MATHEMATICAL MODELING AND COMPUTING, Vol. 11, No. 3, pp. 773–784 (2024)

Enhancing the vision graph model by elevating the precision diagnostics with attention and convolutions in medical imaging

Khaider Y., Rahhali D., En Nahnahi N.

Sidi Mohamed Ben Abdellah University, Faculty of Sciences Dhar EL Mehraz, LISAC Laboratory, Fez, Morocco

(Received 20 January 2024; Revised 16 August 2024; Accepted 17 August 2024)

The COVID-19 showed us that rapid and accurate diagnostics is a necessity. Therefore, researchers began to implement deep learning models that can help the doctors to reach faster and reliable results, but there are more development to be done. In our research paper, we introduced an innovative approach to enhance the Vision Graph model's accuracy for better results. Our method exploits the strength of the ConvMixer architecture and Attention mechanism. We start by utilizing Depthwise convolution and Pointwise convolution to capture spatial information in detail while reducing computational complexity of the model. Additionally, we added a hybrid attention module in which we combine the Convolution-based attention with Self-attention to boost the model's patterns identifying ability. We tested these enhancements on the COVID radiology dataset and demonstrated that our approach can help models be more accurate in their results.

Keywords: *deep learning; medical imaging; attention; CBAM; CNN; vision GNN.*

2010 MSC: 68T45, 68T10, 62P10, 05C85 **DOI:** 10.23939/mmc2024.03.773

1. Introduction

In the early stages of COVID-19, the virus spread rapidly between people without being able to be identified. Since then, researchers [1] have created mathematical models to comprehend the spread of COVID-19 and put strategies to contain the virus. Also analyzing the spreading of the virus with the help of a technique called "center of gravity" [2]. When the COVID-19 virus spread worldwide and the doctors in the medical field were overwhelmed and needed more staff to handle all the patients, researchers integrated deep learning model in a way to help doctors reach highly accurate and precise diagnostics results in a short amount of time to help more and more people. Deep learning model relies on analyzing a huge amount of data then extract features and patterns that prove helpful in identifying the targeted class or classes, an overview was conducted by [3] of the neural network models used on medical imaging in general.

In the field of medical imaging, specifically for COVID-19, we find that researchers have various contributions such as the COVID-Net architecture developed by Wang et al. [4], which is an open-source network, and one of the first models for the detection of COVID-19. Some articles as the "CheXNet" [5] that can produce accurate results in a short time, to reach that speed the "CheXNet" [5] optimizes the sum of unweighted binary cross entropy losses using a modified loss function. All these models including the "COVID-19 Severity Score" model proposed by Wynants et al. [6] and many more contributes greatly in aiding doctors to reach faster and reliable diagnoses, since their capabilities surpasses that of humans in terms of speed and compete with them in terms of accuracy. These models aid medical imaging, speeding up patient care while also enhancing our knowledge of the disease. This progress lays the foundation for more viable treatments and actions.

As the field of computer vision continues to evolve and more powerful GPUs that are becoming available to the public, Vision Transformers [7] models start to shine with their application of transformer modules on images by splitting the image into small patches then treat them as words of a sentence since transformers were originally designed for Natural Language Processing (NLP). Additionally, graph neural networks hold an important place in the deep learning field because GNNs are naturally invariant to rotation and translation since graphs do not involve these concepts and applying CNN to a graph can be challenging. Recently, a new model called Vision Graph Neural Network (Vision GNN) exemplified by ViG [8] architecture has emerged. It combines the vision model and the graph model by splitting the image into patches then process them with a graph model. This framework comprises two fundamental modules: the Grapher module, facilitating graph information aggregation and updating, and the FFN module, responsible for node feature transformation. The adaptability of ViG is due to its isotropic and pyramid architectures, varying in model size as detailed comprehensively in [8].

The biggest challenge for researchers is to build a model with high accuracy while having as low number of parameters as possible. While some researchers counteract this challenge by building more powerful models with high accuracy and high number of parameter. We aspire to combine a decent model with some methods and technique that has low parameters to enhance its accuracy. Therefore, in our paper we present a novel approach aimed at improving the accuracy of the ViG [8] model while adding a low number of parameters. Our method involves adding a layer that combines Depthwise convolution and Pointwise convolution, influenced by the ConvMixer architecture [9]. Afterwards, we add another layer that includes a mix of attention mechanisms. This combination is added after each block in the architecture to improve the model's function and its ability to manage complex visual information.

Our paper is organized as follows: Section 2 cites literature related to our study. In Section 3, we provide an overview of the methodologies employed. Section 4 offers a presentation of our experimental results and key findings. Lastly, in Section 5, we present a general conclusion of our work and discusses potential future research.

2. Related work

In this section, we shed light on prior research concerning of Graph Neural Networks and attention mechanisms, while citing some major contributions that left a huge mark in the deep learning field.

One of the foundational papers in the field of the Graph Neural Networks (GNNs) is "Neural Networks for Graph Recognition" [10] since it laid the foundation the GNNs by proposing neural network models for graph structure recognition and classification even though the computational power at that time were not powerful and publicly available.

The core concept underlying Graph Neural Networks (GNNs) is to adapt neural network architectures to operate effectively on graphs and they are designed to learn informative node-embeddings by aggregating and propagating information from neighboring nodes, which enables the capture of both local and global structural patterns within the graph [11]. More GNN architectures have emerged introducing unique approaches to message passing, node embedding updates and tailored solutions for diverse applications [14] and innovations that are more recent like GraphSAGE [13], Graph Attention Network (GAT) [12], and Graph Transformer Networks [19].

Attention mechanisms is an essential component in deep learning models because it offers versatile ways to focus on critical information in data. The "Scaled Dot-Product Attention" introduced by Vaswani et al. [20] enables models to evaluate the importance of different elements in a sequence, making it highly effective for sequence-to-sequence tasks. "Multi-Head Attention" also presented by Vaswani et al. [20], extends this concept by employing multiple attention heads that models to capture different relationships in data simultaneously. "Graph Attention" proposed by Velickovic et al. [12], empowers graph neural networks to attend to relevant nodes in a graph selectively. The "Self-Attention Mechanism" [20] has revolutionized natural language processing by allowing models to weigh words importance differently in a sentence which gives a better understand context and semantics. The combination of "Scaled Exponential Linear Unit (SELU) Activation with Self-Attention" [21] offers improved convergence properties and training stability. Finally, "Bilinear Attention" [22] introduced by Kim and Xing enhances models by incorporating attention into tree-structured neural networks

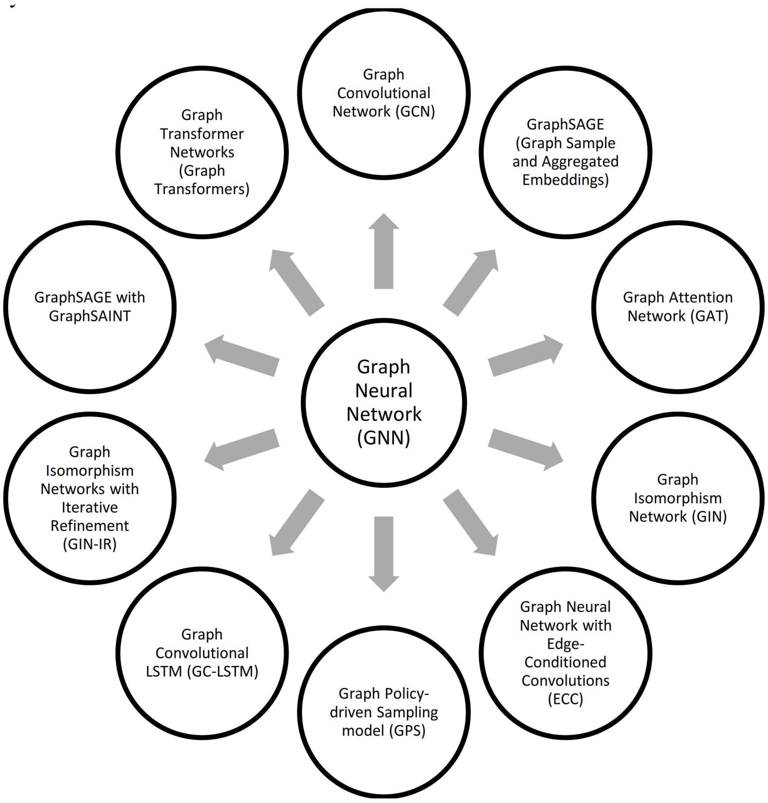
Graph Convolutional Network (GCN)
GraphSAGE (Graph Sample and Aggregated Embeddings)
Graph Attention Network (GAT)
Graph Isomorphism Network (GIN)
Graph Neural Network with Edge-Conditioned Convolutions (ECC)
Graph Policy-driven Sampling model (GPS)
Graph Convolutional LSTM (GC-LSTM)
Graph Isomorphism Networks with Iterative Refinement (GIN-IR)
GraphSAGE with GraphSAINT
Graph Transformer Networks (Graph Transformers)
Graph Neural Network (GNN)

Fig. 1. Various architectures of the Graph Neural Network (GNN) [11–18].

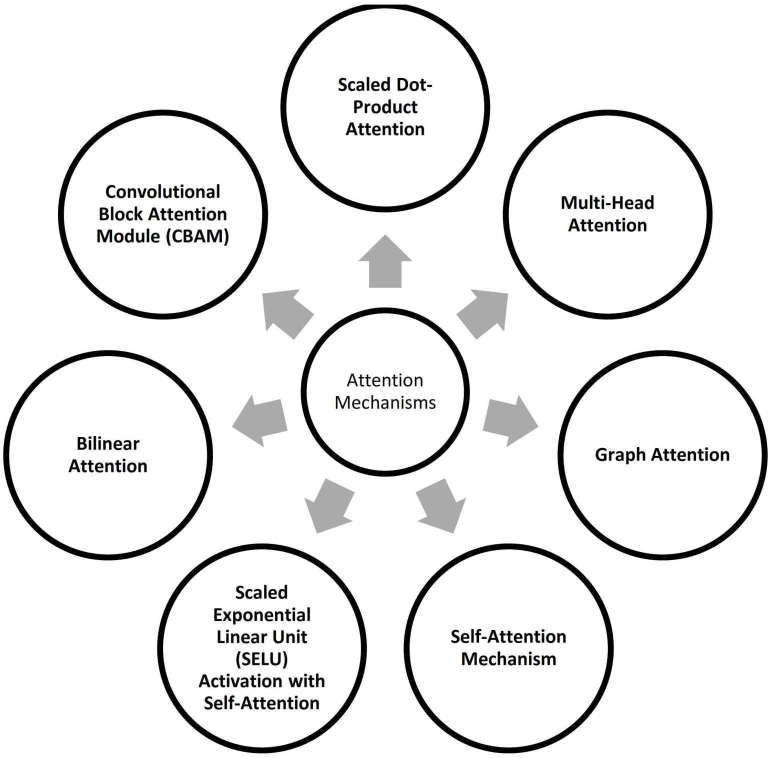
Scaled Dot-Product Attention
Multi-Head Attention
Graph Attention
Self-Attention Mechanism
Scaled Exponential Linear Unit (SELU) Activation with Self-Attention
Bilinear Attention
Convolutional Block Attention Module (CBAM)
Attention Mechanisms

Fig. 2. Different Attention Mechanisms [20–23].

enabling them to adaptively focus on informative regions within data. These attention mechanisms, each with their unique strengths and applications in advancing deep learning across various domains.

3. Methods and background

In this section, we explain our proposed method in detail, covering all the components and techniques used. We describe the key features of our approach, and their respective roles in improving the ViG model's performance. This detailed explanation aims to demonstrate how our method is structured and designed while ensuring clarity and transparency.

3.1. Proposed method

In this paper, we present two innovative techniques aimed at improving the performance of the Pyramid ViG model. In the first approach, we introduce a layer inspired by the ConvMixer architecture, which combines Depthwise convolution and Pointwise convolution. Following this, we incorporate a Convolution-Based Attention layer into the Pyramid ViG architecture. These two layers come after each block in the feature transform section.

In the second method, we retain the same architectural foundation as the first approach. However, in the attention layer we utilize a hybrid approach, combining Convolution-Based Attention with Self-Attention. This hybrid attention mechanism leverages the strengths of both techniques to enhance the model's capacity to attend to critical features and relationships within the input data. These two layers come also after each block in the feature transform section.

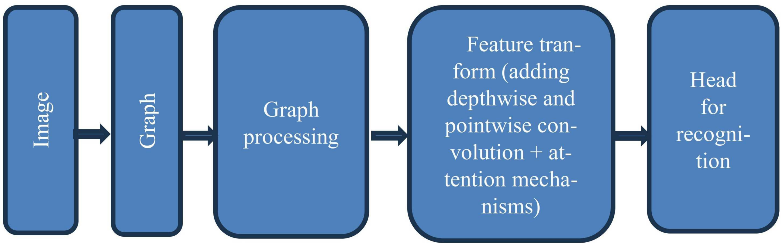
Image
Graph
Graph processing
Feature tran-form (adding depthwise and pointwise con-volution + at-tention mecha-nisms)
Head for recogni-tion

Fig. 3. Proposed method architecture.

These two methods represent our innovative contributions to the Pyramid ViG model, aiming to push the boundaries of its performance and capabilities in handling complex visual data. Through rigorous experimentation and evaluation, we demonstrate the efficacy of these enhancements and their potential impact on a range of visual tasks.

3.2. Vision GNN (ViG)

The authors [8] proposed a Graph Neural Network (GNN) architecture designed to extract rich representations from images by leveraging graph structures. For example, an image represented as a matrix $X \in R^{(N\times D)}$ with dimensions $H \times W \times 3$, they divided it into N patches, transforming each patch into a feature vector $X_i \in R^D$. The resulting feature matrix $X = [x_1, x_2, \ldots, x_N]$ is treated as a set of unordered nodes $= [v_1, v_2, \ldots, v_N]$. They introduced Edges E by establishing connections between nodes based on their K nearest neighbors.

The graph construction process ($G = G(X)$) leads to a graph $G = (V, E)$, where E denotes all the edges. Subsequently, they employed graph convolutional layers for graph-level processing. The mathematical formula of the graph convolution operation is

$$G_0 = F(G, W) = \mathrm{Update}(\mathrm{Aggregate}(G, W_{\mathrm{agg}}), W_{\mathrm{update}}). \tag{1}$$

The expressions W_{agg} and W_{update} represent the learnable weights of aggregation and update operations, respectively. The aggregation operation computes the representation of a node by aggregating features of its neighboring nodes, and the update operation merges the aggregated features. In particular, they used the max relative graph convolution because of its simplicity and efficiency:

$$g(\cdot) = x_{00}^i = \left[x^i, \max\left(\left\{x^j - x^i | j \in N(x^i)\right\}\right)\right], \quad h(\cdot) = x_0^i = x_{00}^i * W_{\mathrm{update}}, \tag{2}$$

$$h(\cdot) = x_0^i = x_{00}^i * W_{\mathrm{update}} \tag{3}$$

To address potential over-smoothing issues in deep graph convolutional networks, they introduced the Vision Graph Neural Network (ViG) block. The ViG block incorporates linear layers, non-linear activations, and a feed-forward network (FFN) on each node. The mathematical formula of the ViG block is

$$Y = \sigma\left(\mathrm{GraphConv}(X * W_{\mathrm{in}})\right) * W_{\mathrm{out}} + X. \tag{4}$$

Furthermore, they applied a FFN module to each node:

$$Z = \sigma(Y * W_1) * W_2 + Y. \tag{5}$$

This ViG block, consisting of Grapher and FFN modules is the fundamental building unit for creating the ViG network for visual tasks. The ViG network architecture demonstrates the capability to maintain feature diversity, as depicted in Figure 4, contributing to the learning of discriminative representations.

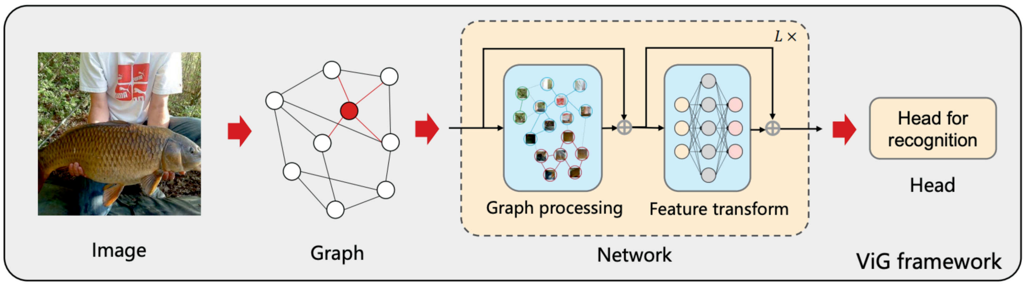

Fig. 4. ViG model architecture [8].

3.3. Attention mechanisms

We meticulously selected Convolution-Based Attention and Self-Attention, each integrated with skip connections, as key components in our model to strike a balance between capturing intricate dependencies and maintaining computational efficiency.

The inclusion of Convolution-Based Attention brings the advantages of convolutional operations into the attention mechanism in order for the model to compute attention scores in a more structured and localized manner. This process can be particularly advantageous when dealing with spatial data like images. Furthermore, with skip connections, we ensure that the model can leverage both local and global context effectively. The skip connections enable also the flow of information across different layers of the network, facilitating the exchange of knowledge and insights between features extracted at various levels of granularity.

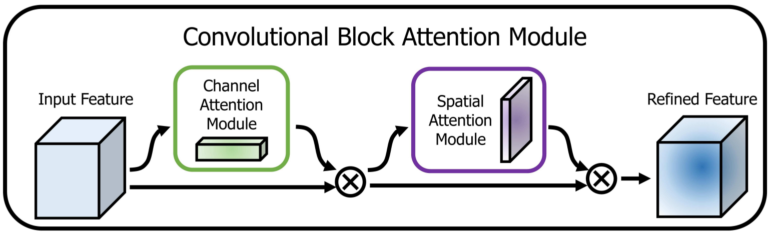

Fig. 5. Convolution-Based Attention (CBAM) [23, 24].

Self-Attention on the other hand, offers a remarkable capability to capture long-range dependencies and intricate relationships within the data. By utilizing multiple attention heads, the model can simultaneously attend to various parts of the input, thereby enhancing its ability to discern local and global patterns. Moreover, through skip connections, we integrate information from self-attention layers with features from previous layers.

This thoughtful integration of attention mechanisms and skip connections represents a strategic design choice aimed at maximizing the model's capability and adaptability. Convolution-Based Attention excels at localized feature processing while ensuring computational efficiency. Self-Attention provides a broader perspective to capture intricate dependencies. The skip connections serve as bridges that allow smooth communication while using both local and global information to improved performance of the model.

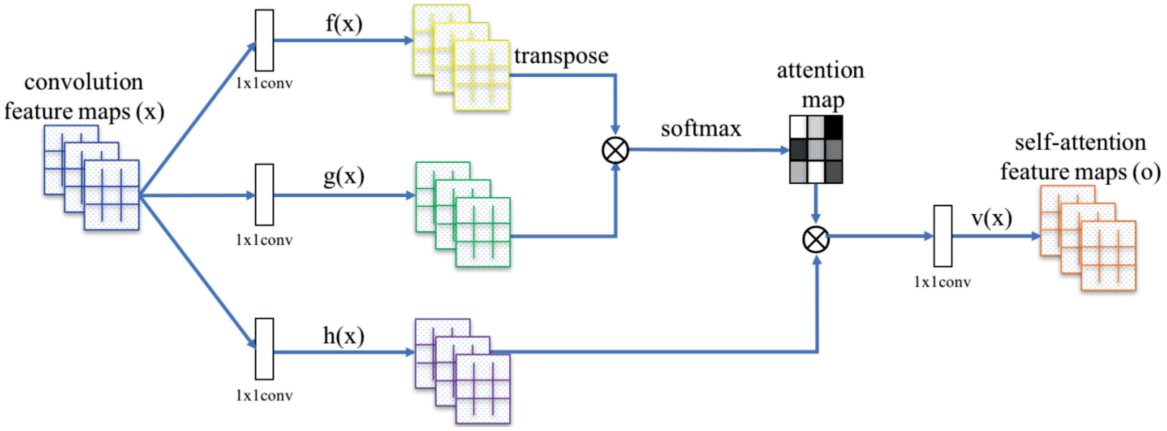

Fig. 6. Self-Attention Module from SAGAN [24].

3.4. ConvMixer

Depthwise convolution and Pointwise convolution play essential roles in modern convolutional neural networks (CNNs), especially in architectures like the ConvMixer. We find in the ConvMixer block that depthwise convolution is employed by applying a separate convolutional filter to each input channel to allow capturing local spatial information within each channel independently resulting in enhancing computational efficiency compared to traditional convolutional layers. It becomes particularly advantageous when using large kernel sizes, enabling the model to grasp broader spatial patterns.

Each input channel has its own Depthwise convolution applied to it:

— Input feature map with C input channels: I (height x width x_C);
— Depthwise convolutional filter: K (kernel height x kernel width x_1).

The equation of the output feature map for depthwise convolution O for each channel is as follows:

$$O_i = I_i * K. \tag{6}$$

The "$*$" represents the convolution operation applied to the input channel I_i with the corresponding depthwise convolutional filter K. After applying depthwise convolution to each channel separately, we obtain an intermediate feature map with the same height and width but with C channels, where each channel captures spatial information independently.

On the other hand, Pointwise convolution uses 1×1 convolutional filters to perform convolution at a single spatial location (pixel) across all channels. It has a crucial role in combining and aggregating information from different channels while reducing dimensionality.

Pointwise convolution uses 1×1 convolutional filters:

— Intermediate feature map with C channels from depthwise convolution: M (height x width x_C);
— Pointwise convolutional filter: P (kernel height x kernel width x_C).

The equation of the output feature map for depthwise convolution O for each channel is as follows:

$$O = M * P. \tag{7}$$

Here, "$*$" represents the convolution operation applied to the intermediate feature map M with the Pointwise convolutional filter P. Pointwise convolution combines information from all channels while reducing dimensionality. The result is a feature map with the same height and width but with a different number of channels, typically fewer channels, effectively mixing and refining the features extracted by depthwise convolution.

In the ConvMixer block [9], they begin by depthwise convolution then followed by pointwise convolution, both of those operations contain activation functions and Batch Normalization to extract and refine features at different levels of abstraction. These convolutional operations facilitate the transformation of input data into a meaningful representation when applied sequentially.

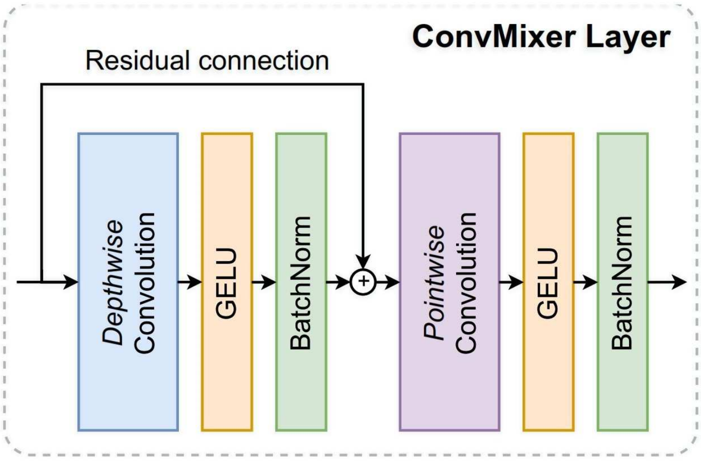

Fig. 7. Depthwise convolution and Pointwise convolution from the ConvMixer model [9].

4. Experimental results

In this section, we first define the dataset used in our experiments. Then, we provide an overview of the experimental settings. Next, we outline the evaluation metrics selected to assess our model's performance and present detailed findings that highlight the effectiveness of our method.

4.1. Dataset

Chowdhury et al. [25] in collaboration with medical practitioners and researchers from the University of Dhaka and Qatar University carefully constructed the dataset used in this study. This extensive dataset constitutes a valuable resource for the research community, offering a diverse collection of chest X-ray images. It encompasses 3616 samples that represent COVID-19 cases, alongside 10202 instances of Normal chest X-rays, 1345 cases of Viral Pneumonia and 6012 of Lung Opacity. To facilitate our experiments, we divided the entire dataset into two subsets: a training set comprising 80% of the data and a test set comprising the remaining 20%. Since there are some classes with low number of images, this partitioning allowed us to effectively train and evaluate our model on distinct sets of data, ensuring robustness and meaningful performance assessment.

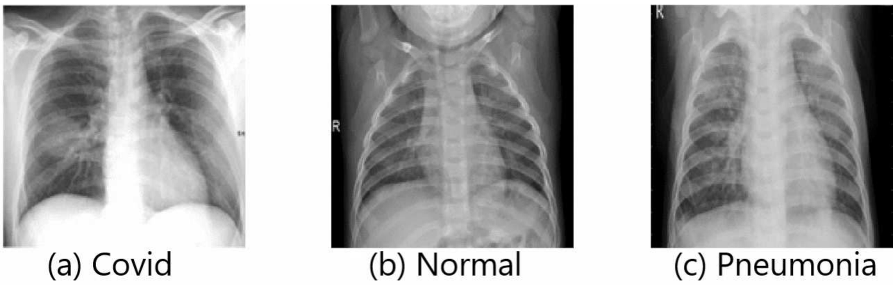
(a) Covid
(b) Normal
(c) Pneumonia

Fig. 8. Examples from the COVID-19 radiography database that contains 4 classes [25, 26].

4.2. Experimenting setting

In our experimental setup, we used the COVID-19 Radiography Database [25] as the dataset for our research. The dataset's rich collection of chest X-ray images allowed us to conduct comprehensive evaluations. To facilitate our computational tasks, we utilized the paid version of Google Colab, utilizing its cloud-based infrastructure for code execution. We stored and accessed the dataset through Google Drive. We chose python as a programming language for our code. We used multiple well-known libraries to help us in our experimentations such as pytorch that facilitate model building a experimenting with its wide range of algorithms and functions, Timm that contains a collection of pre-trained state-of-the-art models, and Math library for mathematical functions.

Table 1. Detailed settings of Pyramid ViG Ti. D: feature dimension, E: hidden dimension ratio in FFN, K: number of neighbors in GCN, $H \times W$: input image size. 'Ti' denotes tiny [8].

Stage	Output size	PyramidViG-Ti
Stem	$H/4 \times W/4$	$Conv \times 3$
Stage 1	$H/4 \times W/4$	$\begin{bmatrix} D = 48 \\ E = 4 \\ K = 9 \end{bmatrix} \times 2$
Downsample	$H/8 \times W/8$	Conv
Stage 2	$H/8 \times W/8$	$\begin{bmatrix} D = 96 \\ E = 4 \\ K = 9 \end{bmatrix} \times 2$
Downsample	$H/16 \times W/16$	Conv
Stage 3	$H/16 \times W/16$	$\begin{bmatrix} D = 240 \\ E = 4 \\ K = 9 \end{bmatrix} \times 6$
Downsample	$H/32 \times W/32$	Conv
Stage 4	$H/32 \times W/32$	$\begin{bmatrix} D = 384 \\ E = 4 \\ K = 9 \end{bmatrix} \times 2$
Head	1×1	Pooling & MLP
Parameters (M)	10.7	

We meticulously illustrated the architecture of our proposed model in Figure 3. We used the pyramid Vision GNN (PViG) and we chose the TI version as detailed in Table 1. Since we added the depthwise and pointwise convolutions and the attentions mechanisms to the same architecture of PViG they are included in the stage block in Table 1 and have the same input and output as illustrated in Table 2.

Within our training configuration, we incorporated several crucial hyper-parameters and settings to ensure effective model training. First, we used the paid GPU V100 in Google Colab and selecting it using the 'device' hyper-parameter. To optimize our training process, we implemented a learning rate scheduler, specifically the ReduceLROnPlateau scheduler. This dynamic scheduler adjusted the learning rate during training based on monitoring metrics, operating in 'max' mode with a patience of 3 epochs and employing a 'learning_rate_reduction_factor' of 0.2. Our training pipeline spanned 100 epochs, as defined by the 'num_epochs' hyperparameter. We also incorporated an early stopping mechanism with a patience threshold of 10 epochs, which determined the maximum number of epochs allowed without improvement in validation accuracy. Throughout training, we tracked the best validation accuracy using the 'best_valid_accuracy' variable. These meticulous settings and configurations ensured a well-structured and robust training process for our model.

Table 2. The parameters of the Deptwise and Pointwise block, and attention block.

	Depthwise and Pointwise block	Attention block
Parameters	Input Features: D Output Features: D Hidden Dimension: $E * D$	Input Features (both local and global): D Output Features: D Attention Features: $D/4$

4.3. Evaluation metric

In our experiment, we used several evaluation metrics to assess the performance of our model. We chose similar metrics as the ones used on the findings presented in the paper [26] for comparison purposes. The evaluation metrics we used are:

- Loss: measures the dissimilarity between the model's predicted outputs and the actual target labels. This loss quantifies the error between predicted and actual class probabilities.
- Accuracy: indicates the percentage of correctly classified samples out of the total samples in a dataset.
- ReduceLROnPlateau: is a learning rate scheduler that adjusts the learning rate based on the validation accuracy, even if it is not an evaluation metrics but it has huge impact on the model's performance.

4.4. Performance evaluation

In our experimentation, we conducted a thorough evaluation as depicted in Table 2. The reported accuracies in the table reflect the performance of various models on the COVID-19 radiography dataset as presented in the paper published by S. Kumar et al. [26]. Our primary research objective revolves around advancing the accuracy of COVID-19 detection specifically of the ViG model through the integration of attention mechanisms and ConMixer.

In the first proposed method we used Convolution-Based Attention and achieved an accuracy of 98.59% in detecting COVID-19 cases and other classes of the COVID-19 radiology dataset. This outcome underscores the effectiveness and potential of incorporating attention mechanisms into the detection process. Moreover, we extended our approach by including Self-Attention and we were able to achieve an even higher accuracy rate of 98.63%, this result demonstrates the significant benefits of combining these attention mechanisms with the ViG model.

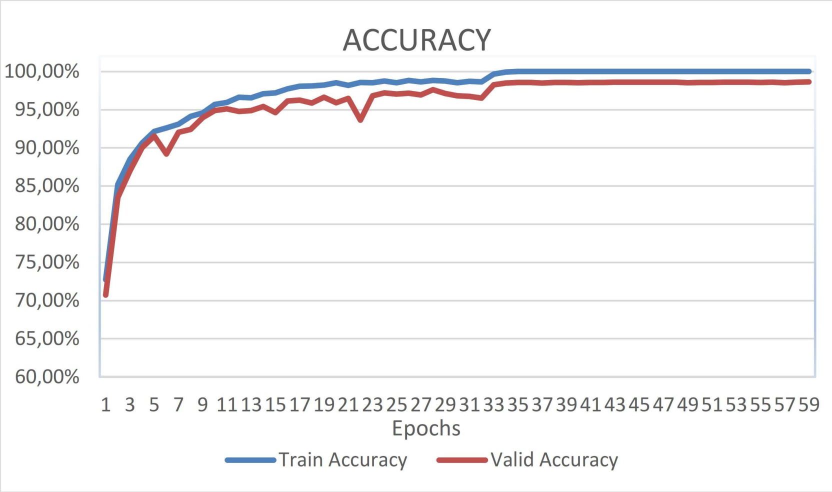

Fig. 9. Accuracy graph of our method with Convolution-Based Attention and Multi-Head Self-Attention.

Table 3. Accuracy of our proposed methods and other SOTA models [26].

Method	**Accuracy**
VGG16	83.47
VGG9	85.43
ResNet50	89.29
ResNet50V2	89.01
MobileNet	88.94
InceptionV3	91.11
Xception	91.53
InceptionResNetV2	88.06
ResNet50 + SVM	83.94
CovidGAN	89.68
COVIDX-Net	89.87
CAAD	91.78
Dark COVID-Net	87.02
CORO-Net	95.63
Covid-Net	93.06
CNN + LSTM	97.04
ResNet+PSO	95.46
DNN+XGBoost	88.62
TOTL	98.43
PyramidViG-Ti	98.50
Our method with only Convolution-Based Attention	**98.59**
Our method with Convolution-Based Attention and Multi-Head Self-Attention	**98.63**

Our method outperformed several well-known architectures such as VGG16, VGG19, ResNet50, and others. It outperformed also the baseline model of Res-Net50 with SVM, which achieved an accuracy of 83.94%. Additionally, "CORO-Net" achieved an impressive accuracy of 95.63%, and "CNN + LSTM" achieved 97.04%, which proves that our method surpasses several state-of-the-art models in terms of accuracy.

It is important to note that the parameter numbers for the models shown in Table 3, such as InceptionV3, Xception, VGG16, VGG19, ResNet50, ResNet50V2, and Inception-ResNetV2, were sourced from a reputable reference paper [26] in the field. These models such as VGG19 and VGG16 have exceptionally high parameter counts of 143 667 240 and 138 357 544 respectively. Moreover, our method with CBAM alone consists of 14 175 782 parameters while the model with the hybrid attention module that adds a Self-Attention to CBAM resulting in a total of 17 184 854 parameters offering a balance between model capacity and computational efficiency. This efficient utilization of parameters is a critical aspect of our method's success in achieving a remarkable accuracy of 98.63%. The reason for GNNs having low number of parameters than convolutional neural networks CNNs is due to the regularization provided by the relationships between elements in a graph-structured data, which allows GNNs to capture complex patterns efficiently. Furthermore, the irregular nature of graphs and the emphasis on capturing local patterns contribute to the reduced need for a large number of parameters.

Table 4. Number of parameters in our proposed methods and other SOTA models [26].

Method	Number of parameters
InceptionV3	23 851 784
Xception	22 910 480
InceptionResNetV2	55 873 736
VGG16	138 357 544
VGG19	143 667 240
ResNet50	25 636 712
ResNet50V2	25 613 800
PyramidViG-Ti	12 486 314
Our method with only Convolution-Based Attention	**14 175 782**
Our method with Convolution-Based Attention and Multi-Head Self-Attention	**17 184 854**

5. Discussion and conclusion

In this study, we started by a general introduction highlighting the importance of graph models and why Vision Graph Neural Networks (ViG) present a promising avenue of exploration. Then we presented various works related our study while mentioning the papers that made a remarkable make in deep learning field specially in GNNs and Attention mechanisms. Moreover, we illustrated our method and all its components in detail. Then, we showed the results of our experiments to prove in the end that the combination we proposed of the CBAM, self-attention, Depthwise and Pointwise convolutions pushes the limits of the ViG model's accuracy while having low number of parameters compared to the state-of-art models.

The results provided in Tables 3 and 4 show that the Depthwise and Pointwise and attention has on Vision Graph model can improve the results of the ViG models and probably other model too. Each of the components we combined with the ViG model extracts useful information that the model cannot in order to help the model improve its accuracy while having low parameter number.

For future works, we aim to apply our method to the other version of the ViG model and on other high performing models and architectures, in order to give a deeper and more complete comparative to better display the strength of our method. Furthermore, we envision adding more mechanisms to our methods while focusing on lowering the execution time and the number of parameters.

[1] Pawar D. D., Patil W. D., Raut D. K. Fractional-order mathematical model for analysing impact of quarantine on transmission of COVID-19 in India. Mathematical Modeling and Computing. **8** (2), 253–266 (2021).

[2] Yavorska O., Bun R. Spatial analysis of COVID-19 spread in Europe using "center of gravity" concept. Mathematical Modeling and Computing. **9** (1), 130–142 (2022).

[3] Khoroshchuk D., Liubinskyi B. B. Machine learning in lung lesion detection caused by certain diseases. Mathematical Modeling and Computing. **10** (4), 1084–1092 (2023).

[4] Wang L., Lin Z. Q., Wong A. COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Scientific Reports. **10**, 19549 (2020).

[5] Rajpurkar P., Irvin J., Zhu K., Yang B., Mehta H., Duan T., Ding D., Bagul A., Ball R. L., Langlotz C., Shpanskaya K., Lungren M. P., Ng A. Y. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. Preprint arXiv:1711.05225 (2017).

[6] Wynants L., Van Calster B., Collins G. S., Riley D. K., Heinze G., Schuit E., Bonten M. M. J., Damen J. A. A., Debray T. P. A., De Vos M., Dhiman P., Haller M. C. Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal. The BMJ. **369**, m1328 (2020).

[7] Dosovitskiy A, Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly S., Uszkoreit J., Houlsby N. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. International Conference on Learning Representations. 1–21 (2021).

[8] Han K., Wang Y., Guo J., Tang Y., Wu E. Vision GNN: an image is worth graph of nodes. NIPS'22: Proceedings of the 36th International Conference on Neural Information Processing System. 603 (2022).

[9] Trockman A., Kolter J. Z. Patches Are All You Need? Proceedings of the International Conference on Learning Representations (ICLR). (2022).

[10] Goldman S. A., Wong H. T. Neural Networks for Graph Recognition. Proceedings of the International Conference on Neural Networks (ICNN). (1996).

[11] Kipf T. N., Wellin M. Semi-Supervised Classification with Graph Convolutional Networks. Proceedings of the International Conference on Learning Representations (ICLR). (2017).

[12] Veličkovié P., Cucurull G., Casano A., Romero A., Liò P., Bengio Y. Graph Attention Networks. Proceedings of the International Conference on Learning Representations (ICLR). (2018).

[13] Hamilton W. L., Ying R., Leskovec J. Inductive Representation Learning on Large Graphs. Advances in Neural Information Processing Systems (NeurIPS). (2017).

[14] Xu K., Hu W., Leskovec J., Jegelka S. How Powerful are Graph Neural Networks? The Seventh International Conference on Learning Representations (ICLR). (2019).

[15] Wang Y., Sun Y., Liu Z., Sarma S. E., Bronstein M. M., Solomon J. M. Dynamic Graph CNN for Learning on Point Clouds. ACM Transactions on Graphics. **38** (5), 146 (2019).

[16] Zhang T., Liu Y., Chen X., Huang X., Zhu F., Zheng X. GPS: A Policy-driven Sampling Approach for Graph Representation Learning. Preprint arXiv:2112.14482 (2021).

[17] Yu B., Yin H., Zhu Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. 3634–3640 (2018).

[18] Zeng H., Zhou H., Srivastava A., Kannan R., Prasanna V. GraphSAINT: Graph Sampling Based Inductive Learning Method. International Conference on Learning Representations. (2020).

[19] Yun S., Jeong M., Kim R., Kang J., Kim H. J. Graph Transformer Networks. Advances in Neural Information Processing Systems (NeurIPS). (2019).

[20] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., Polosukhin I. Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS). (2017).

[21] Klambauer G., Unterthiner T., Mayr A., Hochreiter S. Self-Normalizing Neural Networks. Advances in Neural Information Processing Systems (NeurIPS). (2017).

[22] Tanno R., Arulkumaran K., Alexander D. C., Criminisi A., Nori A. Adaptive Neural Trees. Proceedings of the 36th International Conference on Machine Learning. **97**, 6166–6175 (2019).

[23] Woo S., Park J., Lee J. Y., Kweon I. S. CBAM: Convolutional Block Attention Module. Computer Vision – ECCV 2018. 3–19 (2018).

[24] https://blog.paperspace.com/attention-mechanisms-in-computer-vision-cbam/.

[25] https://www.kaggle.com/tawsifurrahman/covid19-radiography-database.

[26] Kumar S., Mallik A. COVID-19 Detection from Chest X-rays Using Trained Output Based Transfer Learning Approach. Neural Processing Letters. **55**, 2405–2428 (2022).

Удосконалення моделі графічного зору шляхом підвищення точності діагностики за допомогою уваги та згорток у медичній візуалізації

Хайдер Ю., Рахалі Д., Ен Нахнахі Н.

Університет Сіді Мохамеда Бен Абделлаха, Факультет наук Дхар-Ель-Мехраз, Лабораторія LISAC, Фес, Марокко

COVID-19 показав нам, що швидка та точна діагностика є необхідністю. Тому дослідники почали впроваджувати моделі глибокого навчання, які можуть допомогти лікарям досягати швидших і надійних результатів, але попереду ще багато чого потрібно зробити. У нашій дослідницькій роботі запроваджено інноваційний підхід до підвищення точності моделі Vision Graph для досягнення кращих результатів. Наш метод використовує силу архітектури ConvMixer і механізму уваги. Починаємо з використання глибинної згортки та поточкової згортки для отримання детальної просторової інформації, одночасно зменшуючи обчислювальну складність моделі. Крім того, додано гібридний модуль уваги, в якому поєднуємо увагу на основі згортки зі самоувагою, щоб підвищити здатність моделі ідентифікувати закономірності. Ці вдосконалення перевірено на радіологічних даних COVID та продемонстровано, що запропонований підхід може допомогти моделям отримати точніші результати.

Ключові слова: *глибоке навчання; медична візуалізація; увага; CBAM, CNN, зір GNN.*