2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico, December 5-8, 2023

Explainable Image Recognition with Graph-based Feature Extraction and Classification

Basim Azam
IIIS, Griffith University
Brisbane, Australia
basimazam0@gmail.com

Brijesh Verma
IIIS, Griffith University
Brisbane, Australia
b.verma@griffith.edu.au

Deepthi Praveenlal Kuttichira
IIIS, Griffith University
Brisbane, Australia
d.kuttichira@griffith.edu.au

***Abstract*— Deep learning models have proven remarkably adept at extracting salient features from raw data, driving state-of-the-art performance across diverse tasks. However, these models suffer from a lack of interpretability; they function as black boxes, obscuring the feature-level underpinnings of their predictions. Addressing this problem, our work presents an innovative framework that fuses the power of convolutional layers for feature extraction with the versatility of Graph Neural Networks (GNNs) to model relationships among neuron activations. Our framework operates in two phases: first, it identifies class-oriented neuron activations by analyzing image features, then these activations are encapsulated within a graph structure. The GNN leverages the relationships among these neuron activations to generate a final, interpretable classification. The proposed model not only matches, but at times exceeds, the accuracy of current leading models, all the while providing transparency via class-specific feature importance. This novel integration of convolutional and graph neural networks offers a significant step towards interpretable and accountable deep learning models.**

Keywords*— *Graph Neural Networks, Convolutional Neural Network

I. Introduction

The field of image classification has seen significant advancements in recent years, largely due to the advent of Deep Learning (DL) techniques. A critical aspect of these techniques is the extraction of robust and discriminative features from raw image data, which forms the basis for any successful image classification model. These features, which can range from simple color and texture information to complex patterns and objects, provide the necessary input for the model to learn and make accurate predictions.

Feature extraction and image classification have been extensively studied in the field of machine learning and computer vision. Traditional methods for feature extraction include Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and Speeded Up Robust Features (SURF) [1]. These methods, however, require manual engineering and are often not optimal for complex tasks such as image classification [2].

In recent years, CNNs have emerged as a powerful tool for image classification. CNNs are capable of automatically learning hierarchical feature representations from raw pixel data, which has led to significant improvements in image classification performance [3]. Convolutional layers extract features from the input raw data. The classifier component of the CNN architecture maps the features to output class labels. However, despite their effectiveness, CNNs are often criticized for their lack of interpretability. The internal workings of these models are often described as a "black box," making it difficult to understand how they arrive at their predictions [4].

CNNs have been widely adopted in the field of image classification due to their ability to automatically learn hierarchical feature representations from raw pixel data [5]. Despite their success, one of the main criticisms of CNNs is their lack of interpretability [6]. This lack of transparency can be problematic in certain applications where interpretability is crucial, such as medical imaging and autonomous driving [7].

Explainable AI (XAI) has emerged as a promising research direction to address the interpretability issue of deep learning models. XAI aims to make the decision-making process of AI models transparent and understandable to humans [8]. Various methods have been proposed to provide explainability , including saliency maps, layer-wise relevance propagation, deep Taylor decomposition [9], and Graph Neural Networks (GNNs) [10]. Towards this end, we aim to explore graph constructions from neuron activations to provide interpretability of features till the classification output [11]. A potential limitation is the difficulty of handling dynamic graphs, where the structure of the graph changes over time [20].

The research makes significant contributions to the field of computer vision in several notable ways:

- We introduce a novel framework that integrates the power of convolutional layers for feature extraction and robustness of GNNs to model relationship between neuron activations. The framework leads to an advanced model that provides a new level of interpretability.
- Unlike traditional deep learning methods, the proposed architecture provides explainability by tracing predictions back to the specific neurons that contributed to them.
- The proposed model transforms neuron activations into graph structures, which is significant to capture and expose the relationship between different features in an interpretable manner.
- A comprehensive comparison is presented with state-of-the-art models. The proposed model's better accuracy coupled with explainability, and transparency leverages the application of model in decision-making processes.

In the following sections, we will delve into the details of our proposed architecture, the experimental setup, and the results obtained, further demonstrating the effectiveness and advantages of our approach.

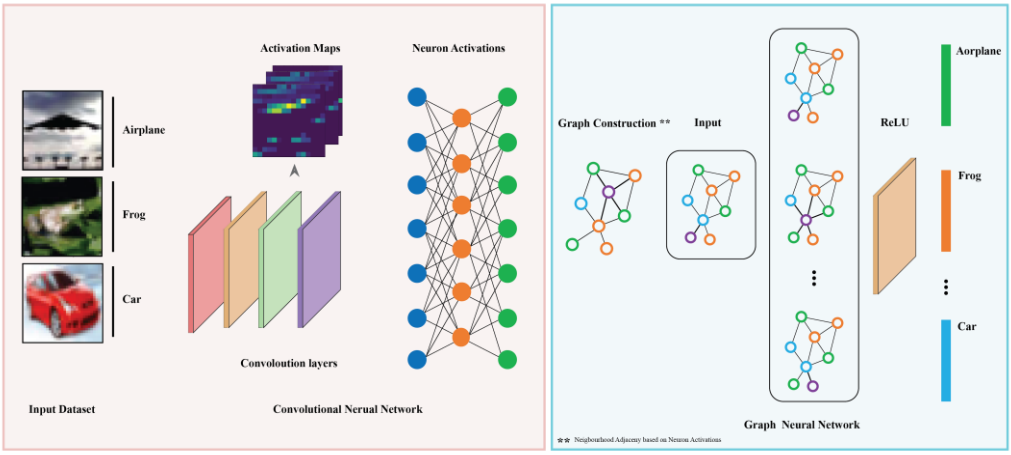

Figure 1 An overview of the proposed architecture providing explanations at each step using graph neural networks.

II. Proposed Methodology

The proposed method for feature extraction and image classification involves the use of Graph Neural Networks (GNNs). GNNs are capable of capturing the complex relationships between nodes in a graph, making them suitable for tasks that involve structured data [16]. In the context of image classification, each pixel in an image can be considered as a node in a graph, and the relationships between pixels can be modelled as edges in the graph. By applying GNNs to this graph representation of an image, we can effectively extract features that capture the spatial relationships between pixels, which can then be used for image classification [17].

The proposed architecture aims to leverage the strengths of both convolutional operations for local image structure extraction and graph-based operations for global relational reasoning. The overall model involves a feature extraction component, a graph construction component, a graph convolution component, and a classification component. These components are not isolated but interconnected, forming an end-to-end trainable system.

Given an input image $x_i \in \mathbb{R}^{H\times W\times C}$, where H, W, and C are height, width, and number of channels of the image, respectively. We utilize the deep neural network, which we denote as $F(\cdot)$, to transform the input image into a set of features defined by the mapping function F_i, where each feature corresponds to the output of a specific filter. This transformation can be represented as :

$$F_i = F(x_i; \theta_F)$$

Where $F_i \in \mathbb{R}^{H'\times W'\times C'}$ are the height, width, and number of channels of the feature maps, θ_F denotes the parameters for the feature extraction.

The next step is to construct a graph $G_i = V_i, A_i$ from feature maps F_i. Each nod $v_{i,j} \in V_i$ in the graph corresponds to a region of the image and is assigned a feature vector $f_{i,j}$ extracted from F_i. The edges of the graph represent the relationships between different regions of the image. The adjacency matric $A_i \in \mathbb{R}^{n\times n}$, where n is the number of nodes, is defined based on the relationship between the feature vectors of the nodes as f the nodes as

$$A_{i.j,k} = \frac{f_{i.j} \cdot f_{i,k}}{\|f_{i.j}\|_2 \|f_{i.k}\|_2}$$

where $A_{i,j,k}$ id the entry at the jth row and kth column of A_i, and $\cdot$ denotes the dot product.

We then perform graph convolution operation, denoted as $GC(.)$, to propagate information through the graph. The operation updates the node features based on their own features and the features of their neighbours capturing the relational information between different regions of the image. The node embedding N_i^l, after l graph convolution layers can be represented as:

$$N_i^l = \sigma(GC\ (N_i^{(l-1)} A_i; \theta_{GC}^l))$$

where $N_i^l \in \mathbb{R}^{n\times d_l}$ denotes the node embeddings after l layers, $N_i^o = V_i, \theta_{GC}^l$ denotes the parameters of the lth graph convolution layer, d_l is the dimension of the node embeddings after l layers, and $\sigma(\cdot)$ and is a non-linear activation function, such as the ReLU function.

The node embeddings N_i^L after L layers of graph convolution are then aggregated to generate a graph embedding $G_i \in \mathbb{R}^{d_L}$. This is done using the softmax function:

$$G_i' = softmax\left(\frac{1}{n}\sum_{j=1}^{n} N_{i,j}^L\right)$$

where $N_{i,j}^L$ is the jth row of N_i^L

The graph embedding G_i' is then passed through a classifier to predict the output label $y_{i,hat}$:

$$y_{i,hat} = argmax\ (C'(G_i'; \theta_C))$$

where θ_C denotes the parameters of the classifier, and the $argmax(\cdot)$ returns the index of the maximum value, indicating the predicted class.

The model is trained by optimizing the parameters $\theta = \{\theta_F, \theta_{GC}^l, \dots, \theta_{GC}^L, \theta_C\}$ to minimize the negative likelihood loss between the predicted labels and the true labels, which can be given as:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log\left(y_{i,hat}\,[y_i]\right)$$

where N is the number of samples, $y_{i,hat}[y_i]$, denotes the predicted probability of the true class for the ith sample, and the log is the natural logarithm.

III. Experimental setup

In this section, the description of dataset, evaluation metrics and details of hardware and software setup used for the experiments are discussed.

MNIST [26], CIFAR-10 [27], and Kaggle Cats and Dogs [28] dataset are used in this study. The evaluation metrics used to evaluate the proposed methodology are accuracy, precision, recall, and f1-score.

Our model achieved an accuracy of 99.26% on the MNIST dataset as indicated in Table 1, indicating that it correctly classified almost every image of the test set. The precision and recall were 99.26%, suggesting that our model was effective in identifying the correct digits and minimizing false positives and negatives. The F1 score, a measure of the model's overall performance, was 99.2% demonstrating a good balance between precision and recall.

Table 2 presents the performance of our model on the CIFAR-10 dataset. This dataset is more complex than MNIST, containing colour images of 10 different classes, including animals and vehicles. On the CIFAR-10 dataset, our model achieved an accuracy of 0.8737, meaning it correctly classified 87.37% of the images. The precision and recall were 87.35% and 87.37%, respectively. These values indicate that our model was effective in correctly identifying the classes of the images while minimizing false positives and negatives. The F1 score was 87.31%, indicating a balanced performance between precision and recall. Table 3 represent the scores achieved Kaggle cats and dogs dataset for each individual category separately, and overall

The input images are processed by the convolution layers to extract the highly important features. The model deems these features necessary for the classification. The fully connected layers describes the high-level features computed initially. The fully connected layers are base point to build graph structures, once the activations are further narrowed down, the proposed network transforms these activations into a graph structure. Each graph node corresponds to the neuron

Table 1 Class-wise performance evaluation on MNIST dataset.

Class/Metric	Precision	Recall	F1-Score
0	0.97	0.99	0.98
1	0.99	0.99	0.99
2	0.96	0.94	0.95
3	0.97	0.99	0.98
4	0.99	0.99	0.99
5	0.96	0.94	0.95
6	0.97	0.99	0.98
7	0.99	0.99	0.99
8	0.96	0.94	0.95
9	0.97	0.99	0.98
Overall	0.99	0.99	0.99
Accuracy		0.993	

Table 2 Class-wise performance evaluation on CIFAR10 dataset.

Class/Metric	Precision	Recall	F1-Score
Airplane	0.91	0.89	0.90
Automobile	0.93	0.91	0.92
Bird	0.87	0.89	0.88
Cat	0.91	0.89	0.9
Deer	0.93	0.91	0.92
Dog	0.87	0.89	0.88
Frog	0.91	0.89	0.90
Horse	0.93	0.91	0.92
Ship	0.87	0.89	0.88
Truck	0.91	0.89	0.90
Overall	0.87	0.87	0.87
Accuracy		0.87	

Table 3 Class-wise performance evaluation on Kaggle Cats and Dogs dataset

Class/Metric	Precision	Recall	F1-Score
Cat	0.91	0.89	0.90
Dog	0.93	0.91	0.92
Overall	0.91	0.91	0.91
Accuracy		0.91	

Table 4 Performance comparison of the proposed model with State-of-the-art approaches on MNIST and CIFAR-10 datasets.

Model Name	**Proposed Model**	**DNN5 [20]**	**FastSNN [23]**	**Tsetlin Machine [22]**	**Park et. al. [21]**	**CBoF & DSH [22]**	**Proposed Model**	**DNN5 [20]**	**CBoF & DSH [22]**	**CCN[26]**	**ResNet -8 [20]**	**CVPR_Class [24]**
Dataset	MNIST	MNIST	MNIST	MNIST	MNIST	MNIST	CIFAR10	CIFAR10	CIFAR10	CIFAR10	CIFAR10	CIFAR10
Accuracy	99.26	97.2	97.97	98.20	98.10	99.45	87.37	86.5	88.70	83.36	86.5	86.65

in dense layer and edges represent the relationship between neurons.

In Table 4, we present a comprehensive comparison of our proposed model with the state-of-the-art models (DNN5 [20], Fast SNN [23], Tsetlin Machine [22], Park et. al [21], and CCN [25]) on two popular datasets: MNIST and CIFAR10. The performance of each model is evaluated based on accuracy scores. The datasets consist of Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, and Truck. The in terms of accuracy scores are as 87.4%, 95.0%, 79.2%, 75.9%m 88.3%, 77.8%, 92.8%, 93.3%, and 91.8%. Figure 3 presents the confusion matrices computed on the MNIST and CIFAR-10 datasets. It can be interpreted from the results that almost all of the classes in both datasets have been classified accurately.

The components of our model include the Convolutional Neural Network (CNN), Neuron Activations, Graph Construction, and Graph Neural Network (GNN). For each class, we evaluated the model's performance with different combinations of these components. The performance metrics used in this study are Accuracy, Precision, Recall, and F1 Score. The scores achieved describe the effectiveness of proposed model along with explainability of the models.

IV. Conclusion

The paper presents a novel explainable graph neural network based approach to produce accurate classification labels. The notable contribution of the approach is to provide explanations to the features and build graph relationships based on neuron activations for each specific class. The proposed graph neural network component of the architecture then models the intricate relationship between the neuron activations. The architecture achieves overall accuracy of 99.26%, 91.44% and 87.37% on MNIST, Cat and Dog Dataset and CIFAR-10 datasets respectively. In comparison to the state-of-the-art approaches, the proposed architecture not only achieves better performance but provided explainability to the features, neuron activations, and the relationship between these activations. The experiments are continued to broaden the applicability of proposed approach, and to evaluate on more diverse image datasets.

V. Acknowledgements

This research was supported under Australian Research Council's Discovery Projects funding scheme (project number DP2101006401).

References

[1] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 886-893). IEEE.

[2] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110.

[3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[4] Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham

[5] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[6] Z. J. Wang, R. Turko, O. Shaikh, H. Park, N. Das, F. Hohman, M. Kahng, andD. H. P. Chau, "CNN explainer: Learning convolutional neural networks with interactive visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1396–1406, 2020.

[7] Holzinger, A., Biemann, C., Pattichis, C. S., & Kell, D. B. (2017). What do we need to build explainable AI systems for the medical domain?. arXiv preprint arXiv:1712.09923.

[8] Montavon, G., Samek, W., & Müller, K. R. (2018). Methods for interpreting and understanding deep neural networks. Digital Signal Processing, 73, 1-15.

[9] Montavon, G., Samek, W., & Müller, K. R. (2018). Methods for interpreting and understanding deep neural networks. Digital Signal Processing, 73, 1-15.

[10] A. Holzinger, B. Malle, A. Saranti, and B. Pfeifer, "Towards multi-modal causability with Graph Neural Networks enabling information fusion for explainable AI," in Information Fusion, vol. 71, pp. 28-37, 2021.

[11] R. G. KDDLab et al., "A survey of methods for explaining Black Box Models," ACM Computing Surveys, vol. 51, pp. 1-42, 2018.

[12] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," in IEEE Access, vol. 6, pp. 52138-52160, 2018.

[13] Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., ... & Sun, M. (2018). Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434.

[14] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems.

[15] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. arXiv preprint arXiv:1710.10903..10903.

[16] 13. Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., ... & Sun, M. (2018). Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434.

[17] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning System.

[18] 15. Liu, Z., Zhou, J., & Li, L. (2018). GeniePath: Graph Neural Networks with Adaptive Receptive Paths. arXiv preprint arXiv:1802.00910.

[19] Cui, P., Wang, X., Pei, J., & Zhu, W. (2018). A survey on network embedding. IEEE Transactions on Knowledge and Data Engineering, 31(5), 833-852.

[20] Pishchik, E. "Trainable Activations for Image Classification." Preprints.org 2023, 2023010463. doi: 10.20944/preprints202301.0463.v1.

[21] J. Park, J. Lee, and D. Jeon, "7.6 A 65nm 236.5 nJ/classification neuromorphic processor with 7.5% energy overhead on-chip learning using direct spike-only feedback," in 2019 IEEE International Solid-State Circuits Conference-(ISSCC), IEEE, 2019.

[22] N. Passalis and A. Tefas, "Training lightweight deep convolutional neural networks using bag-of-features pooling," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 6, pp. 1705-1715, 2018.

[23] Taylor, Luke, Andrew J. King, and Nicol Spencer Harper. "Robust and accelerated single-spike spiking neural network training with applicability to challenging temporal tasks." (2022).

[24] Wightman, Ross, Hugo Touvron, and Hervé Jégou. "Resnet strikes back: An improved training procedure in timm." arXiv preprint arXiv:2110.00476, October 2021.

[25] Jeevan, Pranav, and Amit Sethi. "Vision Xformers: Efficient attention for image classification." arXiv preprint arXiv:2107.02239, July 2

[26] Li Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research ," IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 141–142, Nov. 2012

[27] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 2009.

[28] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," *IEEE Xplore*, Jun. 01, 2012.