# Diagnosing tuberculosis using graph neural network

Nguyen Thi Thu Hang[1,2,*], T.H. Nam[1,2,*], Bac Le[1,2]
[1]Faculty of Information Technology, University of Science, Ho Chi Minh City, Viet Nam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
Email: {18120027,18120473}@student.hcmus.edu.vn, lhbac@fit.hcmus.edu.vn

*Abstract*—**According to the World Health Organization (WHO), tuberculosis (TB) is the top disease deadly worldwide, especially in developing/underdeveloped countries, due to poverty and limited health resources. With severe effects on patient health and rapid spread, early screening for TB is a highly urgent task. Among the methods of diagnosing tuberculosis, chest X-ray images are often used as resources for clinical diagnosis because of their convenience and optimal cost. Currently, research on computer-aided diagnostics (CAD) systems uses machine learning to provide doctors with diagnostic, analytical, and disease-monitoring techniques. Recently, graph neural network has emerged as a research trend; works using GNN bring perfect accuracy in many fields. In this paper, a study is presented on a solution to automatically diagnose tuberculosis on X-ray images (CXR) using the graph neural network method. We classify the CRX dataset into two classes (TB and non-TB). We achieve encouraging results with the proposed model: the accuracy 99.33%, recall 99.07%, precision 99.63%, f1-score 99.35%, AUC 99.97%.**

*Index Terms*—**graph classification, graph neural network, tuberculosis screening**

## I. Introduction

Tuberculosis is a dangerous infectious disease at the top of the world. Mycobacterium virus is the main cause of tuberculosis disease. It is spread from person to person by droplets that contain the virus when coughed by an infected person [1]. Mycobacterium viruses can cause infections anywhere in the patient's body. Still, most detections are in the lungs, which contain high levels of oxygen, consistent with their aerobic properties, and suitable for their growth and spread. The disease can spread quickly, difficult to control in the community. Not only that, the disease causes many dangerous complications to the health and life of the patient. Early detection of the disease is important in treating it and controlling its ability to spread widely in society. [1]

Currently, there are many methods of diagnosing tuberculosis with excellent results; the most common practice is a diagnosis based on CXR. However, the manual diagnosis based on CXR requires high expertise, a long time, and is too subjective. Recently, the research direction of applying machine learning in CAD systems has been increasingly interested and focused. The driving force behind the development of CAD systems was that (i) objective, and subjective factors could influence manual diagnostic results, and (ii) CAD could extract information from medical data, which assists doctors in diagnosing and detecting diseases more quickly and accurately. Recently, research related to graph data has been of increasing interest because of the effectiveness of data modeling. Graph structures can represent data most naturally based on their unique structure, including nodes representing objects and edges representing links between objects. As we know, the image consists of a set of pixels that carry a grid structure. But from another view, we can view the image as an instance of the graph structure. Based on that, the problem of diagnosing pulmonary tuberculosis on X-ray images, essentially classifying X-ray images into two classes of tuberculosis (TB) and non-tuberculosis (non-TB), has become a graph classification task.

In this paper, we will propose a method to automatically diagnose tuberculosis on CXR using the power of graph neural networks. Specifically, our method is to transform the image under the grid structure into a graph structure so that we can take advantage of the power of the graph classification algorithm based on isomorphism (Graph Isomorphism Network). Additionally, experiments on the CXR dataset evaluate the performance of our proposed model. The results prove that our proposed method gives accurate results with optimal inference time. Scientific contributions of the study: (i) propose a new CAD method to automatically diagnose TB on CXR (ii) the proposed architecture achieves optimization in terms of memory space and inference time. The article is structured as follows: Part 2 provides the related knowledge and related works; Part 3: presents our proposed methodology and architecture; Part 4: illustrates the implementation and analysis of the obtained results. And finally, part 5 presents the conclusions and directions for further research.

## II. Related works

Over the years, along with the campaign against tuberculosis of medical organizations, many research works on applying machine learning to CAD systems specifically for tuberculosis have been published.

In 2021, Alawi et al. [2] proposed a new CNN architecture to diagnose tuberculosis through CXR classification. The proposed model has the main structure of 7 convolutional layers 7×7 with the activation function in each layer being ReLU and ending with three fully connected(FC) layers and two layers of dropout. Their dataset contains 7000 images, including 3500 TB cases and 3500 non-TB cases. As a result, the classification accuracy they achieved was 98.72%.

The authors with * have equal contribution.

Nafisah et al. [3] and Rahman et al. [4] have proposed a new pipeline that starts with lung segmentation and then uses classifiers with input is the segmented CXR. Nafisah et al [3] performed experiments on a dataset of 1098 images including 692 TB cases and 406 non-TB cases, achieving classification accuracy of 99.2% with EfficientNetB3 classifier. Rahman et al. [4] performed experiments on a dataset of 7000 images, of which 3500 TB cases and 3500 non-TB cases achieved a classification accuracy of 98.6% with the DenseNet201 classifier.

Not only limited to the methods using the outstanding CNN network, a new branch, Vision Transformers (Vit), has also been researched and applied. With the work to improve the performance of the classification model, supporting the diagnosis of tuberculosis, Linh T.Duong et al. published a paper [5], in which the author proposed to build from 2 main blocks: EfficientNet (Deep Convolutional Neural Network) and Vision Transformer (Self-attention) as the classification engines. They also use transfer learning algorithms to boost performance, with the highest accuracy of 97.72% on a data set of 28,840 images divided into three classes pneumonia, tuberculosis, and normal.

Researchers also exploit ensemble learning techniques. Pasa, F et al. [6] and Rajaraman, Sivaramakrishnan et al. [7] have published two papers presenting the application of ensemble learning technique in diagnosing tuberculosis on CRX. Specifically, Pasa, F et al. [6] have performed ensemble two CNN architectures, VGG16 and InceptV3, resulting in an accuracy of 95.8% experimentally on a dataset of 1098 images, including 1098 images. 692 TB cases and 406 non-TB cases. Rajaraman, Sivaramakrishnan et al [7] have implemented ensemble of two power architectures, Vit and CNN, combined with model transfer learning techniques, with accuracy of 90.57% experimentally on CheXpert CXR [8] and PadChest CRX [9] datasets.

Lu, Si-Yuan, et al. [10] proposed TBNet, which is a solution for applying graph neural networks in diagnosing tuberculosis through X-ray images. The main idea of Si-Yuan Lu et al. is to transform the dataset into a graph, with the corresponding nodes being the images, and the neighbors of each image will be connected to that image through edges. That edge is determined by a coefficient that indicates the similarity of the two images. The features of each node will be obtained through feature extraction from CNN pre-trained model. With this proposal, the method has achieved 98.93% accuracy.

## III. Preliminaries

### A. Graph

Graph is a discrete structure of the nodes and the edges connecting these nodes. A graph can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges. The edge pointing from node $u$ to node $v$ is denoted as $(u, v) \in \mathcal{E}$. The neighborhood of the node $u$ is defined as $\mathcal{N}(u) = \{v \in \mathcal{V} | (u, v) \in \mathcal{E}\}$.

Graph can also be represented as the adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$ ($n = |\mathcal{V}|$ is the total number of nodes). Each element $A_{uv}$ of which equals to 1 if $(u, v) \in \mathcal{V}$, equals to 0 if $(u, v) \notin \mathcal{V}$.

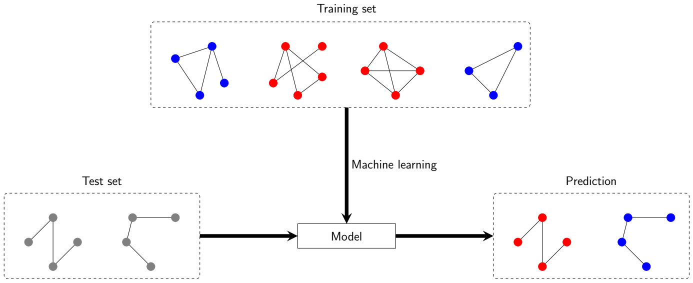


Fig. 1: The graph classification problem

Graphs may have node features $\mathbf{X} \in \mathbb{R}^{d \times n}$, and/or edge features $\mathbf{E} \in \mathbb{R}^{d' \times |\mathcal{E}|}$, where $d, d'$ are dimensions of the feature vectors. Each column vector $\mathbf{x}_u$ of $\mathbf{X}$ represents the feature vector of node $u$, each column vector $\mathbf{e}_{u,v}$ of $\mathbf{E}$ represents the feature vector of edge $(u, v)$.

### B. Types of graph

An undirected graph is a graph in which edges have no orientations, i.e. $(u, v) \in \mathcal{E} \Leftrightarrow (v, u) \in \mathcal{E}$. Otherwise, it is called directed graph.

A simple graph is a graph that allows at most one edge connecting a pair of nodes, and these edges do not connect any node to itself. Otherwise, it is called a multigraph.

In addition, based on different criteria, we can classify graphs in different ways. In this work, we only refer to undirecred simple graphs along with nodes features.

### C. Machine learning on graphs

In recent years, more and more graph data has been generated. It entails our natural need to exploit it effectively.

Machine learning is the process of extracting knowledge from data. The use of machine learning on graph data opens many interesting problems and results that people have not thought of or have not had an effective method for before. Part of the reason is that graph data is not friendly to machine learning models. It is difficult to represent a graph in Euclidean space as efficiently as other types of data. Fortunately, this area has recently received a lot of attention because of its innovative methods, outstanding results, and clear benefits.

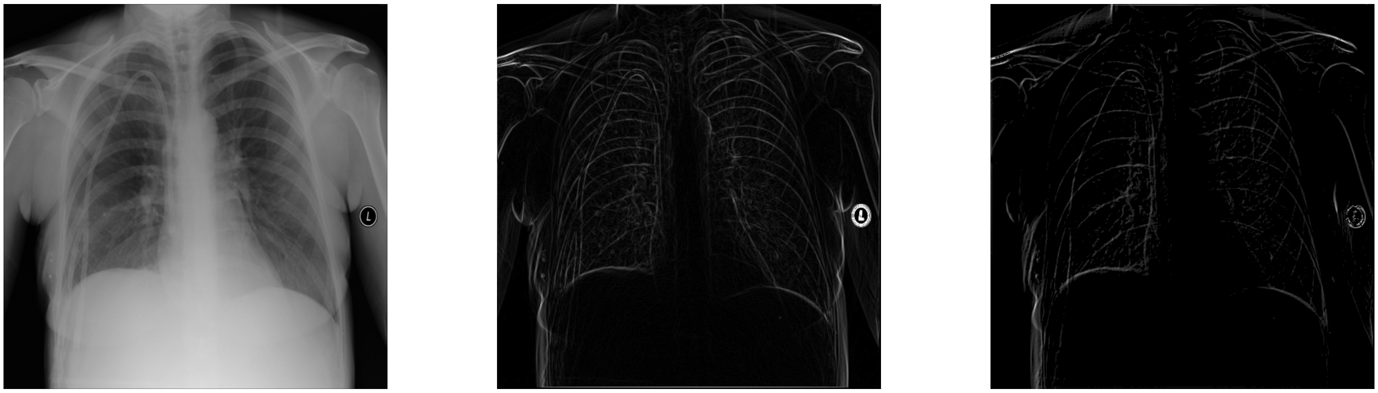
Graph machine learning tasks can generally be divided into three groups:

- **Node-level tasks**: node classificatoin, node regression, node clustering (community detection), ...
- **Edge-level tasks**: edge classification, link prediction, ...
- **Graph-level tasks**: graph classificatoin, graph regression, graph generation, graph matching, ...

Our work in this paper is a graph classification task (Fig. 1)

### D. Graph neural network

Graph neural network (GNN) is the name of a class of deep learning methods that perform learning on graph data. Given a graph $\mathcal{G}$ represented by nodes features $\mathbf{X} \in \mathbb{R}^{d \times n}$



(a) Original image

(b) Edge detection

(c) Selected pixels from the edge detection result image to construct the graph

Fig. 2: Illustration of the feature extraction process

and adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ , our objective is to learn a function $f$ to turn $\mathcal{G}$ into a meaningful embedding $\mathbf{Z} \in \mathbb{R}^{d \times n}$, i.e. $\mathbf{Z} = f(\mathcal{G})$. Which is used for downstream tasks.

Message-passing neural network (MPNN) [11] is a general framework of the most common GNN models. In MPNN, the node feature of $u$ at each iteration $k$, denoted as $\mathbf{x}_u^{(k)}$, will be updated by the message $\mathbf{m}_u^{(k)}$ aggregated from the neighborhood:

$$\mathbf{m}_u^{(k)} = \text{AGGREGATE}^{(k)}\left(\left\{\mathbf{x}_v^{(k)}, \forall v \in \mathcal{N}(u)\right\}\right) \tag{1}$$

$$\mathbf{x}_u^{(k+1)} = \text{UPDATE}^{(k)}\left(\mathbf{x}_u^{(k)}, \mathbf{m}_u^{(k)}\right) \tag{2}$$

Where UPDATE, AGGREGATE are any differentiable functions (neural networks for example). The initial state at $k = 0$ takes the input as $\mathbf{x}_u^{(0)} = \mathbf{x}_u \forall u \in \mathcal{V}$. After $K$ iterations of message passing, we obtain the desired output $\mathbf{z}_u = \mathbf{x}_u^{(K)} \forall u \in \mathcal{V}$. There are several instances of this framework with their own design of these functions, such as GCN [12], GAT [13]. In this work, we used the Graph Isomorphism Network (GIN) proposed by Xu et al. [14] with the following update equation:

$$\mathbf{x}_u^{(k+1)} = \text{MLP}^{(k)}\left((1 + \epsilon^{(k)}) \cdot \mathbf{x}_u^{(k)} + \sum_{v \in \mathcal{N}(u)} \mathbf{x}_v^{(k)}\right) \forall u \in \mathcal{V} \tag{3}$$

Or equivalently:

$$\mathbf{X}^{(k+1)} = \text{MLP}^{(k)}\left(\mathbf{X}^{(k)} \cdot \left((1 + \epsilon^{(k)}) \cdot \mathbf{I}_n + \mathbf{A}\right)\right) \tag{4}$$

In which $\mathbf{I}_n$ is the identity matrix of size $n$, $\epsilon^{(k)}$ is a learnable paremeter. $\text{MLP}^{(k)}$ is user-defined multilayer perceptron and is parameterized by some parameters $\theta_k$.

After getting the embedding $\mathbf{Z} = \mathbf{X}^{(K)}$ whose columns represent node embeddings, node-level and edge-level tasks can be performed directly. For graph-level tasks, we need a global pooling operation to summarize those node embeddings and extract the dense graph embedding $\mathbf{x}_{\mathcal{G}}$:

$$\mathbf{x}_{\mathcal{G}} = \text{READOUT}(\mathbf{Z}) \tag{5}$$

Where READOUT is some function that takes the node embeddings and returns a fixed-size representation to express the graph features. For example, we can do a simple sum or average over the embeddings as follows:

$$\mathbf{x}_{\mathcal{G}} = \text{READOUT}(\mathbf{Z}) = \text{READOUT}\left(\{\mathbf{z}_u, u \in \mathcal{V}\}\right) \tag{6}$$

$$= \sum_{u \in \mathcal{V}} \mathbf{z}_u \tag{7}$$

$$\text{or } \mathbf{x}_{\mathcal{G}} = \frac{1}{n} \sum_{u \in \mathcal{V}} \mathbf{z}_u \tag{8}$$

After that, we can use it for the objective task.

## IV. Method

In this work, we first preprocess and transform the images into graphs one by one. The task now becomes graph classification instead of image classification. We then apply a GNN model to classify graphs into ”Normal” and ”Tuberculosis” classes.

### A. Data Preprocessing

X-ray images, or other kinds of medical images, are almost all in the DICOM format. Aside from the pixel information as 12-16 bit pixels, a DICOM image contains a variety of other data such as patient information. As a result, we want to convert and save them as PNG images.

The chest X-ray images are typically large in resolution, which makes our model bulky and takes a long time to train. Therefore, we take a resize step to process them into a resolution of 512 x 512.

We also take a histogram equalization step using CLAHE (Contrast Limited Adaptive Histogram Equalization) technique [15] for a subset of image dataset that have low contrast, blurry images (Fig. 3).

### B. Feature Extraction

After being preprocessed, the images are ready to be converted to graphs. Firstly, we perform an edge detection step

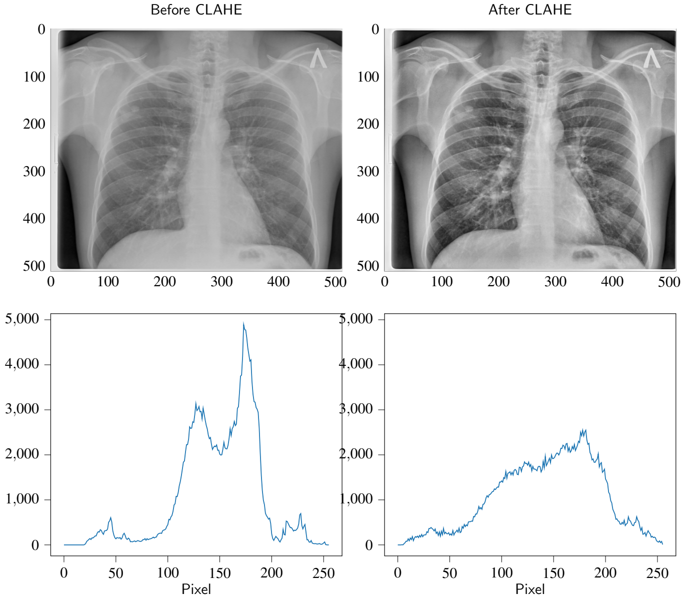


Fig. 3: A visualization of CLAHE along with the histograms.

for finding the boundaries of objects within images using the Prewitt algorithm. The Prewitt algorithm makes use of a pair of 3x3 convolutional kernels:

$$\mathbf{K}_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \mathbf{K}_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{9}$$

We obtained the results $\mathbf{G}_x, \mathbf{G}_y$ after applying these kernels to the original image, which represent the horizontal and vertical gradients. The results are then combined to form the edge image $\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$. Fig. 2a and Fig. 2b show before and after the edge detection process.

The graph will then be constructed with nodes that are pixels within the range [20, 240] (this is obtained from our experiments), and edges connecting adjacent pixels in the image. The Fig. 2c shows the pixels that are selected from Fig. 2b to build graph. In addition, we will also remove a small portion of the isolate nodes in the graph. Using only a fraction of the pixels will remove many features that have little classification power.

We also collect nodes features $\mathbf{X}$. Each of node $u$ are described as $\mathbf{x}_u = [i, j, p, p_x, p_y]^\top$. Where $i, j$ are coordinates of the pixel corresponding to $u$; $p, p_x$ and $p_y$ are the pixel value of the original image, the $\mathbf{G}_x$ and $\mathbf{G}_y$.

### C. Model selection

In this work, we use a GNN model with 3 sequential GIN blocks. For each block, a MLP consisting of 2 layers is applied to the aggregated messages. The outputs of the blocks will then be concatenated in order to get the final graph representation. Then it will be classified using logistic regression. More details are described in Fig. 4 and Algorithm 1.

From the Algorithm 1, the average operations in line 7 and 10 take $\mathcal{O}(n)$ time, the update operations in line 9 will take $\mathcal{O}(m)$ time, where $m = |\mathcal{E}|$. Because if we store the edges as an adjacency list, we can update all of the nodes by traversing them one by one. Resulting into $\mathcal{O}(n+m)$ time complexity overall.

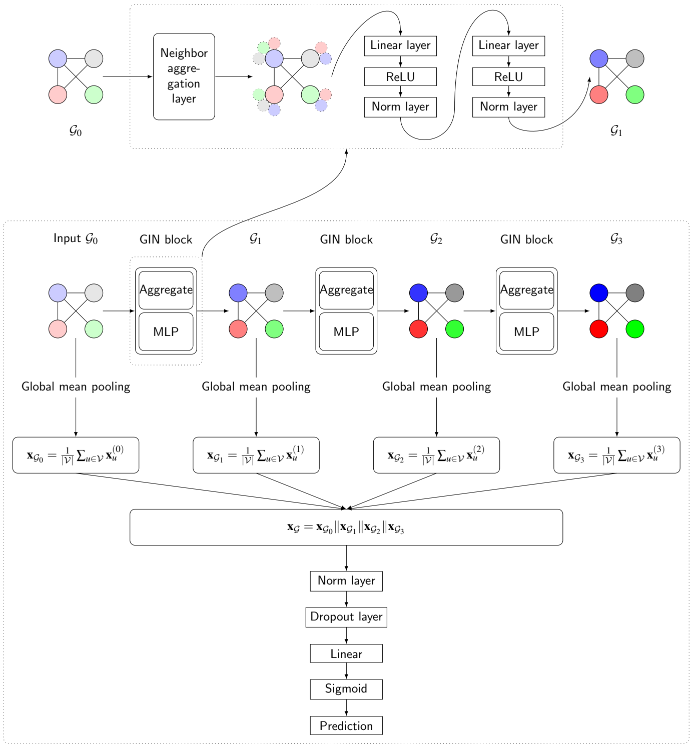

$\mathcal{G}_0$
Neighbor aggre-gation layer
Linear layer
ReLU
Norm layer
Linear layer
ReLU
Norm layer
$\mathcal{G}_1$
Input $\mathcal{G}_0$
GIN block
$\mathcal{G}_1$
GIN block
$\mathcal{G}_2$
GIN block
$\mathcal{G}_3$
Aggregate
MLP
Aggregate
MLP
Aggregate
MLP
Global mean pooling
Global mean pooling
Global mean pooling
Global mean pooling
$\mathbf{x}_{\mathcal{G}_0} = \frac{1}{|\mathcal{V}|}\sum_{u\in\mathcal{V}} \mathbf{x}_u^{(0)}$
$\mathbf{x}_{\mathcal{G}_1} = \frac{1}{|\mathcal{V}|}\sum_{u\in\mathcal{V}} \mathbf{x}_u^{(1)}$
$\mathbf{x}_{\mathcal{G}_2} = \frac{1}{|\mathcal{V}|}\sum_{u\in\mathcal{V}} \mathbf{x}_u^{(2)}$
$\mathbf{x}_{\mathcal{G}_3} = \frac{1}{|\mathcal{V}|}\sum_{u\in\mathcal{V}} \mathbf{x}_u^{(3)}$
$\mathbf{x}_{\mathcal{G}} = \mathbf{x}_{\mathcal{G}_0} \| \mathbf{x}_{\mathcal{G}_1} \| \mathbf{x}_{\mathcal{G}_2} \| \mathbf{x}_{\mathcal{G}_3}$
Norm layer
Dropout layer
Linear
Sigmoid
Prediction

Fig. 4: Proposed model architecture

**Algorithm 1** Feedforward process

```
1:  Input
2:      $\mathcal{G}$    The graph, consist of nodes features matrix $\mathbf{X} \in \mathbb{R}^{5\times n}$ and adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$
3:      $\theta$    Model parameters, consist of $\theta_1$, $\theta_2$, $\theta_3$, $\epsilon^{(1)}$, $\epsilon^{(2)}$, $\epsilon^{(3)}$, $\mathbf{a}, b$
4:  Output
5:      $\hat{y}$    Prediction score in $[0, 1]$

6:  function FEEDFORWARD($\mathbf{X}, \mathbf{A}, \theta$)
7:      $\mathbf{x}_{\mathcal{G}_0} = \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{X}_{:,i}$    ▷ Average over the column vectors
8:      for $k = 1, 2, 3$ do
9:          $\mathbf{X} = \text{MLP}^{(k)}\left(\mathbf{X} \cdot \left((1+\epsilon^{(k)}) \cdot \mathbf{I}_n + \mathbf{A}\right)\right)$
10:         $\mathbf{m}_k = \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{X}_{:,i}$
11:     $\mathbf{x}_{\mathcal{G}} = \mathbf{x}_{\mathcal{G}_0} \| \mathbf{x}_{\mathcal{G}_0} \| \mathbf{x}_{\mathcal{G}_2} \| \mathbf{x}_{\mathcal{G}_3}$    ▷ Concatenation
12:     $\hat{y} = \sigma\left(\mathbf{x}_{\mathcal{G}}^\top \mathbf{a} + b\right)$    ▷ $\mathbf{a}, b \in \theta$
13:     return $\hat{y}$
```

## V. Experiments

### A. Dataset

In our experiments, we use the dataset as same as Rahman et al. [4]. The dataset [18] contains 7000 CXR images, including

TABLE I: The comparative performance of different models for TB classification. Time comparisons can be unfair due to experimental configuration differences.

| Method | Model | Accuracy | Weighted Average | | | | Inference time | Epoch time |
|---|---|---|---|---|---|---|---|---|
| | | | Precision | Sensitivity | Fl-score | Specificity | | |
| **Adapted from Rahman et el. [4], the image data is not lung segmented** | ResNet18 | 93.85 | 94.08 | 93.85 | 93.84 | 93.91 | 0.2 | 39.24 |
| | ResNet50 | 93.11 | 93.4 | 93.11 | 93.09 | 93.16 | 0.35 | 42.7 |
| | ResNet101 | 94.55 | 94.74 | 94.55 | 94.54 | 94.59 | 0.68 | 49.6 |
| | ChexNet | 96.47 | 96.62 | 96.47 | 96.47 | 96.51 | 0.4 | 43.85 |
| | InceptionV3 | 95.72 | 95.92 | 95.73 | 95.73 | 95.92 | 1.02 | 75.59 |
| | Vgg19 | 95.8 | 95.95 | 95.8 | 95.8 | 95.85 | 1.51 | 62.2 |
| | DenseNet201 | 95.07 | 95.27 | 95.07 | 95.07 | 95.12 | 0.58 | 93.6 |
| | SqueezeNet | 94.18 | 94.31 | 94.18 | 94.17 | 94.21 | 0.19 | 38 |
| | MobileNet | 94.33 | 94.65 | 94.33 | 94.32 | 94.39 | 0.13 | 39.42 |
| **Adapted from Rahman et al. [4], lung segmented by U-net [16] or Modified U-net [17]** | ResNet 18 | 96.84 | 97.14 | 96.48 | 96.96 | 97.42 | 0.24 | **32.43** |
| | ResNet50 | 97.07 | 97.34 | 97.07 | 97.14 | 97.36 | 0.4 | 33.45 |
| | ResNet101 | 97.96 | 98 | 97.96 | 97.96 | 97.93 | 0.75 | 43.45 |
| | ChexNet | 98.14 | 98.15 | 98.14 | 98.14 | 98.13 | 0.78 | 38.29 |
| | InceptionV3 | 98.54 | 98.57 | 98.54 | 98.54 | 98.52 | 1.15 | 70.38 |
| | Vgg19 | 97.91 | 97.95 | 97.92 | 97.92 | 97.89 | 1.7 | 59.5 |
| | DenseNet201 | 98.6 | 98.57 | 98.56 | 98.56 | 98.54 | 0.8 | 88.67 |
| | SqueezeNet | 96.58 | 97.08 | 96.75 | 96.66 | 97.32 | 0.22 | 34.2 |
| | MobileNet | 96.9 | 97.29 | 96.68 | 96.86 | 97.62 | 0.15 | 33.23 |
| **Alawi et al. [2] proposed** | CNN | 98.81 | - | 98.86 | - | 98.76 | - | - |
| **Our proposed** | GIN | **99.33** | **99.34** | **99.33** | **99.33** | **99.35** | **0.007** | 34.43 |

TABLE II: Data split details

| Part | Ratio | Amount |
|---|---|---|
| Training set | 0.7 | 4900 |
| Validation set | 0.15 | 1050 |
| Test set | 0.15 | 1050 |
| Total | 1.00 | 7000 |

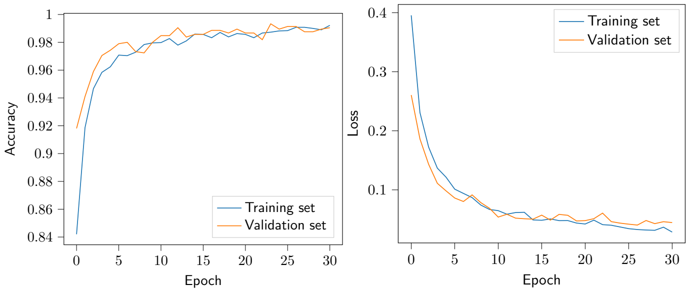


Fig. 5: Learning curves

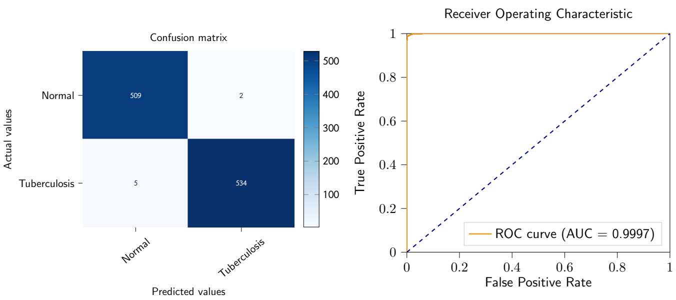


Fig. 6: Results in test set

3500 non-TB cases and 3500 TB cases. It is collected from the data repositories:

- **NLM dataset** [19]: National Library of Medicine (NLM) in the U.S. has made two lung X-ray datasets publicly available: the Montgomery and Shenzhen datasets.
- **Belarus dataset** [20]: Belarus Set was collected for a drug resistance study initiated by the National Institute of Allergy and Infectious Diseases, Ministry of Health, Republic of Belarus.
- **NIAID TB dataset** [21]: NIAID TB portal program dataset, which contains about 3000 TB positive CXR images from about 3087 cases. (Note: This dataset is only available when an agreement is signed)
- **RSNA CXR dataset** [22]: RSNA pneumonia detection challenge dataset, which is comprised of about 30,000 chest X-ray images, where 10,000 images are normal and others are abnormal and lung opacity images. [18]

The dataset contains 3500 non-TBs and 700 TBs can be downloaded at kaggle, the other 2800 TBs can be downloaded from NIAID TB portal by signing an agreement.

### *B. Metrics*

First, to evaluate the effectiveness of the model, we use common classification metrics such as accuracy, precision, sensitivity (recall, F1-score, specificity, confusion matrix, and the ROC-AUC chart.

Second, we will measure the average time to perform training in a single epoch (epoch time); along with the inference and prediction time for an input unit. All that in seconds.

### *C. Experimental environment and hyperparameters*

Here is our computer configuration to run experiments on datasets and evaluate the effectiveness of the proposed method:

1) CPU: Intel(R) Xeon(R) Gold 5220R CPU @ 2.20GHz

2) GPU: NVIDIA A100-PCIE-40GB

The data set was randomly divided into three parts, as shown in Tab. II. Each part was divided into mini-batches with a size of 32. The learning algorithm is Adam, with a learning rate of 2e-4. During training, the learning rate is cut every 5 epochs by 20%. The training will stop when the maximum number of epochs is reached or the model does not improve after 3 consecutive epochs on the validation set.

### *D. Result*

We only apply the preprocessing step for the 2800 images from the NIAID TB portal since the others are pretty nicely preprocessed. After feature extracting, the training process on graphs takes 31 epochs. The outputs at each epoch are analyzed as in Fig. 5. From the results as described in Fig. 6, it can be observed that the model classified the data in the test set very effectively with an accuracy of 99.33%, AUC of 99.97%. We also compare our results with the previous works by Rahman et al. [4] and Alawi et al. [2] in Tab. I. And we can conclude that our results outperformed theirs in terms of classification performance.

## VI. Conclusion

In this paper, we proposed a new CAD method based on a graph neural network from the successes achieved by it in many fields and the desire to join hands to fight tuberculosis in the community. The proposed method achieves high efficiency on the experimental data set in terms of training time and the accuracy of the approach. The CLAHE image preprocessing technique was used in the image preprocessing stage, which improved the input quality of the model. Next, extracting important features through edge detection can reduce the graph size and less critical information, contributing to the optimization of model parameters and training time. The classification process is carried out through the GIN network after transforming the image data set into a graph dataset based on the isomorphic properties of graphs to classify them. The model achieves impressive results on measures such as accuracy 99.33%, recall 99.07%, precision 99.63%, f1-score 99.35%, AUC 99.97% - better than existing methods on the same data set.

In the future, combining cases to perform diagnosis on real data sets is an excellent opportunity to demonstrate the model's goodness in diagnosing tuberculosis.

## Acknowledgment

This work is supported by the TB Portals Consortium and the TB Portals Program of the National Institutes of Health (NIH).

## References

[1] "Global tuberculosis report 2021. geneva: World health organization; 2021. licence: CC BY-NC-SA 3.0 IGO.,"

[2] A. E. B. Alawi, A. Al-basser, A. Sallam, A. Al-sabaeei, and H. Alkhateeb, "Convolutional neural networks model for screening tuberculosis disease," in *2021 International Conference of Technology, Science and Administration (ICTSA)*, pp. 1–5, IEEE, 2021.

[3] S. I. Nafisah and G. Muhammad, "Tuberculosis detection in chest radiograph using convolutional neural network architecture and explainable artificial intelligence," *Neural Computing and Applications*, pp. 1–21, 2022.

[4] T. Rahman, A. Khandakar, M. A. Kadir, K. R. Islam, K. F. Islam, R. Mazhar, T. Hamid, M. T. Islam, S. Kashem, Z. B. Mahbub, *et al.*, "Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization," *IEEE Access*, vol. 8, pp. 191586–191601, 2020.

[5] L. T. Duong, N. H. Le, T. B. Tran, V. M. Ngo, and P. T. Nguyen, "Detection of tuberculosis from chest x-ray images: boosting the performance with vision transformer and transfer learning," *Expert Systems with Applications*, vol. 184, p. 115519, 2021.

[6] F. Pasa, V. Golkov, F. Pfeiffer, D. Cremers, and D. Pfeiffer, "Efficient deep network architectures for fast chest x-ray tuberculosis screening and visualization," *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019.

[7] S. Rajaraman, G. Zamzmi, L. R. Folio, and S. Antani, "Detecting tuberculosis-consistent findings in lateral chest x-rays using an ensemble of cnns and vision transformers," *Frontiers in Genetics*, p. 390, 2022.

[8] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, *et al.*, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 590–597, 2019.

[9] A. Bustos, A. Pertusa, J.-M. Salinas, and M. de la Iglesia-Vayá, "Padchest: A large chest x-ray image dataset with multi-label annotated reports," *Medical image analysis*, vol. 66, p. 101797, 2020.

[10] S.-Y. Lu, S.-H. Wang, X. Zhang, and Y.-D. Zhang, "Tbnet: a context-aware graph network for tuberculosis diagnosis," *Computer Methods and Programs in Biomedicine*, vol. 214, p. 106587, 2022.

[11] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*, pp. 1263–1272, PMLR, 2017.

[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017.

[14] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," *arXiv preprint arXiv:1810.00826*, 2018.

[15] K. Zuiderveld, *Contrast Limited Adaptive Histogram Equalization*, p. 474–485. USA: Academic Press Professional, Inc., 1994.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[17] R. Azad, M. Asadi-Aghbolaghi, M. Fathy, and S. Escalera, "Bi-directional convlstm u-net with densley connected convolutions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

[18] T. Rahman, A. Khandakar, and M. E. H. Chowdhury, "Tuberculosis (tb) chest x-ray database," 2020.

[19] S. Jaeger, S. Candemir, S. Antani, Y.-X. J. Wáng, P.-X. Lu, and G. Thoma, "Two public chest x-ray datasets for computer-aided screening of pulmonary diseases," *Quantitative imaging in medicine and surgery*, vol. 4, no. 6, p. 475, 2014.

[20] "Belarus tb database and tb portal." https://grantome.com/grant/NIH/AAI12021001-1-0-5. Accessed: 2022-06-14.

[21] "Niaid tb portal program." https://tbportals.niaid.nih.gov/. Accessed: 2022-06-14.

[22] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.