

Sử dụng Graph Convolutional Networks trong tác vụ phân loại ảnh X-quang phổi

Phạm Ngọc Tân¹ Võ Khánh An¹ Bùi Trí Dũng¹

Abstract

Viêm phổi là một tình trạng bệnh lý rất hay gặp và được phân thành nhiều loại với các nguyên nhân gây bệnh khác nhau. Tuy nhiên để việc thăm khám và điều trị đúng cách thì các bác sĩ phải phân loại được các loại viêm phổi. Đây là một công việc đòi hỏi rất nhiều kiến thức chuyên gia. Sự thiếu hụt nguồn nhân lực trong lĩnh vực y tế đang khiến cho vấn đề ngày càng trầm trọng hơn. Nhằm kịp thời phát hiện, ngăn chặn bệnh viêm phổi và tiết kiệm nguồn nhân lực, nhiều nghiên cứu đã sử dụng cái kỹ thuật máy học nhằm tự động hoá quá trình phân loại ảnh X-quang xem bệnh nhân đó có bị viêm phổi hay không. Trong bài báo cáo này, chúng tôi sẽ sử dụng mô hình Graph Convolutional Neural Networks (GCNs) trong việc phân loại ảnh X-quang và đạt được độ chính xác 96.75% trên tập dữ liệu Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification.

Từ khóa – Graph Convolutional Neural Networks (GCNs), Chest X-ray (CXR), Pneumonia, Classification.

1. Giới thiệu

Viêm phổi là hiện tượng viêm nhiễm của nhu mô phổi bao gồm viêm phế nang, túi phế nang, ống phế nang, tổ chức liên kết khe kẽ và viêm tiểu phế quản tận cùng. Các phế nang, đường dẫn khí trong phổi có thể chứa đầy chất lỏng hoặc mủ, gây ho có đờm hoặc mủ, sốt, ớn lạnh và khó thở. Thông thường, có nhiều tác nhân gây ra tình trạng viêm phổi, nhưng thường do vi khuẩn, virus và nấm. Bệnh viêm phổi có nhiều mức độ khác nhau từ viêm nhẹ, đến viêm phổi nặng. Bệnh đặc biệt nguy hiểm đối với trẻ sơ sinh, trẻ nhỏ và người trên 65 tuổi có nhiều bệnh nền, hoặc hệ miễn dịch suy

yếu. Viêm phổi có thể được phân loại theo nhiều cách khác nhau tùy theo nguyên nhân và mục đích phân loại, nhưng nhìn chung hiện nay viêm phổi chủ yếu được phân loại dựa trên nguyên nhân gây viêm phổi và nguồn lây nhiễm bệnh. Trong đó nguyên nhân gây bệnh phổ biến nhất là do virus và vi khuẩn. Những tổn thương trong phổi như bệnh viêm phổi và các bệnh lý khác có thể xác định bằng cách chụp X-quang phổi. Hơn thế nữa, X-quang phổi còn cung cấp những hình ảnh của tim, mạch máu, đường thở và các xương, cột sống. Đây là một xét nghiệm thường quy và rất phổ biến trong chẩn đoán và điều trị bệnh. Ảnh X-quang phổi có thể giúp bác sĩ chẩn đoán được tình trạng phổi của bệnh nhân. Ví dụ một ảnh X-quang viêm phổi cho thấy những đám mờ rải rác hay khu trú như nhu mô phổi.

Tuy nhiên, chẩn đoán ảnh y khoa là một chuyên ngành phức tạp, đòi hỏi nhiều năm kinh nghiệm lâm sàng để có thể đưa ra được những chẩn đoán chính xác nhất. Từ đó đưa ra các phác đồ phù hợp cho bệnh nhân. Cùng với sự gia tăng không ngừng của các bệnh về đường hô hấp như COVID-19 hay viêm phổi đã làm gia tăng nhu cầu về chẩn đoán và điều trị bệnh trong các bệnh viện, nhất là nhu cầu về chẩn đoán bệnh với ảnh X-quang phổi. Một ưu điểm nổi bật của sử dụng ảnh X-quang phổi trong chẩn đoán bệnh về đường hô hấp đó chính là giá thành rẻ và tiết kiệm thời gian. Tuy nhiên, mặt hạn chế của việc chẩn đoán này chính là bệnh nhân cần phải đi đến những trung tâm y tế lớn để có được đội ngũ y bác sĩ đủ khả năng chuyên môn trong chẩn đoán ảnh y khoa. Tuy nhiên, các y bác sĩ có chuyên môn cao đôi lúc cũng gặp nhiều sai sót trong chẩn đoán bệnh thông qua ảnh X-quang phổi vì nhiều nguyên nhân khác nhau. Ví dụ như tâm trạng, khối lượng công việc hay những đặc điểm quá nhỏ và quá mờ để phát hiện rõ.

Để giải quyết tình trạng trên, nhiều nghiên cứu đã được đề xuất để giúp các y bác sĩ trong việc chẩn đoán ảnh y khoa thông qua việc xây dựng các hệ thống hỗ trợ bác sĩ trong việc chẩn đoán hình ảnh (Computer-Aided Design hay CAD). Những hệ thống này sử dụng các mạng học sâu như ResNet50, VGG19 [12] hoặc các kỹ thuật trích xuất đặc trưng bằng tay. Sau đó, các bác sĩ sẽ dựa vào kết quả của mô hình như là một tài liệu tham khảo để đưa ra kết luận cuối cùng về tình trạng của bệnh nhân. Tuy vậy, những phương pháp máy học hiện nay đang vấp phải nhiều hạn chế

¹Trường Đại học Công nghệ Thông tin, ĐHQG HCM. Phạm Ngọc Tân <19520925@gm.uit.edu.vn>, Võ Khánh An <19520007@gm.uit.edu.vn>, Bùi Trí Dũng <19521386@gm.uit.edu.vn>.

để có thể triển khai thực tiễn bên trong các hệ thống CAD như việc thiếu đi khả năng giải thích các kết quả đầu ra của chương trình, khả năng phân loại chuẩn xác cho những dạng bệnh có tác nhân gần giống nhau hay những bệnh có ít biểu hiện trên ảnh.

Trong bài báo cáo này, chúng tôi sẽ sử dụng mô hình mạng Neural Đồ thị (Graph Neural Networks hay GNN) trong việc phân loại hai tác nhân khác nhau gây nên bệnh viêm phổi, đó là viêm phổi do vi khuẩn và viêm phổi do virus vì ưu điểm trong việc giải quyết tốt các bài toán thiếu dữ liệu và mất cân bằng dữ liệu tốt của lĩnh vực thị giác máy tính của GNN. Ngoài ra, chúng tôi cũng sẽ trực quan hóa kết quả đầu ra để phân tích những thành phần bị tổn thương bên trong phổi góp phần gây nên bệnh. Bài báo cáo của chúng tôi được chia ra làm .. phần: đầu tiên chúng tôi sẽ giới thiệu sơ qua về các công trình liên quan và cơ sở phương pháp luận trong bài báo cáo này ở phần 2 và 3. Sau đó, chúng tôi sẽ bàn luận sâu hơn về phương pháp mà chúng tôi ứng dụng cho bài toán này ở phần 4. Chúng tôi cũng sẽ đánh giá, so sánh kết quả thực nghiệm và đưa ra kết luận cuối cùng ở phần 5 và 6.

2. Nền tảng

2.1. Graph Neural Networks

Mạng Neural Đồ thị, hay Graph Neural Networks (GNNs), ban đầu được đề xuất và sử dụng để giải quyết các bài toán liên quan đến các đối tượng, thực thể có trong bài toán và tính liên kết giữa các thực thể, đối tượng bên trong bài toán. ví dụ như các đối tượng là phân tử bên trong các bài toán liên quan về nghiên cứu thuốc đặc trị hay các đối tượng là người dùng bên trong các bài toán về hệ thống đề xuất (recommendation system).

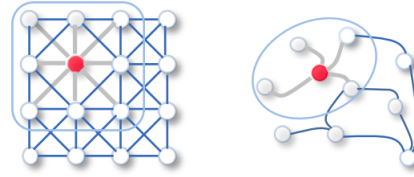
Trong thời gian gần đây, những nhà nghiên cứu về thị giác máy tính đã đặc biệt chú ý đến tính ưu việt của GNNs trong việc xử lý các bài toán về thị giác đòi hỏi phải quan sát và phân tích những thành phần trong ảnh cùng góp phần tạo nên giá trị nhân. Do đó, mạng Neural Đồ thị Tích chập, hay Graph Convolutional Neural Network (GCNs), đã được ra đời với các chỉnh sửa và nâng cấp phù hợp cho các bài toán liên quan đến thị giác máy tính.

2.2. Graph Convolutional Networks

Trong những mô hình kiến trúc Neural Tích chập, hay Convolutional Neural Network (CNNs), các lớp kernels (hay filters) được sử dụng để tổng hợp thông tin từ các pixel lân cận trong ảnh ban đầu. Từ đó trích xuất được các đặc trưng của ảnh nhằm đưa ra các dự đoán phù hợp.

GCNs đã mượn ý tưởng đó từ các mô hình CNNs để sử dụng cho quá trình tổng hợp thông tin ảnh thành các đặc trưng. Thay vì sử dụng các lớp kernels trượt qua từng vùng ảnh,

GCNs sẽ sử dụng đỉnh và cạnh để tổng hợp thông tin từ các đỉnh (pixel) liên kề như trong hình 1 với mặc định rằng các đỉnh càng lân cận thì càng đặc trưng sẽ càng giống nhau.



Hình 1. Minh họa sự khác nhau giữa mô hình CNNs (trái) và GCNs (phải) trong quá trình tổng hợp thông tin.

Các ưu điểm của việc sử dụng GCNs cho ảnh có thể được kể đến như tránh được tình trạng overfit với tập dữ liệu huấn luyện hay mất cân bằng dữ liệu (class imbalance). Ngoài ra, các công trình [13, 9] cho thấy rằng các mô hình GNNs nói chung không cần sử dụng đến GPUs nhưng vẫn đạt được tốc độ huấn luyện nhanh và cho ra kết quả tốt hơn so với các mô hình CNNs. Tuy nhiên, những ưu điểm vượt trội của GNN có thể lý giải bằng việc thiết kế các mô hình đồ thị riêng biệt cho từng bài toán là một công việc đòi hỏi rất nhiều kiến thức chuyên môn mà chỉ có thể được thiết kế bởi các chuyên gia trong cả lĩnh vực thị giác máy tính và chuyên ngành của bài toán được đưa ra để có thể thu được kết quả tối ưu như đã được giới thiệu ở trước, và đặc biệt là không có một mẫu thức chung nào để thiết kế đồ thị phù hợp cho mọi bài toán.

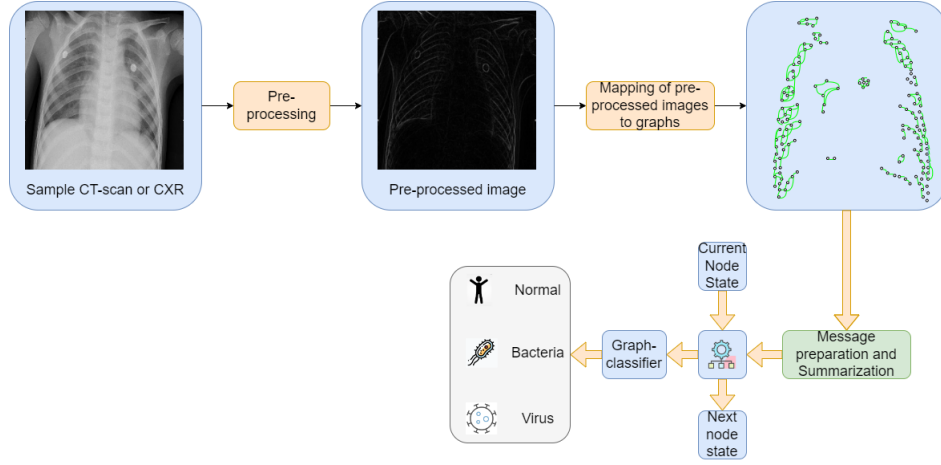
3. Phương pháp

Việc phân loại ảnh X-quang phổi dựa trên những đặc điểm khó nhận ra bằng mắt thường thực sự là thách thức lớn đối với máy tính. Trong bài báo cáo này, chúng tôi sẽ đi giải quyết thách thức này bằng cách sử dụng mô hình GCNs như là một bộ phân lớp nhằm tận dụng triệt để các thông tin thành phần cấu tạo của phổi dù là nhỏ nhất và đánh giá tính liên kết giữa các thành phần đó.

Quy trình tổng quát cho bài báo cáo của chúng tôi gồm có 3 bước chính và được minh họa như ở hình 2: tiền xử lý, cấu trúc đồ thị cho dữ liệu, xây dựng mô hình GCNs và huấn luyện cho đồ thị đã được tạo ra.

3.1. Tiền xử lý

Đầu tiên, ảnh gốc sẽ được điều chỉnh kích cỡ (resize) về 224x224. Sau đó, chúng tôi áp dụng các kỹ thuật xử lý ảnh để thu được những đặc trưng về cạnh phù hợp cho bài toán đã cho. Cụ thể, chúng tôi tiến hành sử dụng 3 phương pháp trích xuất đặc trưng cạnh truyền thống: Sobel [6], Canny [2] và Prewitt [14] với 2 filter 3x3 để nhận diện cạnh ngang



Hình 2. Quy trình tổng quát về phương pháp phân lớp và phân tích ảnh X-quang phổi bằng mô hình GCNs

và cạnh dọc lần lượt là $\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ và $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$.

Kết quả sau khi thực hiện trích xuất đặc trưng cạnh được thể hiện ở hình 3.

Hình 3 cho thấy các cạnh được trích xuất bởi phương pháp Sobel cho ra các cạnh rõ ràng, ít nhiễu. Tuy nhiên, để có thể sử dụng đặc trưng cạnh được trích xuất bởi Sobel vào bên trong mô hình thì sẽ gây cản trở cho mô hình khá nhiều vì kết quả này chỉ nhận diện ra được các khung xương sườn và hoàn toàn mất đi các đặc trưng khác về phổi. Bên cạnh đó, các cạnh được trích xuất bởi Canny có quá nhiều chi tiết dư thừa (quá nhiều nhiễu) và do đó sẽ làm cho các đặc trưng quan trọng và nổi bật bên trong ảnh X-quang bị lu mờ đi. Điều này cũng sẽ khiến cho mô hình của chúng ta gặp nhiều khó khăn trong việc phân lớp một cách chuẩn xác. Cuối cùng, đối với các đặc trưng được trích xuất bởi bộ lọc Prewitt cho ra các kết quả khá tốt khi cân bằng được giữa kết quả Sobel và Canny, giữ được các đặc trưng quan trọng trong ảnh X-quang phổi như khung xương, phổi trái, phổi phải và các đặc điểm kính mờ. Do đó, chúng tôi quyết định chọn Prewitt là phương pháp trích xuất đặc trưng cạnh trong bài này.



Hình 3. Minh họa sự khác biệt giữa các phương pháp trích xuất đặc trưng cạnh truyền thống. Từ trái qua phải lần lượt là: ảnh gốc, ảnh được trích xuất bởi Sobel, ảnh được trích xuất bởi Canny và ảnh được trích xuất bởi Prewitt.

3.2. Cấu trúc đồ thị cho bài toán

Từ đặc trưng của cạnh sau khi đã được trích xuất bằng bộ lọc Prewitt, chúng tôi tiến hành tạo ra dạng đồ thị phù hợp cho bài toán và biểu diễn nó dưới dạng ma trận kẻ qua 3 bước:

- Bước 1: Để chọn được những điểm ảnh đóng vai trò quan trọng trong việc phân loại ảnh X-quang, chúng tôi chỉ chọn ra những pixel có độ sáng (grayscale intensity) lớn hơn hoặc bằng 128. Tiếp theo, chúng tôi biến các pixel đó trở thành các node của đồ thị với giá trị khởi tạo là độ sáng của pixel.
- Bước 2: Nối các pixel kề nhau và xem đó như là các cạnh của đồ thị.
- Bước 3: Chuẩn hóa giá trị của các node về giá trị trong đoạn $[0; 1]$ theo phương pháp chuẩn hóa Min-Max.

3.3. Cấu trúc mô hình mạng cho đồ thị

Trong bài báo cáo này, để giúp cho các node có thể giao tiếp thông tin với nhau một cách hiệu quả trên toàn bộ đồ thị, chúng tôi sử dụng cơ chế trao đổi thông tin như trong mô hình *mạng neural truyền thông tin* (Message-Passing Neural Network hay MPNN) [4].

Quá trình học các biểu diễn đặc trưng bên trong MPNN bao gồm 2 bước chính, đó chính là quá trình cập nhật thông tin (UPDATE) và quá trình chuyển hóa thông tin (READOUT).

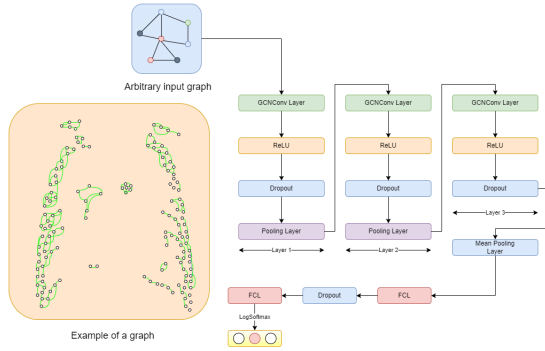
Đối với quá trình cập nhật thông tin bên trong MPNN, vector đặc trưng h_v của node v ($\forall v \in V$) được cập nhật tuần tự qua 2 bước nhỏ hơn được gọi lần lượt là quá trình tổng hợp thông tin (AGGREGATE) và quá trình cập nhật thông tin (UPDATE). Trong quá trình tổng hợp thông tin, đặc trưng của các đỉnh kề đối với đỉnh v đang xét sẽ được tổng hợp thành 1 vector đặc trưng. Vector đặc trưng được tổng hợp

này sẽ được gán trực tiếp vào đỉnh v đang xét. Toàn bộ quy trình tổng hợp thông tin này sẽ được thực hiện trong \mathcal{T} lần cập nhật và toàn bộ các đỉnh v nằm bên trong đồ thị V đều sẽ được cập nhật với cơ chế tương tự đối với mỗi lần cập nhật thứ t với $t \in [0; \mathcal{T}]$.

Đồ thị với các đặc trưng đã được tổng hợp trong \mathcal{T} lần sẽ được đưa qua một hàm kích hoạt Log-Softmax [3] để tổng hợp đặc trưng trên toàn đồ thị bên trong quá trình chuyển hóa thông tin. Trong đó, hàm kích hoạt Log-Softmax được biểu diễn theo công thức 1.

$$\text{LogSoftmax} = \log \left(\frac{e^{x_i}}{\sum_j (e^{x_j})} \right) \quad (1)$$

Kiến trúc mạng neural trong bài báo cáo này được minh họa ở hình 4 với 3 tầng: mỗi tầng bao gồm 1 phép Graph Convolutional Layer sử dụng cơ chế trao đổi thông tin là message passing, 1 hàm kích hoạt ReLU [1], 1 lớp Dropout và ở giữa các tầng là các lớp Pooling. Ngoài ra, kết quả sau khi đã được truyền qua ở tầng thứ 3 được đưa vào 1 lớp mean pooling layer, 2 lớp fully connected và 1 hàm Log-Softmax cuối cùng để đưa ra xác suất ảnh đó thuộc nhân nào trong các nhân đã được định nghĩa.



Hình 4. Kiến trúc GCNs được sử dụng

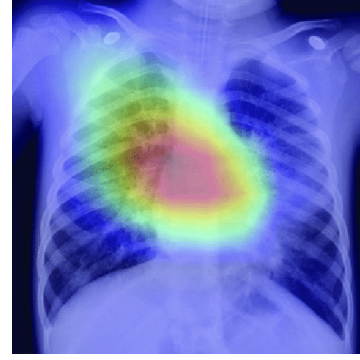
3.4. Trực quan hóa kết quả đầu ra

Để có thể ứng dụng cho các hệ thống CAD và giúp các y bác sĩ có thể phân tích kết quả thu được sau khi tiến hành dự đoán bằng mô hình đã được huấn luyện, chúng tôi trực quan kết quả sử dụng Grad-CAM [10] để thể hiện những thành phần khiến cho phổi bệnh nhân bị viêm. Kết quả sau khi trực quan hóa được thể hiện như ở hình 5.

4. Thực nghiệm

4.1. Độ đo

Để tiến hành đánh giá các mô hình, chúng tôi sử dụng các thang đo như độ chính xác, trị số tiên lượng chính xác



Hình 5. Trực quan hóa kết quả trả về của phương pháp, với màu xanh dương là không có yếu tố gây bệnh và màu đỏ là yếu tố chủ yếu gây nên bệnh.

(positive predictive value hay PPV), độ nhạy (sensitivity) và độ đo F1, trong đó:

$$PPV = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$F1_score = \frac{2 \times PPV \times \text{Sensitivity}}{PPV + \text{Sensitivity}} \quad (5)$$

4.2. Bộ dữ liệu

	Train	Val	Test	Total
Bacteria	2045	503	242	2790
Normal	1058	291	234	1583
Virus	1090	503	148	1741
Total	3193	1297	624	6114

Bảng 1. Độ phân bố trên từng lớp của tập dữ liệu.

Chúng tôi tiến hành thực nghiệm bộ dữ liệu "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification" [7] bao gồm 6114 ảnh X-quang phổi được gán với 1 trong 3 nhãn: bình thường (normal), viêm phổi do vi khuẩn (bacteria) và viêm phổi do virus (virus). Bên cạnh đó, chúng tôi tiến hành chia tập dữ liệu thành 3 tập train, validation và test với độ phân bố của dữ liệu trên từng lớp được thể hiện ở bảng 1.

4.3. Thiết kế thực nghiệm

Đối với mô hình GCNs, chúng tôi sử dụng thuật toán tối ưu là Adam [8] với hệ số học (learning rate) là $1e-4$, số batch trên 1 epoch là 64 và thực hiện huấn luyện trong 50 epochs.

Ngoài ra, chúng tôi cũng thí nghiệm thêm VGG16 [11] và ResNet50 [5] nhằm so sánh hiệu quả với GCNs. Chúng tôi thực hiện transfer learning trên tập dữ liệu ImageNet16-120, các kiến trúc được đồng bằng 75% lớp tích chập đầu. Sử dụng thuật toán Adam với hệ số học là $1e-3$. Chúng tôi cũng thực hiện huấn luyện hai kiến trúc trên trong 50 epochs, sử dụng learning rate scheduler với factor là 0.1, patience là 3 với hệ số học nhỏ nhất có thể được giảm là $1e-6$.

4.4. Phân tích kết quả

	GCNs	ResNet50	VGG16
Accuracy	96.75%	83.81%	80.29%
#Params	37,955	16,088,067	126,014,955
Training time(s)	9642 + 317	4715	5442

Bảng 2. So sánh tổng thể giữa Graph Convolutional Network (GCN), ResNet50 và VGG16.

		GCNs	ResNet50	VGG16
PPV	Bacteria	1.000	0.875	0.765
	Normal	0.979	0.982	0.952
	Virus	0.903	0.656	0.712
Sensitivity	Bacteria	0.949	0.954	0.958
	Normal	0.967	0.717	0.683
	Virus	1.000	0.837	0.736
F1 score	Bacteria	0.974	0.913	0.851
	Normal	0.973	0.829	0.796
	Virus	0.949	0.735	0.724

Bảng 3. So sánh kết quả giữa Graph Convolutional Networks (GCNs) với ResNet50 và VGG16.

Bacteria	222	5	7
Normal	0	234	8
Virus	0	0	139

Bacteria	231	1	10
Normal	11	168	55
Virus	22	2	124

Bacteria	232	7	3
Normal	33	160	41
Virus	38	1	109

Hình 6. Confusion matrix lần lượt từ trái qua phải của GCNs, ResNet50 và VGG16.

Trong bảng 2, ta có thể thấy độ chính xác của GCNs hoàn toàn vượt trội so với ResNet50 và VGG16. Không những thế, số lượng tham số của GCNs cũng nhỏ hơn các mạng CNNs rất nhiều. Tuy vậy, thời gian huấn luyện của GCNs lại nhiều hơn gấp đôi so với ResNet50 và VGG16. Điều này có thể được lý giải bởi việc GCNs cần tốn nhiều thời gian cho giai đoạn xây dựng đồ thị cho mỗi ảnh. Tuy nhiên, cần phải lưu ý là ResNet50 và VGG16 đạt được thời gian như vậy là bởi chúng tôi sử dụng áp dụng kỹ thuật transfer

learning. Nếu như huấn luyện từ đầu thì chưa chắc thời gian của ResNet50 và VGG16 để có thể đạt được độ chính xác trên có thể nhanh như vậy.

Bảng 3 cho thấy GCNs gần như vượt trội hoàn toàn trên các thang đo PPV, Sensitivity, F1 score so với các phương pháp CNNs. Thậm chí ở thang đo PPV trên Bacteria và Virus trên Sensitivity, GCNs còn đạt được kết quả 100%. Việc vượt trội trên nhiều thang đo càng khẳng định hơn nữa tính hiệu quả của GCNs. Ngoài ra, Confusion matrix được biểu diễn trong 6 càng cho thấy hơn nữa sự vượt trội của GCNs so với CNNs.

5. Kết luận

Chúng tôi đã sử dụng mô hình GCNs để giải quyết bài toán phân lớp và phân tích ảnh X-quang cho bài toán phân biệt các tác nhân gây nên bệnh viêm phổi ở người. Qua thực nghiệm, chúng tôi nhận thấy rằng phương pháp sử dụng mô hình GCNs đạt được các kết quả vượt trội về cả mặt hiệu năng trên nhiều thang đo và lượng tham số trong mô hình so với các phương pháp CNNs khác. Dù GCNs có hiệu năng vượt trội và thời gian huấn luyện thấp nhưng vẫn đòi hỏi kiến thức chuyên gia trong lĩnh vực của bài toán để có thể thiết kế ra những mô hình đồ thị phù hợp nhất. Hơn nữa, mặc dù thời gian huấn luyện của GCNs tương đối thấp, việc phát sinh ra các đồ thị tương ứng với từng ảnh trong tập dữ liệu vẫn là một quá trình tốn rất nhiều thời gian và công sức.

Lời cảm ơn

Chúng tôi xin gửi lời chân thành cảm ơn đến TS.Mai Tiến Dũng đã hướng dẫn, góp ý đồ án cũng như giảng dạy môn Thị giác máy tính nâng cao (CS331).

Tài liệu tham khảo

- [1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.
- [2] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.
- [3] Alexandre de Brébisson and Pascal Vincent. An exploration of softmax alternatives belonging to the spherical loss family, 2015.
- [4] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Message passing neural networks. In *Machine learning meets quantum physics*, pages 199–214. Springer, 2020.
- [5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [6] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988.
- [7] Daniel Kermanym, Kang Zhang, and Michael Goldbaum. Labeled optical coherence tomography (oct) and chest x-ray images for classification. 2018.
- [8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.
- [9] Cong Ma, Fan Yang, Yuan Li, Huizhu Jia, Xiaodong Xie, and Wen Gao. Deep human-interaction and association by graph-based learning for multiple object tracking in the wild. *Int. J. Comput. Vision*, 129(6):1993–2010, jun 2021.
- [10] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, oct 2019.
- [11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.
- [12] An Vo and Tan Ngoc Pham. Detecting covid-19 and pneumonia with chest x-ray images using deep convolutional neural networks. *Introduction to Computer Vision*, June 2021.
- [13] Peng Xu, Chaitanya K. Joshi, and Xavier Bresson. Multi-graph transformer for free-hand sketch recognition, 2019.
- [14] Lei Yang, Xiaoyu Wu, Dewei Zhao, Hui Li, and Jun Zhai. An improved prewitt algorithm for edge detection based on noised image. In *2011 4th International Congress on Image and Signal Processing*, volume 3, pages 1197–1200, 2011.