Evaluating Vision GNN on Pediatric Chest X-Ray Images

Dante Miller
Rice University
Houston, TX, 77065
dm85@rice.edu

Abstract

State-of-the-art computer vision models employ uniquely different approaches to image processing. They typically combine a feature extractor with a linear classifier to learn the correct (image, label) relationships. However, these models often utilize a grid-like sequence structure, which lacks the flexibility required to effectively capture irregular and complex objects. The Vision GNN was developed specifically to address this issue. Instead of a grid-like sequence structure, it uses a graph representation of the image as input and learns the correct (image, label) relationships. In our project, we explore the use of training and transfer learning with Vision GNN alongside state-of-the-art CNN, MLP, and transformer models on pediatric radiographs, comparing their performance using standard metrics. We demonstrate that Vision GNN outperforms various model architectures in terms of training and transfer learning.

1 Introduction

Chest radiography stands as a cornerstone in medical imaging, playing a crucial role in the screening and diagnosis of various disorders that affect the lung parenchyma, airways, heart, and mediastinum. It proves invaluable not only in assessing support devices but also in diagnosing critical conditions such as pneumonia (8). Importantly, the interpretation of chest radiographs varies significantly among different age groups due to the changing radiographic appearances with age. Viral respiratory infections, especially COVID-19, have placed a considerable strain on healthcare systems, manifesting in increased clinic visits, hospital admissions, and, regrettably, elevated mortality rates (7). These challenges highlight the urgent need for the development of data-driven methodologies and computational tools that support physicians in diagnosing viral respiratory infections (9).

Employing computer vision models in medical imaging represents a formidable challenge that numerous researchers aim to overcome. The main difficulty involves precisely identifying and correlating specific regions within images to pertinent medical diagnoses, such as types of diseases. Images, although represented in a Euclidean space (2D grid of pixels), often encompass content that depicts non-Euclidean relationships (5). This recognition has sparked a growing interest in processing images as graphs and learning the relationships between graphs and labels, particularly through innovations like Vision Graph Neural Networks (Vision GNNs) (3). Our research is dedicated to analyzing pediatric lung chest X-ray images sourced from both a publicly accessible dataset (4) and a proprietary dataset (10), showcasing examples of paired (medical image, label) samples in Figure 1.

2 Related Work

Early research on applying machine learning to medical diagnosis concentrated on convolution-based models for lung chest X-ray datasets (1)(4)(11). TorchXRayVision (2), recognized as one of the

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

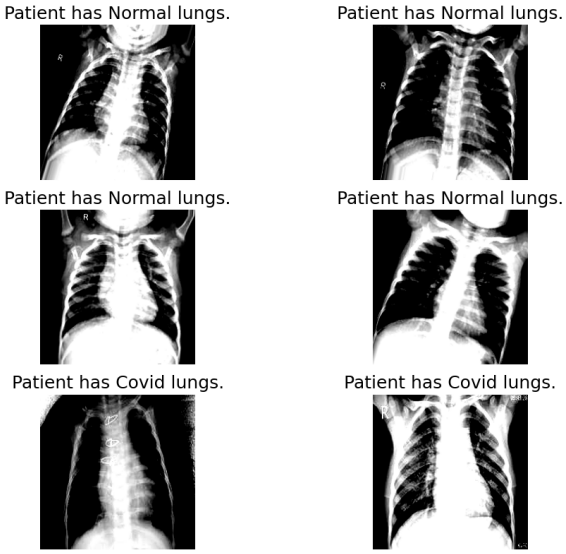

Figure 1: Samples of pediatric lung chest X-ray images and their respective labels are displayed.

state-of-the-art methods in medical diagnosis, trained various convolution-based models on diverse chest X-ray image datasets. Similar efforts have focused on convolution-based models tailored to pediatric chest X-ray image datasets (1)(4)(11). However, there appears to be a lack of similar work applying Vision GNN (3) to medical image datasets. Our work is focused on evaluating Vision GNN alongside ResNet50, CycleMLP-B2, and PVTv2-B2 to determine whether a graph-based model outperforms CNN, MLP, and transformer-based models in terms of training and transfer learning on medical images. It should be noted that these models were also mentioned in (3), so they were chosen to ensure an equal setting, alongside the small Vision GNN and the small Pyramid Vision GNN—a variation of the isotropic version that adopts a pyramid structure—models.

3 Methods

3.1 Graph Representation

In this subsection, we discuss concepts in relation to graph representation and graph processing highlighted in (3).

A graph G is typically defined as a tuple $G = (V, E)$, where V represents the set of vertices, and E represents the edges that establish connectivity between these vertices. In the context of imaging, consider an image with dimensions $H \times W \times 3$; this image is divided into N patches. Each patch is converted into a feature vector x_i in $\mathbb{R}^D$, resulting in a set of feature vectors $X = [x_1, x_2, \ldots, x_N]$ where D represents the dimensionality of the features. These vectors act as nodes $V = \{v_1, v_2, \ldots, v_N\}$, and connections or edges $e_{i,j}$ are established based on the proximity or similarity between the nodes, typically connecting each node v_i to its K nearest neighbors to form the graph $G = (V, E)$.

Upon structuring the initial feature set X into a graph $G = G(X)$, the system employs graph convolutional layers to facilitate robust information exchange among the nodes. This exchange is governed by the operation:

$$G' = F(G, W) = Update(Aggregate(G, W_{agg}), W_{update}),$$

where W_{agg} and W_{update} are the adjustable weights for the aggregation and update processes, respectively. During aggregation, the features of neighboring nodes are summed to compute a node's enhanced representation, which is further refined by the update process using W_{update}. This method of using graph convolutions, particularly the max-relative type, ensures efficient and effective processing, succinctly captured by $X' = GraphConv(X)$.

3.2 Vision GNN

In this subsection, we discuss concepts in relation to Vision GNN as outlined in (3). Building off the graph representation subsection, Vision GNN introduces a multi-head update operation. The aggregated feature x_i' is divided into h heads, which are then updated with distinct weights and concatenated:

$$x_i' = [head_1 W^1_{update}, head_2 W^2_{update}, \ldots, head_h W^h_{update}].$$

The main contribution of Vision GNN is the ViG block, as illustrated in Figure 2. The ViG block introduces the concept of a Grapher module, which is expressed as $Y = \sigma(GraphConv(XW_{in}))W_{out} + X$, where $Y \in \mathbb{R}^{N \times D}$, W_{in} and W_{out} are the weights of the fully-connected layers, σ represents the activation function (e.g., ReLU and GeLU), and the bias term is omitted, given an input feature $X \in \mathbb{R}^{N \times D}$.

Additionally, the ViG block incorporates a Feed-Forward Network (FFN) module, a simple multi-layer perceptron with two fully-connected layers: $Z = \sigma(YW_1)W_2 + Y$, where $Z \in \mathbb{R}^{N \times D}$, W_1 and W_2 are the weights of the fully-connected layers, and the bias term is omitted. Batch normalization is applied after every fully-connected or graph convolution layer within both the Grapher and FFN modules. In the output layer, they employ pooling and an MLP model.

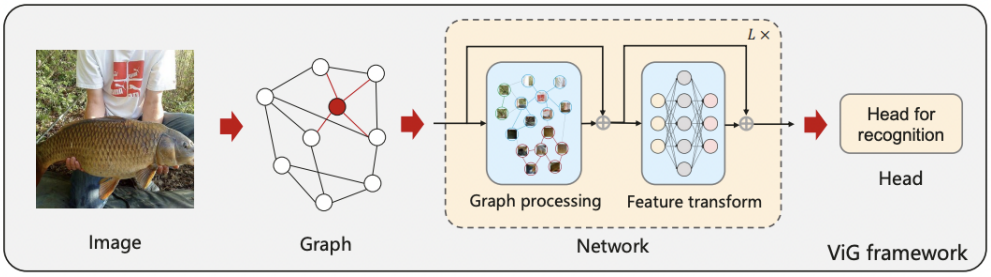

Figure 2: Visualizing the Vision Transformer (ViT) Architecture. Image sourced from (3).

4 Code Availability

Detailed information about Vision GNN, ResNet-50, CycleMLP-B2, and PVTv2-B2 can be found in their respective repositories: Vision GNN, ResNet-50, CycleMLP-B2, PVTv2-B2. My code and experimentation details are available at ViG-Pediatric.

5 Experiments and Results

5.1 Data

For these experiments we considered two types of pediatric lung chest X-ray image datasets: P1 (ORI) (10) and P2 (UCSD)(4). Further details about these datasets can be found in Figure 1.

5.2 Experiment Types

5.2.1 Training

The first experiment we consider involves training the models on each dataset to determine which model exhibits the best performance.

5.2.2 Transfer Learning

After training the models on a specific dataset, we proceed to evaluate their performance on the opposing dataset.

Dataset	Pneumonia	Normal	Train	Validation	Test
P1 (UCSD)	1493	1583	1967	493	616
P2 (ORI)	404	419	526	132	165

Table 1: Distribution of pneumonia and normal cases across different datasets with train, validation, and test splits.

5.3 Implementation Details

The majority of the code was implemented using PyTorch. The model components for Vision GNN, CycleMLP-B2, ResNet-50, and PVTv2-B2 were referenced from their respective repositories. Further details can be found in the code availability section above. Initially, the code retrieves data from my Google Drive, followed by a preprocessing stage that normalizes the data. Subsequently, we visualize the images, with examples available in the results folder. We then conduct the training experiment, and these results are saved. Afterward, the transfer learning experiment is performed, and these results are also saved. The code for the experiments was written by me, referencing the model repositories for the model to train.

Regarding data characteristics, I used batch sizes of 64, and the images were resized to 224×224 pixels. For training, the AdamW optimizer was employed; the models use a learning rate of $lr = 1 \times 10^{-4}$, with weight decay set at 1×10^{-2}. All models are trained for 50 epochs, incorporating early stopping with a patience of 30 epochs. For future work, I aim to implement parameter tuning. However, due to time constraints in this project, I determined the parameters through experimentation. I decided set the parameters for all models to a default setting, as the results were generally consistent across different parameter settings, allowing for an equal comparison by holding the factors related to parameters constant.

5.4 Results

The experimental results presented in the tables demonstrate the performance of the models across pediatric datasets ORI and UCSD, with a focus on training and transfer learning. When examining 2, it is noticeable that the models train well across the board for UCSD, which is an easier dataset to train on. This raises the question of whether they are learning relevant information that generalizes well across different pediatric radiograph datasets. In the case of ORI, we observe better results in terms of recall for ViG-S compared to other models, with all metrics above 70 percent. ResNet-50 shows almost similar results but has worse recall. Interestingly, Pyramid ViG-S appears to have lower recall, but its other metrics are high, prompting speculation about whether this model performs at the same level as ViG-S. Overall, it is evident that the Vision GNN models outperform Cycle MLP-B2, ResNet-50, and PVTv2-B2 in terms of overall performance.

When examining 3, we see poor recall across the board when considering the Cycle MLP-B2, ResNet-50, and PVTv2-B2 trained on UCSD and then evaluated on ORI. An intriguing observation from ORI is the performance of Pyramid ViG-S, which is on par with the best model trained on ORI. Although the results for ViG-S are poorer than those for Pyramid ViG-S, they are still better than those for Cycle MLP-B2, ResNet-50, and PVTv2-B2. When assessing the generalization ability to UCSD, ViG-S performs poorly compared to Cycle MLP-B2, ResNet-50, and PVTv2-B2. In this section, the best model is most likely Pyramid ViG-S, but PVTv2-B2 follows very closely in terms of performance. From these results, it is clear that the generalization ability of Pyramid ViG-S outperforms the other models, but ViG-S is also easier to train. The performance difference between Pyramid ViG-S and ViG-S is attributed to the pyramid structure adopted by the former, which allows it to learn more complex information as it involves downscaling then upscaling, explaining why it might be more difficult to train compared to ViG-S, yet it shows improvements in overall generalization ability.

In Figures 3 and 4, the Grad-CAM results are noticeable. The first observation is that Cycle MLP-B2 tends to highlight the peripheries, whereas ResNet-50 highlights a larger central area. When examining Vision GNN models and PVTv2-B2, there is a noticeable focus on the right lungs. This may suggest that the features learned from the UCSD dataset are broader, as evidenced by the increasing specificity of the features when observed in the UCSD context. Cycle MLP-B2 ignites or ignores a lot in specific regions. ResNet-50, on the other hand, highlights a more specific area when it

Dataset	Method	Accuracy	Precision	Recall	AUC
ORI	CycleMLP-B2	0.7515	0.7941	0.6667	0.8110
ORI	Pyramid ViG-S	0.7697	0.8413	0.6543	0.8276
ORI	PVTv2-B2	0.7152	0.8542	0.5062	0.8132
ORI	ResNet-50	0.7394	0.7639	0.6790	0.8144
ORI	ViG-S	0.7576	0.7733	0.7160	0.8042
UCSD	CycleMLP-B2	0.9269	0.9601	0.8863	0.9835
UCSD	Pyramid ViG-S	0.9448	0.9461	0.9398	0.9827
UCSD	PVTv2-B2	0.9562	0.9755	0.9331	0.9885
UCSD	ResNet-50	0.9513	0.9590	0.9398	0.9861
UCSD	ViG-S	0.9529	0.9592	0.9431	0.9911

Table 2: Training experimental results for various models on ORI and UCSD datasets.

Dataset	Method	Accuracy	Precision	Recall	AUC
ORI	Cycle MLP-B2	0.6355	0.8611	0.3069	0.7224
ORI	Pyramid ViG-S	0.7473	0.7722	0.6881	0.8110
ORI	PVTv2-B2	0.6671	0.7955	0.4332	0.7047
ORI	ResNet-50	0.6950	0.8558	0.4554	0.7419
ORI	ViG-S	0.6889	0.7701	0.5223	0.7603
UCSD	Cycle MLP-B2	0.6447	0.6894	0.4876	0.6957
UCSD	Pyramid ViG-S	0.7520	0.7542	0.7254	0.8336
UCSD	PVTv2-B2	0.7331	0.7228	0.7301	0.8075
UCSD	ResNet-50	0.6814	0.6277	0.8446	0.7823
UCSD	ViG-S	0.3768	0.4040	0.5975	0.3490

Table 3: Transfer-learning experimental results for various models on ORI and UCSD datasets. The models used were previously trained on the counterpart dataset.

is correct. ViG-S, one of the weaker performers in terms of this aspect, shows performance scattered everywhere. Surprisingly, both the transformer-based Pyramid ViG-S model and other related models highlight the same area of the right lungs but still slightly in other areas that might not be as relevant.

5.5 Discussion

From the results section, we observed that Vision GNN models are easier to train compared to Cycle MLP-B2, ResNet-50, and PVTv2-B2. It should be noted that we are considering smaller variations of these models, which show a substantial 2 percent performance increase compared to their smaller counterparts when trained on ImageNet (3). This difference is significant in terms of fine-tuning, as it

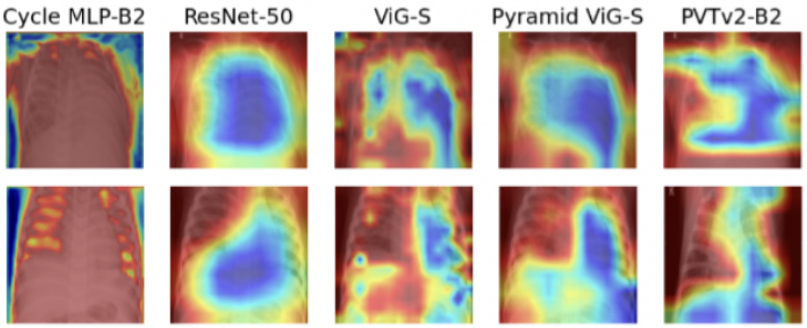

Figure 3: Showing Grad-CAM examples of the transfer learning experimental results for various models on ORI. The models used were previously trained on the counterpart dataset.

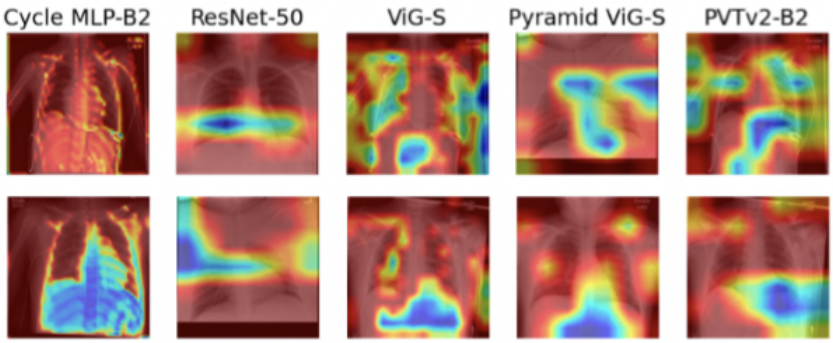

Figure 4: Showing Grad-CAM examples of the transfer learning experimental results for various models on UCSD. The models used were previously trained on the counterpart dataset.

can lead to vast improvements in overall transfer learning capabilities. However, the effectiveness depends on factors such as whether we are learning what is intended, the number of parameters, and the size of the data. Interestingly, the generalization ability of pyramid models is generally good compared to other models. I suspect this might be due to the parameters behaving differently. One issue with models like convolutions, transformers, and MLPs is that they learn both relevant and irrelevant connections. For instance, while convolutions use a sliding window, not all patches or connections might be relevant. In contrast, graph-based models, which focus more on nodes, imply that all connections are slightly more pertinent. Here, each vertex's new associated value is determined more by its connections with other vertices.

There are many potential avenues for exploration. In my survey (6), I examined graph models for images, such as DeepGCN, Vision GNN and Vision HGNN, to understand more about these issues as they relate to my current project. We learned that modifying fundamental aspects of graph theory—such as the types of graphs used and the assumptions about their structure—can profoundly influence the learning process. This includes understanding the (image, label) relationships and integrating key concepts from previous models like dense connections and residual connections. Further exploration of computer vision-based models or incorporating these concepts into graph models like Vision GNN could potentially lead to improvements in efficiency and accuracy. The inclusion of experts in analyzing the results is also necessary. I cannot determine whether the model's focal point is accurate, but actual doctors can help us understand where the model should be focusing.

In terms of predicting pneumonia in pediatric patients, I am curious about training Vision GNN on adult radiographs. The main consideration arises from comparing the availability of pediatric radiograph datasets for pneumonia prediction with the availability of adult radiographs. Notably, there are significantly more adult radiograph datasets. This scarcity makes it challenging to train models on pediatric datasets with the hope that they will generalize well to new pediatric datasets. We cannot assume generalization will be effective, as we likely have not observed the complete distribution due to data limitations. Using adult radiographs addresses this issue, but a problem arises: the features learned may not be relevant, as the diagnostic features in radiographs, such as those for diagnosing COVID, differ with age. Therefore, we cannot expect these models to generalize as well.

From this experiment, we observed that Vision GNN performs better in terms of overall generalization ability. I wonder if this performance is due to it learning fewer irrelevant features compared to other models in this study. When those models are applied to the task mentioned above, they suffer because they learn both irrelevant and relevant features, making it difficult to distinguish the correct class. I wonder if Vision GNN could counter this issue because it focuses not necessarily on relevant features but on features that are more heavily weighted in terms of model connections.

5.6 Main Challenges

I have not encountered any major challenges in this project; rather, I found it very interesting to learn how graph theory can be incorporated into computer vision and the vast potential therein. If there was a primary challenge, it was merely the computing resources. With more computing resources, I could have conducted a comprehensive comparison with most of the models mentioned in (3).

References

[1] M. E. Chowdhury et al. Can ai help in screening viral and COVID-19 pneumonia? *IEEE Access*, 8:132665–132676, 2020.

[2] J. P. Cohen et al. Torchxrayvision: A library of chest X-ray datasets and models. In *International Conference on Medical Imaging with Deep Learning*. PMLR, 2022.

[3] K. Han et al. Vision gnn: An image is worth graph of nodes. In *Advances in Neural Information Processing Systems*, volume 35, pages 8291–8303, 2022.

[4] D. S. Kermany et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131, 2018.

[5] G. Li et al. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[6] D. Miller. Graph models for images: A survey on deepgcn and vision gnn. Technical report, Rice University, Houston, TX, 2024. Available at Rice University (dm85@rice.edu).

[7] G. Nino et al. Pediatric lung imaging features of COVID-19: A systematic review and meta-analysis. *Pediatric Pulmonology*, 56(1):252–263, 2021.

[8] S. Padash et al. Pediatric chest radiograph interpretation: how far has artificial intelligence come? a systematic literature review. *Pediatric Radiology*, 52(8):1568–1580, 2022.

[9] H. H. Pham et al. Pedicxr: An open, large-scale chest radiograph dataset for interpretation of common thoracic diseases in children. *Scientific Data*, 10(1):240, 2023.

[10] Texas Children's Hospital. Data provided by texas children's hospital, 2023. Private communication.

[11] R. M. Wehbe et al. Deepcovid-xr: An artificial intelligence algorithm to detect COVID-19 on chest radiographs trained and tested on a large U.S. clinical data set. *Radiology*, 299(1):E167–E176, 2021.