Appendix 1 Supplementary methods

The network structure and parameter setting of Inception-v3 model

In this study, the classical Inception-v3 model is initially adopted (39). The input ROI is initially resized to 299×299 pixels. Next, the input image is processed through 6 convolutional layers and 1 max pooling layer in a sequential manner. Subsequently, the feature maps are fed into the Inception modules consecutively. Afterwards, the outputs of these modules are passed through an average pooling layer to filter feature information. Finally, the resulting feature vector is mapped to a two-dimensional vector using three fully connected layers, and the prediction results are generated as output. The cross-entropy function is utilized as the loss function for model training in this study, as illustrated in Eq. [5].

$$J\left(\theta;I_k,C_k\right)=-\frac{1}{K}\sum_{k=1}^{K}\left[C_k\log\left(P\left(Y_k=C_k\mid I_k,\theta\right)\right)+\left(1-C_k\right)\log\left(P\left(Y_k=\left(1-C_k\right)\mid I_k,\theta\right)\right)\right]\qquad[5]$$

Among them, I_k and C_k represent the k th region of interest image and its corresponding label, θ represents the weights and biases estimated in the model, and K represents the total number of training samples.

The network structure and parameter setting of graph attention network model

Firstly, select the 5 slices from the registered 3D voxel MR slice sequence, which are located in the middle and set the ROI as 128×128. Construct a 3D voxel data with a dimension of 5×128×128, totaling 606 cases. Then, divide each case's 5×128×128 3D voxel into n small patches, replacing the original complete voxel with these n patches. It is assumed the value of n is 5 and the size of each patch is 16×16. In subsequent experiments, the effects of different patch sizes and numbers are further analyzed. Each type of patch is flattened into a 1-dimensional vector, forming a feature vector with a dimension of 1×1,280. After passing through a fully connected layer for each of these 5 feature vectors, 5 feature vectors with dimensions of 1×128 are obtained. These vectors represent the features of the voxel and serve as input nodes for the graph convolutional neural network. The entire data preprocessing process is shown in *Figure S1*.

By calculating the differences between patches in the hyperbilirubinemia and normal control groups, as well as within each group itself, we select patches that maximize inter-class differences and minimize intra-class differences. Finally, we choose the most representative k classes as the new data samples from the n classes of patches. The process of calculating the difference between patches is shown in *Figure S2*. We take 3 samples from the hyperbilirubinemia group and 3 samples from the normal control group, and provide explanations based on the extraction of 3 classes of patches from them. Among them, classes a, b, and c represent patches located in three different positions within the hyperbilirubinemia group, while classes d, e, and f represent regions in the normal control group corresponding to classes a, b, and c, respectively. For patch "a1" in hyperbilirubinemia sample 1, it is necessary to calculate the differences by comparing it with patches "a2" and "a3" in the same position from hyperbilirubinemia sample 2 and sample 3, respectively. By adding up the difference values from these 2 comparisons, the calculation of intra-class differences is completed. Similarly, for patch "a1", it is also necessary to calculate the differences by comparing it with patches "d1", "d2", and "d3" in the same position from normal control sample 1, sample 2, and sample 3, respectively. By adding up the difference values from these 3 comparisons, the calculation of inter-class

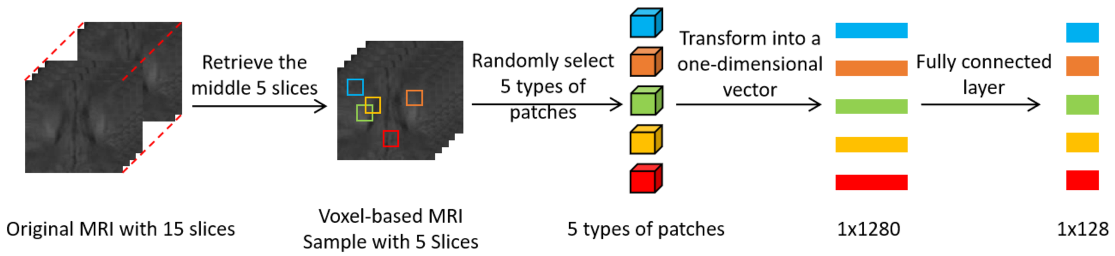
Retrieve the middle 5 slices
Randomly select 5 types of patches
Transform into a one-dimensional vector
Fully connected layer
Original MRI with 15 slices
Voxel-based MRI Sample with 5 Slices
5 types of patches
1x1280
1x128

Figure S1 Graph data preprocessing process.

https://dx.doi.org/10.21037/qims-24-1050

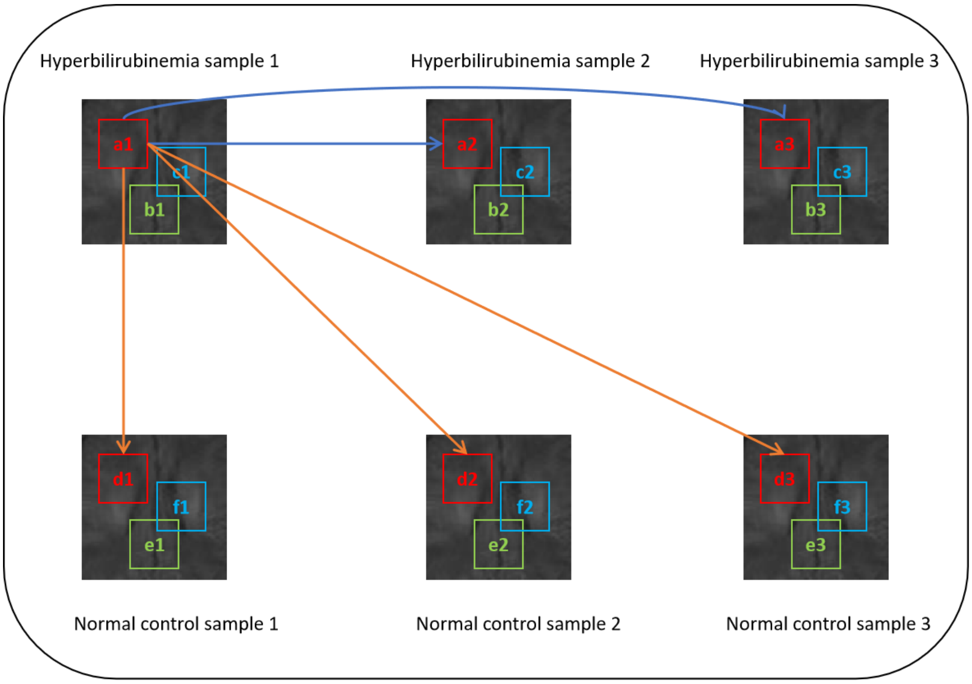

Figure S2 Calculate the difference between different patches.

differences is completed. In this way, the same calculation procedure is applied to compare the b-class patch with the e-class patch, as well as the c-class patch with the f-class patch. Through this process, the inter- and intra-class differences for each category of patches can be obtained.

The feature vectors are obtained by preprocessing the extracted 3D patches. Afterwards, the calculation of cosine similarity is performed to measure the differences and similarities between and within classes. The calculation of the dot product and the norm of vectors is required. The dot product represents the sum of the element-wise multiplication of two corresponding feature vectors. The inner product (*A*•*B*) of 2 n-dimensional vectors $A = (a_1, a_2, ..., a_n)$ and $B = (b_1, b_2, ..., b_n)$ can be computed using Eq. [6].

$$A \bullet B = \sum_{i=1}^{n} a_i \times b_i \qquad [6]$$

The cosine similarity $Similarity(S_1, S_2)$ between 2 vectors can be calculated using Eq. [7] after computing their inner product and norms.

$$Similarity(A,B) = \frac{A \bullet B}{|A| \times |B|} = \frac{\sum_{i=1}^{n} a_i \times b_i}{\sqrt{a_1^2 + a_2^2 + \cdots + a_n^2} \times \sqrt{b_1^2 + b_2^2 + \cdots + b_n^2}} \qquad [7]$$

Assuming there are m samples in the hyperbilirubinemia group and m samples in the normal control group, we obtain feature vectors for 2 groups of patches located at the same positions within these 2 classes. These feature vectors, obtained after preprocessing, are labeled as L_1, L_2… L_m and K_1, and K_2…K_m respectively. Initially, calculate the cumulative differences between a particular sample patch (*L1*) and the various samples at the same position from the other group (K_1, K_2,...,K_m), denoting it as difference1. Simultaneously, calculate the cumulative differences between L1 and the remaining samples within the same category (L_1, L_2,...,L_m), denoting it as difference2. Eqs. [8] and [9] represent the formulas for calculating the inter- and intra-class differences, respectively.

$$difference_1 = \frac{1}{m} \times \sum_{i=1}^{m} Similarity(L_1, K_i) \qquad [8]$$

$$difference_2 = \frac{1}{m-1} \times \sum_{i=1}^{m-1} Similarity(L_1, L_i) \qquad [9]$$

To ensure the selected patches effectively represent their respective categories and clearly exhibit their differences from the corresponding categories, they should simultaneously satisfy the principles of significant inter-category differences and minimal intra-category differences. The measurement of each class of patches is calculated by dividing the inter-category differences by the intra-category

https://dx.doi.org/10.21037/qims-24-1050

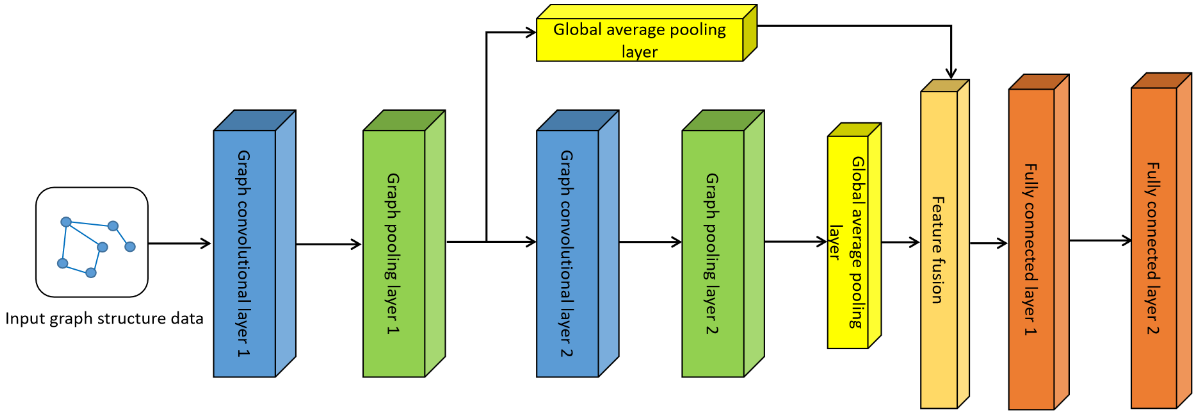
Global average pooling layer
Input graph structure data
Graph convolutional layer 1
Graph pooling layer 1
Graph convolutional layer 2
Graph pooling layer 2
Global average pooling layer
Feature fusion
Fully connected layer 1
Fully connected layer 2

Figure S3 Structure diagram of graph convolutional neural network.

differences, denoted as η and represented by Eq. [10].

$$\eta = \frac{difference_1}{difference_2} \quad [10]$$

We select 25 types of 3D patches from ROI as nodes for the GCN network, with each node having a feature vector dimension of 1×128. For each node, we select the 5 types of patches with the minimum discrepancy within the same image as neighboring nodes, thus creating a graph-structured data. All samples in the dataset undergo the same process, resulting in a total of 606 examples of graph data. The entire image is used as the input for the GCN network, which outputs the classification result for this sample. The schematic diagram of the GCN network is shown in *Figure S3*. First, a graph pooling layer with a parameter set to 0.8 is applied. The top 80% of the most important feature vectors from the 25 types are selected, whereas the remaining nodes are discarded. Then, the remaining 20 nodes with 1×128 features undergo a global average pooling layer, and the resulting output is denoted as x_1. Next, x_1 is passed through graph convolutional layer 2, outputting node feature vectors with a dimension of 1×128. Subsequently, it is passed through another graph pooling layer with a parameter set to 0.9. As a result, the number of nodes in the graph structure is reduced to 18, with node features dimensions of 1×128. The output is then subjected to another global average pooling layer, and the resulting output is denoted as x_2. The feature vectors x_1 and x_2 obtained from the 2 global average pooling layers are then combined, resulting in a 1×128 feature vector that effectively represents the feature information of the patch. This combined vector is then passed through a fully connected layer 1 to reduce its dimension, resulting in a 1×64 feature vector. Finally, the 1×64 feature vector is mapped to 2 nodes, and the classification result for this sample is obtained.

During the training of data using the graph convolutional neural network, we introduce the graph attention mechanism. For each type of patch in the graph structure, the weight S_i in all patch categories can be represented by calculating the cosine similarity between its own feature vector X_i and the global feature vector X. Higher values of cosine similarity indicate a stronger similarity between the feature vector X_i and the global feature vector, whereas lower values indicate a weaker similarity. The calculation for this is specified in Eq. [11].

$$S_i = \frac{X_i \bullet X}{|X_i| \times |X|} \quad [11]$$

By calculating the cosine similarity between the feature vectors of each patch type and the global feature vector, we can select the top k patches with the highest similarity. Identifying the specific locations of these k patch types in the image enables us to obtain more precise location information for neonatal hyperbilirubinemia in brain MRI. Furthermore, these selected k patch types can serve as representative and refined feature information for the sample data. By summing the feature vectors X_1, X_2, ..., X_k of these k patch types, we obtain a new feature vector V. The calculation formula is presented in Eq. [12].

$$V = \sum_{i=1}^{k} X_i \quad [12]$$

Extract k types of patches from each sample data and

https://dx.doi.org/10.21037/qims-24-1050

Table S1 Results of cosine similarity calculation

Patch type	η	Patch type	η
1	0.75	10	0.76
2	0.05	11	0.56
3	0.94*	12	0.25
4	0.42	13	0.14
5	0.90*	14	0.92*
6	0.49	15	0.88*
7	0.60	16	0.34
8	0.84*	17	0.77
9	0.02	18	0.76

*, the best results of the metrics. η, value of the cosine similarity. By calculating the cosine similarity between the feature vectors of these 18 patch types and the global feature vector, we identified the top 5 types with the highest values (i.e., types 3, 5, 8, 14, 15).

Table S2 Comparison of classification performance for different numbers of patch categories

Number of patches	AUC	ACC	SEN	SPE
15	0.57	0.56	0.55	0.58
20	0.60	0.59	0.54	0.65
25	0.66*	0.66*	0.65*	0.67*
30	0.64	0.63	0.60	0.66
35	0.62	0.58	0.63	0.58

*, the best results of the metrics. AUC, area under the curve; ACC, accuracy; SEN, sensibility; SPE, specificity.

Table S3 Comparison of classification performance for different patch sizes

Size of patch	AUC	ACC	SEN	SPE
8×8	0.57	0.56	0.56	0.58
16×16	0.69*	0.66*	0.68*	0.65*
24×24	0.58	0.57	0.60	0.55
32×32	0.60	0.58	0.58	0.61
40×40	0.55	0.55	0.52	0.57

*, the best results of the metrics. AUC, area under the curve; ACC, accuracy; SEN, sensibility; SPE, specificity.

compute the sum of their feature vectors. The resulting output will consist of 606 reconstructed feature vectors along with their respective labels. By using a supervised learning method and a logistic regression model, the task of classification and prediction of neonatal hyperbilirubinemia can be further completed.

Patch selection

We selected a subset of representative patches from the original image consisting of 18 patch types (originally 25 types, but reduced to 18 after two pooling layers). These selected patches served as feature vectors to represent the original samples. Subsequently, the attention mechanism of the graph CNN determined the weights associated with each node. By calculating the cosine similarity between the feature vector of each patch type and the global feature vector, we could identify the top 5 patch types with the highest similarity to the global feature vector. The results are presented in *Table S1*.

Patch ablation experiment

Different numbers and sizes of patches extracted during the patch extraction process may impact the classification performance of the model. Therefore, we conducted a comparative analysis through experiments. The experimental setup involved fixing the size of patches at 16×16 and selecting 25 patch categories. Initially, while maintaining a constant patch size, we analyzed the performance by manipulating the number of patch categories. The comparison of classification performance for varying numbers of patch categories is shown in *Table S2*. In this instance, the chosen numbers of patch categories were 15, 20, 25, 30, and 35.

Subsequently, the influence of patch size on the classification performance of graph CNNs was investigated while keeping the number of patch categories fixed at 25. Patch sizes of 8×8, 16×16, 24×24, 32×32, and 40×40 were chosen for evaluation. *Table S3* displays the prediction and classification results of the models for different patch sizes.

The experimental results demonstrate that variations in the number and size of patches have different effects on the

https://dx.doi.org/10.21037/qims-24-1050

classification performance of graph CNNs. Comparative analysis of the experimental results indicates that selecting patches with dimensions of 16×16 and categorizing them into 25 classes enhances the experimental outcomes by mitigating noise interference and maximizing the inclusion of category information in the images.

References

1. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016:2818-26.

https://dx.doi.org/10.21037/qims-24-1050