduce the concept of graph neural networks. [22, 23] further elaborated the concept of graph neural networks.

In the last few years as the use of non-Euclidean data[24] has become more widespread, more attention has been focused on the study of graph neural networks. Wu, et al. [25] classify graph neural networks into four categories. Zhang, et al. [26] provide a comprehensive review of deep learning methods on different types of graphs. Thomas, et al. [17] categorizes graph neural networks based on their ability to process graph types and attributes. Waikhom and Patgiri [27] outline the learning approaches of graph neural networks. Zhou, et al. [28] propose a generic pipeline design for graph neural network models. There are also many research works on graph neural network learning methods. Cao, et al. [29] extract feature information in hyperspectral classification to avoid the problem of over-smoothing of message passing caused by Yang, et al. [30]. As the research work progressed, methods based on contrast learning were also successful. Okuda, et al. [31] proposed unsupervised graph representation learning to discover common objects and a set of specific objects in an image for localization. The node classification and edge detection combine two learning methods, random walk, and language modeling, and the learned representation can be used for downstream tasks.

The rest of the paper is as follows: Section 2 introduces the basic structure of a graph neural network. Section 3 details the graph convolutional neural network, an important part of a graph neural network. Section 4 introduces the graph attention network. Section 5 focuses on the application of graph neural networks in the process of medical image denoising. Section 6 introduces the training and learning methods of graph neural networks. Section 7 explains the importance of denoising medical images. In Section 8, the paper is summarized and the applications and prospects of graph neural networks in medical image denoising are prospected.

2. Basic Structure of GNNs

Graph Neural Network (GNN) is a deep learning model specially designed for processing graph data. Different from traditional neural networks, GNN [32] can capture complex relationships in graph data and thus has a wide range of applications in the fields of social network analysis, recommender systems, bioinformatics, etc. The basic structure of GNN includes components such as graph representation, graph convolutional layer, aggregation function, activation function, graph pooling layer, and graph attention mechanism [33]. Among them, the graph convolution layer is the core component, which learns the representation of a node through feature propagation from neighboring nodes. The aggregation function determines how nodes interact with their neighbors and the activation function[34] introduces nonlinearity. The graph pooling layer is used to reduce the size of the graph, while the graph attention mechanism gives different weights to different neighbor nodes to improve the network representation. Multiple graph convolutional layers [35] can be cascaded and stacked to form deep networks that are trained by backpropagation algorithms. GNNs can efficiently learn and express complex relationships in a variety of graph-structured data through the combination of these structures and components. Figure **1** shows the process of updating the information of the central node by aggregating the information of the neighboring nodes based on the message-passing paradigm.

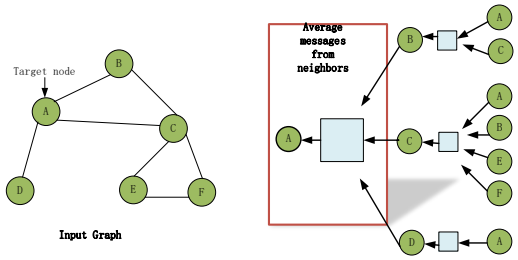

Figure 1: Node information update

The core of a graph neural network is node embedding. Each node in the graph is associated with a feature vector

that encodes information about that node. These feature vectors are the initial representations of the nodes in the graph. Node embeddings capture attributes of nodes, such as text, images, or numeric data, which form the starting point for computation in a graph neural network. In each layer of a graph neural network, a node collects information about neighboring nodes by aggregating the embeddings of neighboring nodes [36]. In different graph neural network architectures, the aggregation may be done in different ways, such as weighted summation, graph convolution, or attention mechanisms. The aggregated information will be used to update the embedding of the nodes. Aggregation functions define how to merge information from neighboring nodes. Common aggregation functions include mean aggregation (averaging), sum aggregation (summing), and more complex operations such as attention-based aggregation, where nodes weigh the importance of neighboring information based on a learned attention score. The choice of aggregation function can greatly affect the performance and expressiveness of the GNN.

Common graph neural network models include **G**raph **C**onvolutional **N**eural Network (GCN) [37, 38], **G**raph **At**tention Network (GAT) [39, 40], GraphSAGE, Graph Isomorphism Network (GIN), etc. This paper focuses on GCN and GAT.

3. Graph Convolutional Neural Networks

GCN is a type of convolutional neural network that acts directly on the graph and uses its structural information. The main idea of GCN is that, for each node, we take into account all its neighbors as well as the feature information it contains. Assuming that we use the average() function, then performing the above operations on each node gives us a representation of the average value that can be fed into the neural network. Modern GCNs mimic CNNs by designing convolution and readout functions to learn common local and global structural patterns of graphs.

We begin with a discussion of the convolution operation and then move to the readout operation and some other improvements. Convolutional neural networks play a central role in building many other complex GNN models. Graph convolutional neural networks include spatial-based graph convolutional neural networks and spectral-based graph convolutional neural networks. Table **1** explains some of the notations covered in section 3.1.

3.1. Space-based graph convolutional neural network

The spatial domain-based graph convolutional neural structure mainly consists of three types of operators: neighbor sampling, message computation, and message aggregation. The neighboring nodes represented by a particular node are aggregated using the aggregate operation in GCN as a way to achieve message passing between nodes. Figure 2 represents the passing of node messages in spatial domain-based GCN.

Table 1: Notations Definition

Notation Name	Notation Expression
Graph	G
Node features	x
The adjacency matrix of the graph	E
Unit matrix	I
Normalised graph Laplace matrix	$\boldsymbol{L}$
Diagonal matrix of node degrees	$\boldsymbol{D}$
Eigenvector matrix	$\boldsymbol{U}$
Diagonal eigenmatrix	$\boldsymbol{\Lambda}$
Graph signals	**X**(feature vectors consisting of individual nodes in the graph)
Filters	g

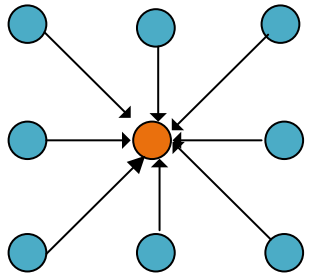

Figure 2: Spatial domain-based GCN node messaging

The simplest aggregation process is to do the product operation between the node characteristics(x) of the graph and the topological structure information of the graph (adjacency matrix E). The specific process is shown in Figure 3.

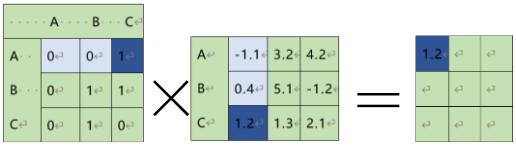

Figure 3: Information aggregation process of GCN based on spatial domains

To solve the problem that Figure 3 does not calculate the node's features and aggregation by summation directly will lead to gradient explosion or disappearance, we can add the

unit matrix I to the adjacency matrix E and aggregate the features of the neighboring nodes by taking a weighted average.

Depending on the different methods of convolutional layer stacking, spatial-based GCNs can be further classified into two categories: recurrent-based and composition-based spatial GCNs. recurrent-based methods use the same graph convolutional layers to update the hidden representations, and composition-based methods use different graph convolutional layers to update the hidden representation. Figure 4 illustrates this difference.

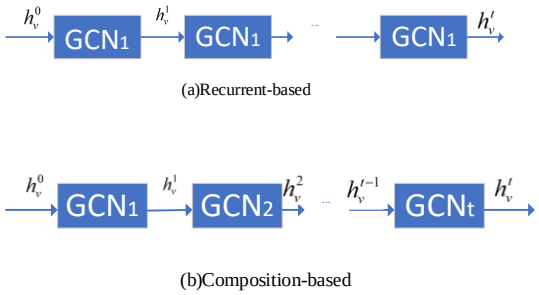

Figure 4: Comparison of Recurrent-based and Composition-based method in spatial-based GCNs

The spatial approach is to define the convolution directly in the spatial domain, the problem faced is that, because each node's neighbors are not the same size, it is not possible to define a neighborhood of the same size, so the implementation of parameter-sharing faces greater difficulties, but the idea is still, convolution is still a node in the neighboring nodes on the weighted average, so the subsequent many methods aimed at solving the problem of parameter sharing.

3.2. Spectral-based graph convolutional neural networks

Convolution based on spectral methods is a special case of convolution based on spatial methods. Spectral domain-based graph convolution via neural network investigates the properties of the graph with the help of eigenvalues and eigenvectors of the Laplace matrix of the graph. Filters are introduced to define convolution from a signal-processing perspective. Firstly, multiplication is done on the signal in the spectral domain using the theorem of convolution. Secondly, the convolution is implemented by transforming the signal into the original space using the Fourier transform [41]. This approach avoids the problem of difficulty in defining convolution caused by the fact that the graph data does not satisfy translation invariance. Because the structure of the graph does not satisfy the translation invariance, it is not possible to define the convolution directly in the spatial domain, so the signal is transformed to the frequency domain, the convolution operation is implemented in the frequency domain and converted back to the spatial domain, which is the spectral method.

Graph convolutional neural networks based on spectral methods assume that the graph is undirected. The normalized graph Laplace matrix is the mathematical representation of the undirected graph and is defined as:

$$L = I_n - D^{(-1/2)} E D^{(-1/2)}, \tag{1}$$

where $\boldsymbol{D}$ denotes the diagonal matrix of node degrees. $\boldsymbol{D}_{ii} = \sum_j (\boldsymbol{E}_{i,j})$, $\boldsymbol{E}$ denotes the adjacency matrix of the graph. Using the property of symmetric positive semidefinite of the graph Laplace matrix, the normalized Laplace matrix can be decomposed as:

$$L = U \Lambda U^T, \tag{2}$$

where $\boldsymbol{U} = [\boldsymbol{u_0}, \boldsymbol{u_1}, \cdots, \boldsymbol{u_{n-1}}] \in \boldsymbol{R}^{n \times n}$ is the Feature vector matrix, $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues (spectrum), $\boldsymbol{\Lambda}_{ii} = \lambda_i$. The feature vectors of the regularised Laplacian matrix form a set of orthogonal bases. In graph signal processing, the signal of a graph $X \in R^N$ is a feature vector consisting of the individual nodes of the graph, X_i representing the i^{th} node. The Fourier transform and Fourier inverse transform of a graph G are defined as:

$$F(X) = U^T X, \tag{3}$$

$$F^{(-1)}(\hat{X}) = U\hat{X}, \tag{4}$$

where, $\hat{X}$ is the result of the Fourier transform. To better understand the Fourier transform of a graph, we can see from its definition that it does project the input graph signal into an orthogonal space whose base is made up of the eigenvectors of the regularised graph Laplacian. The elements of the transformed signal are the coordinates of the graph signal in the new space so that the original input signal can be expressed as:

$$X = \sum_i \hat{X}_i u_i. \tag{5}$$

This is the result of the Fourier inverse transform. Next, we can define the graph convolution operation on the input signal X.

$$X *_G g = F^{-1}\big(\mathscr{F}(X) \odot \mathcal{F}(g)\big) = U(U^T X \odot U^T g) \quad (6)$$

where, $g \in R^N$ is the filter we define; $\odot$ Indicates the Hadamard product. Suppose we define such a filter:

$$g_\Theta = diag(U^T g). \quad (7)$$

Thus, the graph convolution operation can be represented in a simplified way as:

$$X *_G g_\theta = \boldsymbol{U} \boldsymbol{g}_\theta \boldsymbol{U}^T X. \quad (8)$$

Spectral-based graph convolution networks all follow this pattern, with the key difference between them being the choice of filters. Existing spectral-based graph convolution network models are the following: Spectral CNN, Chebyshev Spectral CNN (ChebNet), and Adaptive Graph Convolution Network (AGCN).

4. Graphical Attention Network

Graph Attention Network(GAT)[27] consists of many functionally identical blocks (Graph Attention Layer). Its properties include high efficiency, low storage type, inductive learning, and full graph access. The graph attention layer has a feature value of $\vec{h} = \{\vec{h}_1, \vec{h}_2, \cdots, \vec{h}_N\}, \vec{h}_i \in \mathbb{R}^F$ for the node at input. where N represents the number of nodes and F represents the dimensionality of the node features. After a Graph Attention Layer, a new feature vector is output, which can be represented as $\vec{h}' = \{\vec{h}'_1, \vec{h}'_2, \ldots, \vec{h}'_N\}, \vec{h}' \in \mathbb{R}^{F'}$, assuming that the dimension of the node feature of this feature vector is F'. As shown in Figure 5.

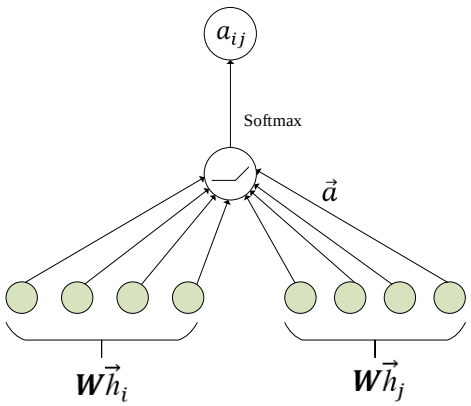

Figure 5: Attention layer in GAT

The purpose of using Self-attention is to improve the expressiveness of $\vec{h}'$. In the Graph Attention Layer, a weight matrix $\boldsymbol{W} \in \mathbb{R}^{F' \times F}$ is first applied to each node using a weight matrix, and then self-attention is used for each node to calculate an attention coefficient, the shared self-attention mechanism used here, denoted as:

$$e_{ij} = a(\boldsymbol{W}h_i, \boldsymbol{W}\vec{h}_j), \quad (9)$$

e_{ij} represents the importance of node j for node i. In theory, we can calculate the weight of any node in the graph to the central node. In GAT, to simplify the calculation, the nodes are restricted to the one-hop neighbors of the central node, and in addition, the nodes take themselves into account as neighboring nodes. In the existing studies, there are many ways to choose. For example, choosing a single-layer feedforward network with parameter $\vec{a} \in \mathbb{R}^{2F'}$ and then using LeakyReLU to do a non-linearisation gives.

$$e_{ij} = \text{LeakyReLU}\big(\vec{a}^T[\boldsymbol{W}\vec{h}_i \parallel \boldsymbol{W}h_j]\big). \quad (10)$$

Finally, the neighboring nodes of the central node are normalized using softmax:

$$a_{ij} = \text{softmax}\,(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}i} \exp(e_{ik})}. \quad (11)$$

The output feature $\vec{h}'$ is obtained by weighting the input features.

$$\vec{h}' = \sigma\big(\textstyle\sum_{j \in \mathcal{N}_i} a_{ij}\vec{h}_j\big). \quad (12)$$

To improve the generalization of the attention mechanism, GAT chose to use a multi-headed attention layer, i.e. a single-headed attention layer from a set of K mutually independent graph attention layers, and then stitch their results together. At this point, h'_i is:

$$h'_i = \|_{k=1}^K \sigma\big(\textstyle\sum v_j \in \widetilde{\mathcal{N}}(v_i) a_{ij}^{(k)} \boldsymbol{W}^k h_j\big), \quad (13)$$

where $\|$ represents the splice operation, $a_{ij}^{(k)}$ represents the weight factor calculated from the k^{th} group of attention mechanisms, and $\boldsymbol{W}^k$ is the weight factor of the k^{th} module. To reduce the dimensionality of the feature vector, we can also use the averaging operation instead of the splicing operation, as shown in the following equation.

$$\vec{h}_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in N j} a_{ij}^{k}\boldsymbol{W}^{k}\vec{h}_j\right). \quad (14)$$

5. Application of graph neural networks in medical image denoising

Graph neural networks have a wide range of potential applications in medical image denoising. Medical images [42] usually have complex structures and are highly noisy, e.g., X-rays [43, 44], CT scans [45], MRIs [46], etc., which must be of high quality in diagnosis and analysis. The main application directions of graph neural networks in medical image denoising [47, 48] include image noise reduction, image enhancement, motion artifact removal, data recovery, super-resolution reconstruction, and sequence image denoising. These applications will be specifically described below. Figure 6 is the comparison of medical images before and after denoising.

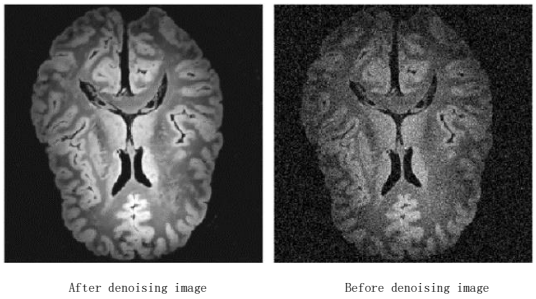

Figure 6 Comparison of images before and after denoising

(i) Image Noise Reduction. Graph neural networks can remove noise from medical images and improve image quality by learning the relationships between pixel points or blocks of pixels in an image. This approach is particularly important for image quality improvement in low-dose imaging techniques such as X-ray images.

(ii) Image Enhancement. In medical images, certain subtle structures may be masked by noise, and graph neural networks can help enhance these details, allowing doctors to see important features in the image more clearly.

(iii) Motion Artifact Removal. Artifacts due to factors such as patient movement or breathing are a common problem in medical imaging. Graph neural networks can learn and remove these artifacts to improve the accuracy of images.

(iv) Data Recovery. During medical image acquisition, data may be missing due to reasons such as equipment failure or sensor damage. Graph neural networks can be used to recover these missing data, enabling doctors to get more complete image information.

(v) Super-resolution reconstruction. Graph neural networks can be used for super-resolution reconstruction of medical images, i.e., recovering high-resolution images from low-resolution images to improve the clarity and detail of the images.

(vi) Sequence image denoising. In medical images, some data exists in the form of sequences, such as time-sequenced images in magnetic resonance imaging (MRI). Graph neural networks can be applied to these sequential data for denoising in the temporal and spatial domains to improve the quality of the images.

In conclusion, graph neural networks can better cope with the complex structure and highly noisy features of medical images by learning the relationship between pixel points or pixel blocks in medical images, providing clearer and more accurate images for medical imaging diagnosis.

6. Training and Learning Methods of GNNs

The training and learning of graph neural network (GNN) is similar to that of traditional neural networks, but since graph neural networks are trained on graph-structured data, the following special factors also need to be considered.

Forward propagation and backpropagation. During forward propagation, the GNN passes information through the graph. In each GNN layer, nodes aggregate information from their neighbors based on aggregation functions (e.g., mean, sum, maximum). This aggregated information is used to update the node embeddings[49-51]. The GNN uses gradient-based optimization techniques such as backpropagation to learn the model parameters, compute the loss function (a measure of the error between the predicted value and the actual value), and update the model's weights by backpropagating the gradients through the graph neural network layers.

Initialization and parameter learning. GNNs usually start with random or pre-trained node embeddings, which are used as initial node representations. Meanwhile, GNNs learn parameters through training to improve the model's effectiveness. Commonly used parameter optimization algorithms include stochastic gradient descent (SGD) algorithm [52] or its variants, etc. Figure 7 illustrates the stochastic gradient descent.

Semi-supervised learning. GNN[53] is very effective in semi-supervised learning environments where only a small fraction of the nodes in the graph are labeled. During training, the GNN makes predictions using both labeled and unlabelled nodes. Unlabelled nodes benefit from the information propagated through the graph, which helps to improve generalization. The loss function typically consists of a supervised loss term for the labeled nodes and a regularisation term, which encourages smoothness in the embedding of neighboring nodes. The regularisation term ensures that nodes with similar network neighborhoods have similar embeddings.

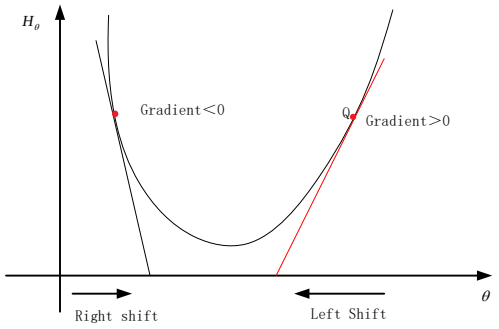

Figure 7: Stochastic Gradient Descent (SGD)

Overfitting and regularisation. Like other neural networks, GNNs can suffer from overfitting problems, especially when the graph is small or the model is too complex. To prevent overfitting, regularisation techniques such as dropout and L2 regularisation can be used. In addition, aggregation functions (e.g. attention-based mechanisms) in the GNN architecture can weigh neighboring nodes so that they learn to focus on important neighboring nodes, which can be used as a form of implicit regularisation.

Graph Convolution and Aggregation Functions. The choice of aggregation function is crucial in GNN training. Different functions lead to different results in information propagation. Researchers often experiment with various aggregation functions to find the best fit for a particular task. Architectures such as Graph Convolutional Neural Networks (GCN), Graph Attention Networks (GAT), and GraphSAGE use specific aggregation functions that are effective in different scenarios. Learning the parameters of these functions is an important part of GNN training.

7 Importance of Medical Image Denoising

Medical image denoising plays a vital role in all aspects of healthcare and medical research, Table 2 describes several common methods of denoising medical images. The importance of denoising medical images includes several aspects.

Improved diagnosis and treatment. Medical images such as X-rays, CT scans, MRI scans, and ultrasound images are essential tools for medical professionals to diagnose diseases and develop treatment plans. Noise in these images can distort important features, making it difficult to detect subtle anomalies. By reducing noise in medical images through denoising technology, medical practitioners can obtain clearer and more accurate anatomical structures, lesions, and other pathological features. This will lead to more confident diagnoses, better treatment planning, and ultimately improv ed patient outcomes.

Table 2: Medical image denoising methods

Medical image denoising methods	Method Description
Mean Filtering	Replace the value of each pixel with the average of the surrounding neighborhood pixels for mild noise.
Gaussian Filtering	Smooths the image using the Gaussian function, effective for Gaussian noise (normally distributed noise). Consider pixel weights for continuity noise.
Median Filtering	Replacing the value of each pixel with the median value of the neighborhood pixels works well for pretzel noise (bursty noise).
Wavelet Denoising	The image is decomposed into different frequency subbands and the noise is removed by thresholding.
Non-local Means Denoising	Compare the similarity of different regions, weighted average per pixel, for high similarity regions.
Deep Learning	Using Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs), image noise distribution is learned and denoised on large-scale datasets, adapting to different noise types.
Total Variation Denoising	Nonlinear denoising method for smoothing images, preserving edge information.
K-SVD Denoising	Based on dictionary learning, the image is decomposed into linear combinations of bases, and noise is removed.

Reduce radiation exposure. In the case of X-rays and CT scans, denoising medical images helps to reduce the required radiation dose. Reducing the amount of radiation is essential for patient safety, especially for patients who require frequent imaging, such as cancer patients undergoing radiation therapy. Denoising algorithms can improve the quality of low-dose images, making them diagnostic while minimizing the potential risks associated with radiation exposure. This ensures that patients are not unnecessarily harmed while undergoing the necessary medical imaging.

Contribute to research and training. Denoising medical images is critical to medical research because clean and high-quality images are needed to study diseases, develop

new treatments, and evaluate the effectiveness of medical interventions. Medical professionals and researchers also use denoised images for educational and training purposes. Clear images are essential for the teaching of medical students and for the training of medical practitioners in image interpretation and surgical planning.

Telemedicine and tele-consultation. With the rise of telemedicine and remote consultations, especially in underserved or remote areas, denoising technology can help improve the quality of transmitted medical images. Telemedicine relies heavily on the accurate interpretation of images by remote medical specialists. Removing noise from images before transmission ensures that telemedicine providers make accurate diagnoses and treatment recommendations. This is especially important in emergency cases and in situations where specialized medical services are limited.

8. Conclusions

This paper describes graph neural networks and the application and impact of graph neural networks in medical image denoising. With the continuous deepening of research, graph neural network technology is also developing, and new models and algorithms are emerging. Therefore, we can foresee that the future research and application of graph neural networks in medical image denoising will be more extensive and in-depth. However, the current application of graph neural networks in medical image denoising also faces many challenges. For example, (1) insufficient data volume and labeling difficulties. Medical image[54] data is usually relatively limited, and labeling medical image data is a very expensive and time-consuming task. Therefore, the lack of large-scale and high-quality labeled data is a challenge, especially when a large amount of labeled data is required in deep-learning models. (2) Complex noise types and distribution. There are various types of noise in medical images, including random noise, artifacts, motion artifacts, etc., and the distribution may be very complex. Different types of noise require different processing methods, and it is a challenge to design graph neural networks that can handle various types of noise. (3) Computational complexity. Graph neural networks usually require a large amount of computational resources, especially for complex graph structures and large-scale medical image datasets. In practical applications, it is necessary to consider how to efficiently train and deploy these models in environments with limited computational resources. In the future, research on the application of graph neural networks in medical image denoising should be considered in the context of these issues.

Acknowledgements.

We thank all the anonymous reviewers for their hard reviewing work.

Funding

This research did not receive any grants.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-44, May 28 2015.

[2] Y. Zhang, "Smart detection on abnormal breasts in digital mammography based on contrast-limited adaptive histogram equalization and chaotic adaptive real-coded biogeography-based optimization," *Simulation*, vol. 92, no. 9, pp. 873-885, September 12, 2016 2016.

[3] S. Wang, "Pathological Brain Detection by a Novel Image Feature—Fractional Fourier Entropy," *Entropy*, vol. 17, no. 12, pp. 8278-8296, 2015.

[4] I. Nedjar, M. Brahimi, S. Mahmoudi, K. A. Ayad, and A. Chikh, "Exploring Regions of Interest: Visualizing Histological Image Classification for Breast Cancer using Deep Learning," *ArXiv*, vol. abs/2305.20058, 2023.

[5] O. Koepueklue, S. Hoermann, F. Herzog, H. Cevikalp, and G. Rigoll, "Dissected 3D CNNs: Temporal skip connections for efficient online video processing," *Computer vision and image understanding: CVIU*, p. 215, 2022.

[6] A. Amrouche and J. Rouvaen, "On the Use of the Nonparametric Regression in Neural Network Based Approach Applied to Arabic Speech Recognition," 2022.

[7] W. Ceusters, "Medical Natural Language Understanding as a Supporting Technology for Data Mining in Healthcare," 2022.

[8] L. Butcher, "Big data," *Engine + Powertrain Technology International*, no. Jan., 2022.

[9] M. M. Al-Qutt, H. Khaled, and R. E. Gohary, "Neural Network Inversion-Based Model for Predicting an Optimal Hardware Configuration: Solving Computationally Intensive Problems," *International journal of grid and high performance computing*, no. 2, p. 13, 2021.

[10] Z. Alavi, S. Sharma, L. Zhou, and K. Chen, "Scalable Euclidean Embedding for Big Data," in *IEEE International Congresson Big Data;IEEE Asia -Pacific Conference on Services Computing;IEEE International conference on cloud computing;IEEE World Congress on Services*.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017.

[12] L. Backstrom and J. Leskovec, "Supervised Random Walks: Predicting and Recommending Links in Social Networks," in *Web Search and Data Mining*, 2010.

[13] M. Curado, L. Tortosa, and J. F. Vicent, "Deep learning model of convolutional neural networks powered by a genetic algorithm for prevention of traffic accidents severity," 2023.

[14] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural Message Passing for Quantum Chemistry," 2017.

[15] S. Wu, Y. Tang, Y. Zhu, L. Wang, and T. Tan, "Session-Based Recommendation with Graph Neural Networks," 2019.

[16] Michael *et al.*, "Geometric Deep Learning: Going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18-42, 2017.

[17] J. M. Thomas, A. Moallemy-Oureh, S. Beddar-Wiesing, and C. Holzhüter, "Graph Neural Networks Designed for Different Graph Types: A Survey," pp. 200-212, 2022.

[18] K. Muhammad, "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation," *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3613-3632, 2019.

[19] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting," *IEEE*, no. 11, pp. 4883-4894, 2020.

[20] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, p. 714, 1997.

[21] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *IEEE International Joint Conference on Neural Networks*, 2005.

[22] 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, p. 61, 2009.

[23] C. Gallicchio and A. Micheli, "Graph Echo State Networks," in *International Joint Conference on Neural Networks*, 2010.

[24] Y. L. Xu, K. Konstantinidis, and D. P. Mandic, "Tensor Networks for Multi-Modal Non-Euclidean Data," 2021.

[25] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *IEEE transactions on neural networks and learning systems*, no. 1, p. 32, 2021.

[26] Z. Zhang, P. Cui, and W. Zhu, "Deep Learning on Graphs: A Survey," 2018.

[27] L. Waikhom and R. Patgiri, "Graph Neural Networks: Methods, Applications, and Opportunities," 2021.

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

[29] Z. Cao, X. Li, and L. Zhao, "Unsupervised Feature Learning by Autoencoder and Prototypical Contrastive Learning for Hyperspectral Classification," 2020.

[30] L. Yang, J. Gu, C. Wang, X. Cao, and Y. Guo, "Toward Unsupervised Graph Neural Network: Interactive Clustering and Embedding via Optimal Transport," in *2020 IEEE International Conference on Data Mining (ICDM)*, 2020.

[31] M. Okuda, S. I. Satoh, Y. Sato, and Y. Kidawara, "Unsupervised Common Particular Object Discovery and Localization by Analyzing a Match Graph," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021.

[32] S.-H. Wang, "Covid-19 classification by FGCNet with deep feature fusion from graph convolutional network and convolutional neural network," *Information Fusion*, vol. 67, pp. 208-229, 2020/10/09/ 2021.

[33] D. S. Guttery, "Improved Breast Cancer Classification Through Combining Graph Convolutional Network and Convolutional Neural Network," *Information Processing and Management*, vol. 58, 2, 2021, Art no. 102439.

[34] N. Takahashi, T. Yamakawa, Y. Minetoma, T. Nishi, and T. Migita, "Design of continuous-time recurrent neural networks with piecewise-linear activation function for generation of prescribed sequences of bipolar vectors," *Neural Networks: The Official Journal of the International Neural Network Society*, 2023.

[35] Y.-D. Zhang, "Multiple sclerosis identification by convolutional neural network with dropout and parametric ReLU," *Journal of Computational Science*, vol. 28, pp. 1-10, 2018/09/01/ 2018.

[36] A. K. Sangaiah, "Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization," *Neural Computing and Applications*, vol. 32, pp. 665-680, 2020.

[37] C. Tang, C. Hu, J. Sun, S. H. Wang, and Y. D. Zhang, "NSCGCN: A novel deep GCN model to diagnosis COVID-19," *Comput Biol Med*, vol. 150, p. 106151, Sep 30 2022.

[38] 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*, vol. PP, no. 99, pp. 1-1, 2021.

[39] J. X. A. B *et al.*, "Attention adjacency matrix based graph convolutional networks for skeleton-based action recognition - ScienceDirect," *Neurocomputing*, vol. 440, pp. 230-239, 2021.

[40] P. Velikovi, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," 2017.

[41] S. H. Wang and Y. D. Lv, "Alcoholism Detection by Data Augmentation and Convolutional Neural Network with Stochastic Pooling," *Journal of Medical Systems*, vol. 42, no. 1, Jan 2018, Art no. 2.

[42] Sato, Yoshinobu, Nakamoto, and Masahiko, "Image Guidance of Breast Cancer Surgery Using 3-D Ultrasound Images and Augmented Reality," *IEEE Transactions on Medical Imaging*, 1998.

[43] V. D. S. Rogier, N. Bhaskhar, D. Rubin, C. Langlotz, and A. Chaudhari, "Exploring Image Augmentations for Siamese Representation Learning with Chest X-Rays," 2023.

[44] V. T. N. Pham, Q. C. Nguyen, and Q. V. Nguyen, "Chest X-Rays Abnormalities Localization and Classification Using an Ensemble Framework of Deep Convolutional Neural Networks," *Vietnam Journal of Computer Science*, vol. 10, no. 01, pp. 55-73, 2023.

[45] Katja *et al.*, "Estimating the age of healthy subjects from T1-weighted MRI scans using kernel methods: Exploring the influence of various parameters," *NeuroImage*, vol. 50, no. 3, pp. 883-892, 2010.

[46] K. E. Watkins *et al.*, "Structural Asymmetries in the Human Brain: a Voxel-based Statistical Analysis of 142 MRI Scans," *Cerebral Cortex*, no. 9, pp. 868-877.

[47] P. Arora, P. Singh, A. Girdhar, and R. Vijayvergiya, "Performance analysis of various denoising filters on intravascular ultrasound coronary artery images," *International Journal of Imaging Systems and Technology*, vol. 33, no. 3, pp. 965-984, 2023.

[48] S. Y. Zhang *et al.*, "Hformer:highly efficient vision transformer for low-dose CT denoising," vol. 34, no. 4, pp. 161-174, 2023.

[49] H. Chi, Y. Wang, Q. Hao, and H. Xia, "Residual Network and Embedding Usage: New Tricks of Node Classification with Graph Convolutional Networks," *JPhCS*, 2022.

[50] E. P. Barracchia, G. Pio, A. Bifet, H. M. Gomes, B. Pfahringer, and M. Ceci, "LP-ROBIN: Link prediction in dynamic networks exploiting incremental node embedding," *Information Sciences: An International Journal*, p. 606, 2022.

[51] Y. Zhu, J. Wang, K. Zhang, and J. Zhang, "Node Embedding and Classification with Adaptive Structural Fingerprint," *Neurocomputing*, vol. 502, pp. 196-208, 2022.

[52] Y. D. Zhang and S. Satapathy, "A seven-layer convolutional neural network for chest CT-based COVID-19 diagnosis using stochastic pooling," *IEEE Sens. J.*, vol. 22, no. 18, pp. 17573 - 17582, 2022.

[53] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," 2016.

[54] X. Liu, M. Qin, Y. He, X. Mi, and C. Yu, "A new multi-data-driven spatiotemporal PM2.5 forecasting model based on an ensemble graph reinforcement learning convolutional network," *Atmospheric Pollution Research*, pp. 274-320, 2021.