

Review of Image Classification Algorithms Based on Graph Convolutional Networks

Wenhao Tang^{1,*}

¹ School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan 454000, PR China

Abstract

In recent years, graph convolutional networks (GCNs) have gained widespread attention and applications in image classification tasks. While traditional convolutional neural networks (CNNs) usually represent images as a two-dimensional grid of pixels when processing image data, the classical model of graph neural networks (GNNs), GCNs, can effectively handle data with the graph structure, such as social networks, recommender systems, and molecular structures. This paper summarizes the classical convolutional neural network models, highlighting their innovative approaches. And it will introduce the problems that graph convolutional networks have had, such as over-smoothing, and the ways to solve them, and suggest some possible future directions.

Keywords: Graph Convolutional Networks, Convolutional Neural Networks, Graph Neural Networks, over-smoothing.

Received on 17 June 2023, accepted on 02 July 2023, published on 06 July 2023

Copyright © 2023 Tang *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [Creative Commons Attribution license](#), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/airo.2023.3462

1. Introduction

1.1. Traditional Algorithms

Image classification [1] is a task in computer vision that aims to classify the input images into a predefined class or classes. The main processes of image classification are data preprocessing, feature extraction and selection, model training, evaluation, and application [2]. Preprocessing includes operations such as scaling, cropping, grayscale, and normalization for subsequent processing and analysis [3]. The feature extractor extracts meaningful features from the image and filters the extracted features to remove redundant or irrelevant features and keep the most representative ones [4]. After the model is trained, the performance needs to be evaluated with a test set to calculate the accuracy, recall, precision, and other metrics of the classifier. Finally, the trained classifier is applied to the new image data for classification prediction.

Traditional image classification methods include the k-nearest neighbor algorithm [5], support vector machine, rule-based models [6], etc. The k-nearest neighbor algorithm overly relies on the choice of distance metric function and k-value [7], is computationally intensive, requires large memory, is not very interpretable, and has a slow prediction speed. The training goal of traditional support vector machines [8] is to have the smallest error rate for the learned classifier obtained from the training set to approximate the distribution of the training set [9], which is achieved by finding the optimal classification hyperplane that maximizes the data interval of the training set. However, in some practical cases, choosing the correct kernel function to avoid overfitting is a difficult task when the number of features is much larger than the number of samples. Therefore k-nearest neighbor algorithm and SVM are suitable for classifying simple images [10]. And for some images that are heavily affected by noise or incomplete, the accuracy of classifiers based on them is significantly reduced [11].

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

1.2. Application Of Graph Convolutional Networks In Image Classification

One of the main advantages of deep learning models over traditional machine learning methods with hand-crafted features is the ability to efficiently extract high-level semantic information and rich detailed information [12].

The important backbone models of deep learning are convolutional neural networks, residual neural networks, attention mechanism, etc.

Graph Convolutional Network (GCN) is a deep learning model based on graph theory that operates directly on the graph structure [13]. In image classification tasks, GCN can transform an image into an undirected graph and compute the feature vectors of nodes directly using the adjacency matrix [14] and the trainable weight matrix [15]. Then, the values of the nodes and their connections are updated by a series of convolution operations [16]. Finally, the final classification results are obtained after a fully connected layer.

The graph convolution of the GCNs would mix node features and their nearby neighbors [17], that is, by using the local neighborhood information of the graph to perform feature aggregation for each node to learn the node representations.

In image classification tasks, GCNs can be used for semi-supervised classification tasks, where the core idea is to update the representation of nodes by propagating information between them. GCNs can be modeled by representing images as hypergraphs. In this case, each node represents a pixel in the image, and the edges represent the similarity between pixels. This approach can capture both global and local information in an image. By applying GCNs, the local features of each node can be combined with information from surrounding nodes to obtain a more discriminative feature representation.

The graph convolutional network has many advantages in image classification tasks. First, it can capture both local and global structural information in an image, which leads to a better understanding of the semantics of the image. Second, GCNs can learn the weights of each node adaptively, thus reducing the need for manually designed features. In addition, GCNs can handle changing graph structures with strong robustness and generalization ability.

1.3. Thesis Outline

Section 2 of this paper first summarizes classical convolutional neural networks in chronological order, highlighting their innovative points and advantages. Then, the concept of graph neural networks is elaborated and the classical models are summarized. Finally, the object of study, graph convolutional networks, is introduced and its concepts are elaborated. Section 3 of the paper first shows the common datasets of GCNs. Then it summarizes the advantages and disadvantages of recent graph convolutional network models with good performance and presents the problems they have not yet solved. Section 4

of the paper concludes and shows the future trends of graph convolutional networks.

2. Graph Convolutional Networks

2.1. Classic Convolutional Neural Networks

Convolutional neural networks (CNN) are a class of feed-forward artificial neural networks that use convolutional computation to extract features, as shown in Figure 1, mainly consisting of a convolutional layer, an activation layer, a pooling layer, and a fully-connected layer. The convolutional layer is responsible for extracting local features of the image, the activation layer introduces non-linearity, the pooling layer reduces the dimensionality of the features and the fully connected layer achieves the final classification task.

CNNs have a long history of development, and this paper reviews many classic models, starting with AlexNet, which won first place in the ImageNet Large Scale Visual Recognition Challenge.

AlexNet [18] made an important breakthrough in the history of CNNs by implementing the first deep CNN structure on a large-scale image dataset. AlexNet consists of five convolutional layers, three pooling layers, and three fully connected layers. Compared to LeNet, AlexNet has a deeper network structure and a larger parameter size. In addition, it introduces techniques such as the ReLU activation function, Dropout techniques, and data augmentation.

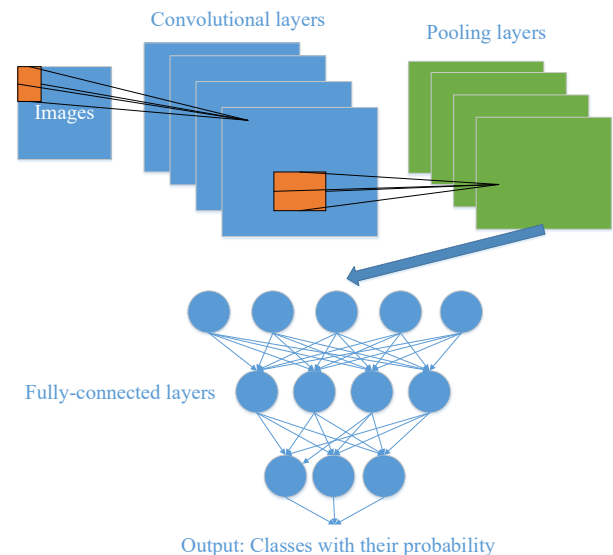


Figure 1. The architecture of CNN.

Simonyan and Zisserman (2014) proposed VGGNet with a higher number of layers, increasing the number of layers to 19. The main feature of VGGNet is the use of consecutive convolutional kernels of size 3×3 instead of

larger ones, which reduces the number of parameters and increases computational efficiency. The authors' experiments further confirm the conjecture that the higher the number of layers, the more accurate the prediction. The top-1 validation error dropped to 25.5% and the top-5 error dropped to 8.0% by layer 19.

The architecture of the Inception network [20] (also known as GoogLeNet) differs from the original preference for hierarchical stacking to a more flexible network model. The main innovation of Inception is the introduction of the Inception module, a structure for parallel stacking of convolutional kernels at different scales. Unlike the stacking of a single network of residuals or VGG networks, GoogLe-net solves the problem of model degradation at high levels in a different way. And a new network architecture with bifurcation is proposed. One of the reasons why the authors propose an optimization of a single network model like the VGG model is that as the layers are stacked, the useless weights at the shallow level cause a significant computational loss for the stacked weights at the deep level. Therefore, the authors propose a bifurcated network architecture to reduce the impact of this magnifying glass effect.

The bold innovation of the network structure of GoogLe-Net v1[20] also enabled the invention to beat VGG-net in the same year, winning first place in the 2014 Image-Net competition by a narrow margin of 0.01% accuracy improvement.

GoogLe-Net has evolved with its unique and convergent network structure, making good use of the BN layer in Inception-v2 [21] and replacing the convolutional kernel of size 5×5 with two convolutional kernels of size 3×3 to improve performance. Chollet (2017) replaced the Inception module with a depthwise separable convolution based on Inception-v3 (where the order of separable convolution is reversed from mobilnet, with pointwise convolution followed by depthwise convolution, and then combined with ResNet's On ImageNet, Xception is slightly more accurate than Inception-v3, while the number of parameters decreases, and the inclusion of a ResNet-like residual connectivity mechanism in the network significantly speeds up its convergence process and achieves significantly higher accuracy.

The key innovation of ResNet [23] is the introduction of the skip connection, as shown in Figure 2, which allows direct information transfer between network layers. This structure effectively solves the problem of gradient disappearance in deep networks, allowing the network to reach very large depths. The whole idea of residual networks treats the action between convolutional layers as a mapping function $H(x)$, and the authors believe that the degradation problem arises because the training of $H(x)$ becomes extremely difficult as the layers are superimposed. If this part of the work can be superimposed in steps, that is, fitting a residual function $F(x) = H(x) - x$, then the objective function can be derived as $H(x) = F(x) + x$. Since $F(x)$ and x are the output and input respectively, if $F(x)$ can be found correctly, then $F(x)$ can be brought in as x' to find the deeper $F'(x)$ to

achieve effective hierarchical accumulation without the degradation problem encountered when training deep convolutional networks in a single pass. In the concrete implementation, the authors added a batch normalization (BN) layer after each convolutional layer and before the activation layer instead of the dropout method to reduce overfitting. Due to the excellent stackability of Resnet, the authors incrementally increase the network level from 18 to 152 layers and obtain better results than 18 layers on the ImageNet test set.

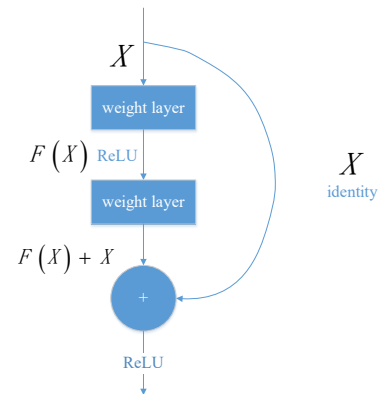


Figure 2. Diagram of skip connection.

Huang, et al. (2017) et al. proposed the densely connected network (DenseNet). The core idea of DenseNet is to connect the output of each layer to all subsequent layers to form a densely connected structure. This connection enhances feature propagation, improves the parameter utilization of the network, and reduces the training cost.

EfficientNet [25] is a CNN structure optimized based on neural network search techniques (NAS). The main contribution of EfficientNet is the introduction of a balanced network expansion strategy that improves performance by adjusting the depth, width, and resolution of the network. EfficientNet achieves state-of-the-art performance on several image classification tasks while having a low number of parameters and computational costs.

Although the classical models mentioned above and many current CNNs perform better on image classification tasks, there are some drawbacks compared to GCNs:

- CNNs are mainly designed for 2D grid structures (e.g., images) and cannot directly deal with non-Euclidean structured data [26]. Because discretized convolutions are only defined for regular domains [27]. However, the graph convolution of GCN can handle data with a non-Euclidean structure [28].
- CNNs ignore the relationships between nodes. When processing images usually only focus on local relationships between neighboring nodes and ignore the connections between more distant nodes. Thus, for

images with globally related nodes, CNNs may not be suitable for processing.

- In CNNs, convolution operations are usually used to aggregate spatial information. However, the convolution operation has high computational complexity. When the depth of the network increases, the computation and parameters of convolution become difficult to control [29].
- Although CNNs excel in dealing with two-dimensional grid structures such as image data, GCNs can better capture the relationships [30] between nodes and local features for more general graph-structured data, especially when the node relationships and attribute features are more complex, providing stronger modeling and representation capabilities.

2.2. Graph Neural Networks

For datasets containing images, traditional machine learning methods first preprocess the graph structure data [31], mapping the graph structure information to a simple representation such as a high or low-dimensional feature vector. This pre-processing stage may add to the image noise by losing the topological dependency of the information of the graph nodes [32].

Graph neural Networks (GNN) [32] are based on an information diffusion mechanism. Its appearance extends existing neural network methods for processing the data represented in graph domains. GNN pre-processing differs from traditional deep learning models in that it requires the conversion of images into node and edge representations.

For large-size images or images with deep structure, certain GNN models face three problems: neighbor explosion, node dependency, and over-smoothing [33]. In addition to the graph itself, these problems are attributed to the design of a multilayer GNNs framework [33]. To address these problems, [33] proposed the Ripple Walk Training (RWT) method for deep and large GNNs. general subgraph-based training framework RWT does not train directly on the whole picture but takes subgraphs from the whole picture and constructs small batches for training. Their proposed complete GNN is based on small-batch gradient updates. By computing small batches of gradients within subgraphs, subgraphs with acceptable sizes can avoid neighbor explosion altogether. Also, the gradients do not depend on nodes outside the subgraph, which solves the node dependency at the subgraph level. Aggregation between subgraphs usually occurs accidentally. However, propagation-aggregation occurs in the subgraphs, so the over-smoothing problem can be solved.

The classical models of GNN are GCN, GraphSAGE, and GAT.

GraphSAGE [34] is a graph neural network algorithm proposed in 2017, which solves the limitations of GCN networks: GCN training requires the adjacency matrix of the entire graph, depends on the specific graph structure, and can generally only be used in transductive learning.

GraphSAGE uses a multi-layer aggregation function, where each layer aggregates the information of nodes and their neighbors to obtain the feature vector of the next layer. GraphSAGE employs the neighborhood information of nodes and does not depend on the global graph structure. An innovative improvement of GraphSAGE is to use node features to learn an embedding function that enables invisible nodes to generate embeddings.

GraphSAGE has good performance in both unsupervised and supervised learning.

GAT [35] introduces what is essentially a single-layer pre-feedback neural network with an attention mechanism that allows the model to learn itself. This mechanism is performed by adding a model learnable coefficient to each edge and performing node feature fusion with an attention coefficient α . This allows the process of convolutional fusion feature to adjust the model parameters according to the task and become adaptive for better results. The formula for the Attention mechanism is defined as

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}(\vec{a}^T [W\vec{h}_i || W\vec{h}_j])\right)}{\sum_{k \in N_i} \exp\left(\text{LeakyReLU}(\vec{a}^T [W\vec{h}_i || W\vec{h}_k])\right)}, \quad (1.)$$

where \vec{a} is the weight vector of the attention mechanism, α_{ij} is the attention coefficient between the i -th node and the j -th node, T is the transposition, $||$ is the concatenation operation, W is the weight matrix, and N_i is some neighbor nodes of i -th node. After getting the α_{ij} of each edge, the node feature of the i -th point after the fusion of attention can be expressed as the following formula, which is essentially a weighted feature summation process, except that the weight coefficients in each fusion are learned with the model training, and finally after a nonlinear activation function.

$$\vec{h}'_i = \sigma\left(\sum_{j \in N_i} \alpha_{ij} W\vec{h}_j\right). \quad (2.)$$

On this basis, to make the attention mechanism more scalable, the defined multi-head attention mechanism is used to calculate the attention weights and then contact them to get the node feature. When at the last layer of the network, the dimensionality of the node features output after contact is too large. To make the data reasonable, it was switched to accumulate and then average, and then output to do specific tasks. Experiments on the PPI dataset show that GAT has better performance than GraphSAGE. GNNs do not perform well on training sets with class imbalance. This is because, in the class imbalance node classification, Song, et al. (2022) found that compensating for sub-nodes that deviate from the class connectivity pattern can easily lead to sub-node false positives. Juan, et al. (2023) point out that increasing the participation of a few nodes in the propagation process is an effective solution. Their INS-GNN contains Self-supervised pre-training, Self-training, and Self-supervised edge augmentation. self-supervised pre-training focuses on the contribution of a few

nodes, allowing the contribution to model learning, the model does not favor the majority of nodes. This is done by randomly sampling the edges of the graph and masking a few nodes. Self-training aims to reduce the cost of semi-supervised learning. It enables unlabeled nodes to have labels by creating pseudo-labels, which facilitates the performance of two different numbers of nodes. Self-supervised edge augmentation aims to involve a few nodes more in information transfer so that the majority of nodes have less influence on the model. However, Self-training may introduce noise while expanding the dataset. However, Self-supervised learning may have an advancing effect on solving the problem of unstable performance of GNNs in settings with too few labeled nodes [38-40].

2.3. Graph Convolutional Networks

In terms of the convolution method, graph convolution networks can be divided into spectral-based and spatial-based. Spectral-based GCN is a convolutional method based on spectral theory and convolution theorem, which converts data from the spatial domain to the spectral domain for processing and has a solid theoretical foundation. In contrast, spatial-based GCN is a method that does not rely on the spectral convolution theory but defines the convolution operation directly on the space, which is very flexible. Compared with spatial-based graph convolution methods, spectral-based graph convolution is currently less commonly used. Kipf and Welling (2016) proposed spectral convolution, which is to transfer the filter of the convolution network to the Fourier domain simultaneously with the graph signal before processing. While Niepert, et al. (2016) proposed spatial-based convolution which allows the nodes in the graph to connect and build hierarchical structures in the spatial domain and then convolve.

Suppose the undirected graph $G = \{V, E, X\}$. v is the set of N graph nodes, $V = \{v_i\}_{i=1}^N$. E is the set of edges of the nodes. X_i denotes the matrix of features of node v_i . If X_i is a d -dimensional feature ($X_i \in \mathbb{R}^d$), then $X \in \mathbb{R}^{N \times d}$ denotes the matrix of all node features.

The propagation equation of the improved GCN [41] is

$$H^{(l+1)} = \sigma(\tilde{A}H^{(l)}W^{(l)}). \quad (3.)$$

Here, H^l is the node feature of the GCN layer l and σ refers to the activation function of each graph convolution layer. D is the diagonal matrix representing the degree of nodes and $D_{ii} = \sum_j A_{ij}$. A_{ij} refers to the edge-forming relationship of nodes i and j . If there is an edge between i and j , then $A_{ij} = 1$, otherwise $A_{ij} = 0$. $\hat{A} = [A_{ij}] \in \mathbb{R}^{N \times N}$ is a node-self-connected adjacency matrix, $\hat{A} = A + I_N$, and I_N is the unit matrix. $\tilde{A} = \tilde{D}^{-\frac{1}{2}}\hat{A}\tilde{D}^{-\frac{1}{2}}$ is a sparse matrix. W^l refers to the learnable parameters. The graph convolution layer has two stages: feature fusion and feature extraction. $\tilde{A}H^{(l)}$ is the aggregation phase, which serves to control the node properties of local neighbors to make them similar. $H^{(l)}W^{(l)}$ is the feature extraction phase, where common features between neighboring nodes can be

extracted after feature aggregation [32]. Before the improvement of the equation,

$$\tilde{A} = \tilde{D}^{-\frac{1}{2}}\hat{A}\tilde{D}^{-\frac{1}{2}} + I_N, \quad (4.)$$

where $\hat{A} = A$. The eigenvalues of the above equation are in a small range. Therefore, when used in deep neural network models, repeated application of this operator can lead to numerical instability, and gradient explosion or disappearance [41]. To alleviate this problem, Kipf and Welling (2016) let $\hat{A} = A + I_N$, and then

$$\tilde{A} = \tilde{D}^{-\frac{1}{2}}\hat{A}\tilde{D}^{-\frac{1}{2}}. \quad (5.)$$

3. Image Classification Based on Graph Convolutional Networks

3.1. Common datasets

Citeseer, Cora, and Pubmed [43, 44] are commonly used datasets for GCN image classification. The datasets contain sparse bag-of-words feature vectors for each document and a list of citation links between documents [41]. The following is a description of these datasets (Table 1):

Table 1. Description of the three common datasets.

Dataset	Number of nodes	Dimension of node features	Classes of nodes	Number of edges
Cora	2708	1433	7	5429
Citeseer	3312	3703	6	4732
Pubmed	19717	500	3	44338

Citeseer. Nodes in the CiteSeer dataset denote papers and edges denote citation relationships between papers. So, this dataset contains 3312 papers in 6 categories in the field of computer science, represented by 3312 nodes. The citation relationships between them are represented by 4732 edges. These nodes can all be classified into 6 categories and are all represented by 3703-dimensional feature descriptors.

Cora. This dataset contains 2708 papers on the field of machine learning in 7 categories, which can also be represented by 2708, nodes in one of the 7 categories. These nodes have 5429 edges. Each node has a 1433-dimensional feature descriptor.

Pubmed. This dataset contains 19717 papers on the field of biomedicine in 3 categories. Each paper is represented by a bag-of-words model, and the edges between nodes indicate the co-citation relationships between papers. It can also be said that the dataset network contains 19717 nodes and 44338 edges, each represented by a 500-dimensional feature descriptor.

3.2. Graph Convolutional Networks Models

However, there are still some challenges and limitations of the original graph convolutional networks in image classification. One of them is the problem of computational efficiency because GCNs need to perform aggregation operations on the whole graph, resulting in high computational complexity [45]. In addition, GCNs may encounter the problem of memory limitation when dealing with large-scale image data because of the need to store and process a large number of nodes and edges [46]. Further, the original GCN can capture information only about immediate neighbors with one layer of convolution. When multiple GCN layers are stacked, information about larger neighborhoods is integrated [47]. This is because the graph convolution layers of GCNs can be considered low-pass filters [48], and this property causes the signal to become smoother, which is an inherent advantage of GCNs; however, if the number of GCN layers is large, performing this signal smoothing operation multiple times will make the signal converge, which loses the diversity of node features, which is the over-smoothing problem [49].

Too many graph convolution layers may cause the over-smoothing problem, however, there are image classification studies that solve this problem to some extent [50-52], allowing the model to extract deep-level features.

SelfSAGCN [50] overcomes the overfitting problem and the over-smoothing problem by combining feature aggregation and semantic alignment. SelfSAGCN first applies feature aggregation to extract semantic information from the labeled nodes layer by layer, which does not suffer from the over-smoothing phenomenon. The core idea of selfSAGCN is that the node features obtained from the semantic and graph structure should be consistent when the categories are the same. This is not affected by the over-smoothing phenomenon. The unlabeled node features obtained from graph aggregation are aligned with semantic features by semantic alignment techniques to find additional supervisory information. This improves the performance of the model and enhances the identifiability of node features. The semantic alignment of selfSAGCN is based on the central similarity of homogeneous class information, which enables the model to transfer relevant features to unlabeled data after learning semantics from the labeled data. Yang, et al. (2021) additionally utilized central similarity optimization to align node features obtained from semantic and graph structure aspects, respectively features are aligned, which has a significant effect on mitigating over-smoothing. Also, the central similarity of labeled and unlabeled nodes can provide additional supervised information, which further improves the classification accuracy of unlabeled nodes. Moreover, they use the pseudo-labeling trick for unlabeled data and also suppress the noise using the practice of updating the centers. It is experimentally confirmed that selfSAGCN has better performance on different datasets even when the labeled nodes are severely limited. This indicates that the overfitting problem does not affect it too much. Even if the number of layers is increased to 16, selfSAGCN can mitigate the over-smoothing problem.

Pan, et al. (2022) proposed that no deep GCN model has been used for medical diagnosis because of the problem of over-smoothing. To overcome the over-smoothing problem, they used a snowball GCN module to build a multiscale convolutional module. The snowball GCN [53] is a densely connected graph network that can connect multiscale features. This graph network can superimpose all the learned features as input to subsequent hidden layers. This network also overcomes the gradient vanishing problem and reduces the number of parameters, etc. The key to solving the over-smoothing problem in Luan, et al. (2019) is that they define the graph convolution of a spectral filter as the product of a block Krylov matrix and a specific form of learnable parameter matrix. The formula $K_m(A, B) := [B, AB, \dots, A^{m-1}B]$ for the block Krylov matrix comes from the transformation of S -span of S -span of $\{X_k\}_{k=1}^m$ and $K_m^S(A, B)$. They stated that it is difficult to apply the block Krylov method directly to the GCN, so they developed the snowball and the truncated Krylov. For $output = softmax(L^p C W_c)$, if $p = 1$ and $L^p = L$, then the snowball implementation maps back to the Fourier basis of the graph, thus achieving a "snowball" The Adaptive multi-channel fusion GCN implemented by Pan, et al. (2022) also contains the channel common convolution module, which solves the problem of extracting the nodes in a particular channel embedding and the common information shared between channels. As shown in Figure 3, the output of the channel common convolution module is represented by the equation

$$H_c = \alpha H_c + \beta H_{cfH_0} + \gamma H_{cp}, \quad (6)$$

where, α, β, γ are hyperparameters that measure the importance of the common convolution, respectively. The multi-channel attention they introduce can fuse the outputs of different channels and the features of each channel to integrate the learned embeddings for prediction. The ablation experiments demonstrate that their proposed MAMF-GCN has strong robustness and high accuracy.

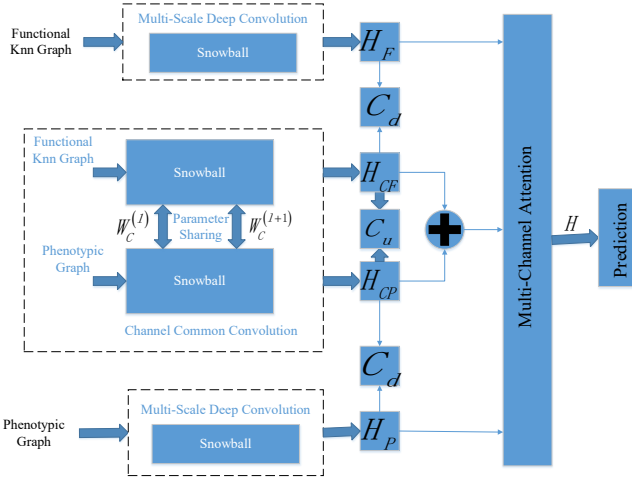


Figure 3. The structure of adaptive multi-channel fusion GCN.

The prominent innovation of NSCGCN [52] is to overcome the over-smoothing problem using feature fusion based on the node-self-convolution algorithm and to preserve the spatial structure information of the original feature graph using the feature reconstruction algorithm. The innovative point of the node-self-convolution algorithm is that the input undirected graph G^l retains only the node-self-connected degree matrix I . The result of the convolution of the input graph node features X is given by

$$Z^{(l+1)} = D^{-\frac{1}{2}} I D^{-\frac{1}{2}} X^{(l)} W^{(l)}, \quad (7.)$$

where $W^{(l)}$ is the learnable parameter. A new graph structure G^{l+1} is then obtained by combining the original adjacency matrix of the undirected graph G^l with $Z^{(l+1)}$ to regain a new graph structure $G^{(l+1)}$. The feature reconstruction algorithm is based on the image neighborhood structure, which converts the image into graphically structured data and has better classification performance than the down-sampling approach. Although NSCGCN also has outstanding performance, it also suffers from overfitting. And the feature reconstruction stage is not adaptive and does not achieve the best structure. For the overfitting phenomenon due to an insufficient amount of data, semantic alignment techniques can be considered to allow labeled nodes to guide unlabeled nodes [50].

For the problem of large memory consumption of graph convolution operations, Bi-GCN [54] provides a solution. Bi-GCN has two innovative approaches, one is to perform binarized network parameters and input node feature operations in the feature extraction phase, which can theoretically reduce memory consumption by up to 30 times and improve inference speed by up to 47 times, and the other is to design a new backpropagation method to accommodate binarized weights. Wang, et al. (2021)

showed that it is not feasible to down-sample the original image to the right size and compress the input graph data and GNN model to reduce the memory consumption of the model. They state that this is because of two ways of down-sampling, one, neighborhood sampling will result in "neighborhood explosion" when the number of graph convolution layers increases, i.e., many neighborhoods are making it difficult to train. Second, although graph sampling can avoid "neighborhood explosion," it does not guarantee that each node will be sampled once throughout the training or inference process. For compressed input images, it is difficult to manipulate because the original graph is small. For compressed GNN models, the relationship between the compression rate and the accuracy of the GNN model needs to be carefully designed to ensure that both nodes in the high-dimensional semantic space and nodes in the low-dimensional feature space can be characterized. However, this approach is more difficult. The core idea of Bi-GCN is to binarize the network parameters (e.g., weights) and input node features in the feature extraction phase, while not operating in this way in the feature aggregation phase. In addition, the original matrix multiplication of the graph convolution operation is modified to a binary addition operation. Another core idea is to binarize the node features by separating them and assigning attention weights to each node. By deploying additional parameters, the model can remain effective to learn more useful information. In theory, Bi-GCN can reduce the memory consumption of network parameters and input data by an average of about 30 times and increase the inference speed by an average of about 47 times. However, Bi-GCN had not experimented based on solving the over-smoothing problem, i.e., Bi-GCN does not perform well at deeper layers.

Similar to CNNs, GCNs have a multilayer structure [13], where each layer extracts higher-level features by aggregating features from neighboring nodes and applying a nonlinear activation function. This allows GCNs to take full advantage of the topology of the graph and thus better capture the relationships between nodes. However, applying GCNs may not be able to adaptively extract the most relevant information between topologies or node features. Xu, et al. (2021) proposed a multiscale skeleton adaptive weighted graph convolution network (MS-AWGCN) for skeleton-based human action recognition in IoT. MS-AWGCN solves the problem of learning potential graph topology in an adaptive extraction style by an adaptive information aggregation strategy to weight information from different sampling strategies more efficiently. The adaptive weighted graph convolution block formulation of MS-AWGCN is as follows,

$$f_{out} = \sum_k^K \alpha W_k f_{in} (A_k + L_k). \quad (8.)$$

where W_k denotes the weight matrix, L_k is learnable, and f_{in} is the node feature of the input.

Relative to the formulation of [56] graph self-learning module

$$f_{out} = \sum_k^K W_k f_{in}(L_k), \quad (9.)$$

MS-AWGCN introduces weights α initialized to 1 that control the importance of different sampling strategies and α normalized adjacency matrix A_k to make learning graph topology more efficient. MS-AWGCN also contains an attention mechanism, and the algorithm for the mechanism is SE-Net.

The core idea of the SE-Net [57] algorithm is to learn to use global information to selectively emphasize information-rich features and suppress less useful features. The algorithm allows the network to recalibrate features. Since the SE block is relatively simple, it has been applied to many CNNs to improve the performance of the model [58-61]. Although the SE block is currently less used on GCNs, it may be an option to consider from the perspective of incorporating attention mechanisms without adding too much to the model complexity.

4. Conclusion

This paper focuses on graph convolutional networks for image classification. Firstly, some classical models of convolutional neural networks are discussed, and their advantages and limitations on image classification tasks are pointed out. Then, graph neural networks are introduced, summarizing the current status of graph neural networks that perform poorly on training sets with class imbalance. And the solution of introducing self-supervised learning techniques is summarized. Finally, graph convolutional networks are introduced. I summarize some graph convolutional network models that solve the over-smoothing problem and explain the reasons for solving the problem. The model that solves the high memory consumption of graph convolution operation is also summarized, and its advantages and disadvantages are analyzed. And based on the analysis of successful cases, the self-attentive module is suggested to be introduced into graph convolutional networks. This paper does not analyze the over-smoothing problem and the high memory consumption problem of graph convolutional networks in more detail. In future work, I will analyze more literature and summarize more methods.

Graph convolutional networks have potential and wide application prospects in image classification tasks. With the continuous development and improvement of technology, it is believed that GCNs will play a greater role in image classification and other graph data fields.

References

- [1] J. M. Herzog and V. Sick, "Design of a line-of-sight fluorescence-based imaging diagnostic for classification of microbe species," *Measurement Science and Technology*, vol. 34, no. 9, Sep 2023, Art no. 095703.
- [2] M. Castillo-Cara, R. Talla-Chumpitaz, R. Garcia-Castro, and L. Orozco-Barbosa, "TINTO: Converting Tidy Data into image for classification with 2-Dimensional Convolutional Neural Networks," *SoftwareX*, vol. 22, May 2023, Art no. 101391.
- [3] A. Husain and V. P. Vishvakarma, "RES-KELM fusion model based on non-iterative deterministic learning classifier for classification of Covid19 chest X-ray images," *Journal of Intelligent Systems*, vol. 32, no. 1, Jun 2023, Art no. 20220235.
- [4] M. Hida *et al.*, "Development of Hallux Valgus Classification Using Digital Foot Images with Machine Learning," *Life-Basel*, vol. 13, no. 5, May 2023, Art no. 1146.
- [5] 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.
- [6] Y. Zhang, "A Rule-Based Model for Bankruptcy Prediction Based on an Improved Genetic Ant Colony Algorithm," (in English), *Math. Probl. Eng.*, Article vol. 2013, 2013, Art no. 753251.
- [7] Q. H. Wang, J. W. Liu, and L. L. Zhang, "Study on the Classification of K-Nearest Neighbor Algorithm," *Journal of Xi'an Technological University*.
- [8] S. Wang, "Detection of Dendritic Spines using Wavelet Packet Entropy and Fuzzy Support Vector Machine," *CNS & Neurological Disorders - Drug Targets*, vol. 16, no. 2, pp. 116-121, 2017.
- [9] D. Zhao, W. Zou, and G. M. Sun, "A fast image classification algorithm using Support Vector Machine," in *2010 2nd International Conference on Computer Technology and Development*.
- [10] Y. Zhang, "Pathological brain detection in MRI scanning by wavelet packet Tsallis entropy and fuzzy support vector machine," *SpringerPlus*, vol. 4, no. 1, 2015, Art no. 716.
- [11] X. Zheng and R. S. Cloutier, "A Review of Image Classification Algorithms in IoT," *EAI Endorsed Transactions on Internet of Things*, vol. 7, no. 28, pp. 1-11, 2022.
- [12] Y. Zhang *et al.*, "Unsupervised domain selective graph convolutional network for preoperative prediction of lymph node metastasis in gastric cancer," *Medical Image Analysis*, vol. 79, p. 102467, 2022/07/01/ 2022.
- [13] J. Zhou, J. X. Huang, Q. V. Hu, and L. He, "SK-GCN: Modeling Syntax and Knowledge via Graph Convolutional Network for aspect-level sentiment classification," *Knowledge-Based Systems*, vol. 205, p. 106292, 2020/10/12/ 2020.
- [14] M. Ghayekhloo and A. Nickabadi, "CLP-GCN: Confidence and label propagation applied to Graph Convolutional Networks," *Applied Soft Computing*, vol. 132, p. 109850, 2023/01/01/ 2023.
- [15] Y. Li *et al.*, "Identification of Mild Cognitive Impairment based on quadruple GCN model constructed with Multiple features from higher-order brain connectivity," *Expert Systems with Applications*, p. 120575, 2023/06/02/ 2023.
- [16] S. Zheng, L. Qiu, and F. Lan, "TSO-GCN: A Graph Convolutional Network approach for real-time and

- generalizable truss structural optimization," *Applied Soft Computing*, vol. 134, p. 110015, 2023/02/01/ 2023.
- [17] X. Zhu, C. Li, J. Guo, and S. Dietze, "CNIM-GCN: Consensus Neighbor Interaction-based Multi-channel Graph Convolutional Networks," *Expert Systems with Applications*, vol. 226, p. 120178, 2023/09/15/ 2023.
- [18] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in neural information processing systems*, vol. 25, no. 2, 2012.
- [19] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Computer Science*, 2014.
- [20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, and A. Rabinovich, "Going Deeper with Convolutions," *IEEE Computer Society*, 2014.
- [21] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *JMLR.org*, 2015.
- [22] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE*, 2016.
- [24] G. Huang, Z. Liu, L. Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [25] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," 2019.
- [26] X. Liu, Z. You, Y. He, S. Bi, and J. Wang, "Symmetry-Driven hyper feature GCN for skeleton-based gait recognition," *Pattern Recognition*, vol. 125, p. 108520, 2022/05/01/ 2022.
- [27] H. Jiang, P. Cao, M. Xu, J. Yang, and O. Zaiane, "Hi-GCN: A hierarchical graph convolution network for graph embedding learning of brain network and brain disorders prediction," *Computers in Biology and Medicine*, vol. 127, p. 104096, 2020/12/01/ 2020.
- [28] Y. Zhang, S. Satapathy, D. Guttery, J. Gorriz, and S. Wang, "Improved Breast Cancer Classification Through Combining Graph Convolutional Network and Convolutional Neural Network," *Information Processing and Management*, vol. 58, p. 102439, 01/18 2021.
- [29] Y. Zang, D. Yang, T. Liu, H. Li, S. Zhao, and Q. Liu, "SparseShift-GCN: High precision skeleton-based action recognition," *Pattern Recognition Letters*, vol. 153, pp. 136-143, 2022/01/01/ 2022.
- [30] S. Wang, V. Govindaraj, J. Gorriz, X. Zhang, and Y. Zhang, "Explainable diagnosis of secondary pulmonary tuberculosis by graph rank-based average pooling neural network," *Journal of Ambient Intelligence and Humanized Computing*, 03/11 2021.
- [31] S. Wang, "Advances in data preprocessing for biomedical data fusion: an overview of the methods, challenges, and prospects," *Information Fusion*, vol. 76, pp. 376-421, 2021.
- [32] 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.
- [33] Y. Liu, M. Zhang, C. Ma, B. Bai, and G. Yan, "Graph neural network," 2020.
- [34] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," 2017.
- [35] P. Velickovi, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," 2017.
- [36] J. Song, J. Park, and E. Yang, "TAM: Topology-Aware Margin Loss for Class-Imbalanced Node Classification," presented at the Proceedings of the 39th International Conference on Machine Learning, Proceedings of Machine Learning Research, 2022.
- [37] X. Juan, F. Zhou, W. Wang, W. Jin, J. Tang, and X. Wang, "INS-GNN: Improving graph imbalance learning with self-supervision," *Information Sciences*, vol. 637, p. 118935, 2023/08/01/ 2023.
- [38] Q. Q. Zhao, H. F. Ma, L. J. Guo, and Z. X. Li, "Hierarchical attention network for attributed community detection of joint representation," *NEURAL COMPUTING & APPLICATIONS*, vol. 34, no. 7, pp. 5587-5601, APR 2022.
- [39] Z. Yang, Y. Yan, H. Gan, J. Zhao, and Z. Ye, "A safe semi-supervised graph convolution network," *Mathematical Biosciences and Engineering*, vol. 19, no. 12, pp. 12677-12692, 2022.
- [40] Z. Zhou, J. Shi, S. Zhang, Z. Huang, and Q. Li, "Effective stabilized self-training on few-labeled graph data," *Information Sciences*, vol. 631, pp. 369-384, 2023/06/01/ 2023.
- [41] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," 2016.
- [42] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning Convolutional Neural Networks for Graphs," *JMLR.org*, 2016.
- [43] B. Jiang, Z. Zhang, D. Lin, J. Tang, and B. Luo, "Semi-Supervised Learning With Graph Learning-Convolutional Networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [44] Prithviraj *et al.*, "Collective Classification in Network Data," *Ai Magazine*, 2008.
- [45] T. Altaf, X. Wang, W. Ni, R. P. Liu, and R. Braun, "NE-GConv: A lightweight node edge graph convolutional network for intrusion detection," *Computers & Security*, vol. 130, Jul 2023, Art no. 103285.
- [46] R. Corrias, M. Gjoreski, and M. Langheinrich, "Exploring Transformer and Graph Convolutional Networks for Human Mobility Modeling," *Sensors*, vol. 23, no. 10, May 2023, Art no. 4803.
- [47] L. Yao, C. Mao, and Y. Luo, "Graph Convolutional Networks for Text Classification," 2019, pp. 7370-7377.
- [48] N. T. Hoang and T. Maehara, "Revisiting Graph Neural Networks: All We Have is Low-Pass Filters," 2019.
- [49] Q. Li, Z. Han, and X. M. Wu, "Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning," 2018.
- [50] X. Yang, C. Deng, Z. Dang, K. Wei, and J. Yan, "Self-SAGCN: Self-Supervised Semantic Alignment for Graph Convolution Network," in *Computer Vision and Pattern Recognition*, 2021.
- [51] J. Pan, H. Lin, Y. Dong, Y. Wang, and Y. Ji, "MAMF-GCN: Multi-scale adaptive multi-channel fusion deep graph convolutional network for predicting mental disorder," *Computers in Biology and Medicine*, vol. 148, p. 105823, 2022/09/01/ 2022.
- [52] C. Tang, C. Hu, J. Sun, S.-H. Wang, and Y.-D. Zhang, "NSCGCN: A novel deep GCN model to diagnosis

- COVID-19," *Computers in Biology and Medicine*, vol. 150, p. 106151, 2022/11/01/ 2022.
- [53] S. Luan, M. Zhao, X. W. Chang, and D. Precup, "Break the Ceiling: Stronger Multi-scale Deep Graph Convolutional Networks," ed, 2019.
- [54] J. Wang, Y. Wang, Z. Yang, L. Yang, and Y. Guo, "Bi-GCN: Binary Graph Convolutional Network," in *Computer Vision and Pattern Recognition*, 2021.
- [55] W. Xu, M. Wu, J. Zhu, and M. Zhao, "Multi-scale skeleton adaptive weighted GCN for skeleton-based human action recognition in IoT," *Applied Soft Computing*, vol. 104, p. 107236, 2021/06/01/ 2021.
- [56] G. Zhu, L. Zhang, H. Li, P. Shen, and M. Bennamoun, "Topology-learnable Graph Convolution for Skeleton-based Action Recognition," *Pattern Recognition Letters*, vol. 135, 2020.
- [57] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 8, pp. 2011-2023, 2020.
- [58] N. Ettehad *et al.*, "Automated Multiclass Artifact Detection in Diffusion MRI Volumes via 3D Residual Squeeze-and-Excitation Convolutional Neural Networks," (in English), *Frontiers in Human Neuroscience*, Original Research vol. 16, 2022-March-30 2022.
- [59] Y. Chen, Z. Zhang, L. Zhong, T. Chen, J. Chen, and Y. Yu, "Three-Stream Convolutional Neural Network with Squeeze-and-Excitation Block for Near-Infrared Facial Expression Recognition," *Electronics*, vol. 8, no. 4, p. 385, 2019.
- [60] J. Chen *et al.*, "An Efficient Memristor-Based Circuit Implementation of Squeeze-and-Excitation Fully Convolutional Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 4, pp. 1779-1790, 2022.
- [61] G. A. Altuwajri, G. Muhammad, H. Altaheri, and M. Alsulaiman, "A Multi-Branch Convolutional Neural Network with Squeeze-and-Excitation Attention Blocks for EEG-Based Motor Imagery Signals Classification," *Diagnostics*, vol. 12, no. 4, p. 995, 2022.