Improvements in Vision Graph Neural Networks

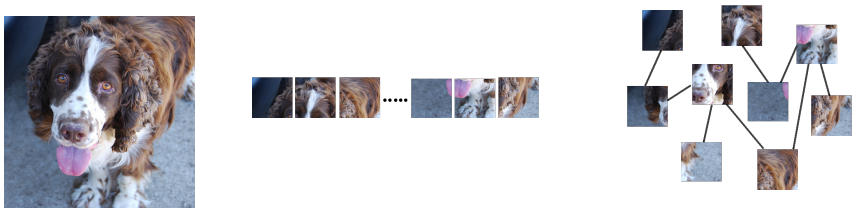
Bartłomiej Wójcik[0009−0003−5096−0755]
Arkadiusz Tomczyk[0000−0001−9840−6209]

Łódź University of Technology
Institute of Information Technology
al. Politechniki 8, 93-590 Łódź, Poland
bartlomiej.wojcik@dokt.p.lodz.pl
arkadiusz.tomczyk@p.lodz.pl

Abstract. *Vision Graph Neural Networks (ViG) have demonstrated superior performance in computer vision tasks compared to Vision Transformers (ViTs) and Convolutional Neural Networks (CNNs). ViG's adaptability to varying spatial relationships and irregular structures within images, coupled with its dynamic information aggregation, positions it as a robust solution for understanding of both fine-grained details and broader scene context. However, challenges such as vanishing gradient during training and the methods of defining edges need attention. In this work, we propose improvements to ViG, focusing on mitigating vanishing gradient issues, introducing novel edge generation strategies, and incorporating trainable edge weights.*
Keywords: *vision graph meural networks, edge generation, residual connections, adaptive adjacency matrix, graph convolution*

1. Introduction

Vision Graph Neural Networks [1] (ViG) emerge as powerful contenders for computer vision tasks, surpassing Vision Transformers [2] (ViTs) and Convolutional Neural Networks (CNNs) in flexible processing and seamless aggregation of global context. Graph Neural Networks (GNNs), designed to operate on graph-structured data, exhibit a remarkable ability to adapt to varying spatial relationships and irregular structures within images. Their flexibility enables the dynamic aggregation of information across nodes, facilitating effective propagation of context throughout the graph. Unlike the fixed receptive fields of CNNs, GNNs naturally handle complex structures of images. Comapring to ViT, they need not to



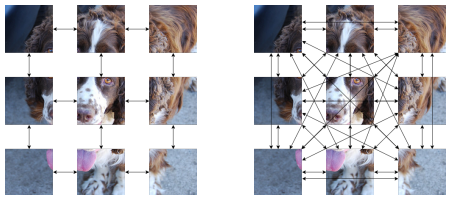
(a) (b) (c)

Figure 1: Comparison of input image representation: (a) - when using CNNs, we are constrained by the grid of pixels in the image, (b) - with the ViT architecture all patches are arranged in a sequence, which is further processed by transformer encoder layers, (c) - ViG allows for arbitrary dependencies to be set between patches.

considered a fully conntected graph and offer a variety of different graph convolutional operators.

The process of graph creation in ViG starts with dividing image into N patches (Figure 1). It is done using a simple CNN block, which transforms each patch into a D dimensional feature vector in the embedding space. All these features are then assigned respectively to a set of unordered nodes V. The next step is the addition of edges E between nodes. Authors of ViG paper generate edges between K nearest neighbours of nodes using the distance calculated in node embedding space. After computing the edges, the graph $G = (V, E)$ is constructed and graph operators are applied. Their goal is to update node embeddings propagating messages along edges from neighboring nodes. The final embeddings are aggregated and passed to a classifier block. Such a model allows for an end-to-end training (botch classical and graph convolutional layers are trained together).

Working with ViGs we have encountered vanishing gradient problem [3]. Backpropagating through these networks causes oversmoothing, eventually leading to features of graph vertices converging to the same value. Moreover, it seems not to be natural to create edges basing on node embeddings. This solution is expensive computationally, as it requires computation of all distances between patches. Additionally, it links only patches with similar embeddings, which means that patches of one object that are visually different will not be connected. Intuitively, however, information about node properties and graph structure should be rather a separate source of knowledge in considered image analysis tasks. In this work, we aim to improve mentioned above ViGs problems.



(a) (b)

Figure 2: Static edge creation: (a) - neighbour edges, (b) - complete edges.

2. Materials and methods

We demonstrate our improvements in effectiveness of ViG model on image classification task. To ensure the comparability of the experiments, in all of them we use similar architecture (Figures 3 and 4): the same CNN block to convert the image into patches, the same number of graph convolutions, global pooling and linear classifier. To compare different architectures we have used the Imagenette[1] dataset, featuring a subset of 10 easily distinguishable classes from ImageNet, containing: tenches, English springers, cassette players, chain saws, churches, French horns, garbage trucks, gas pumps, golf balls, and parachutes. The proposed novelties in ViG's architecture include: alternative static edge creation strategies, residual connections and trainable edge weights.

In contrast to the computationally demanding approach of generating edges based on the K nearest neighbors [1], we propose alternative strategies involving neighbor and complete versions, which provide compelling advantages in the context of graph construction for vision tasks (Figure 2). Rather than fixing on a specific number of neighbors basing on proximity, the generation of neighbor edges offers a more stable solution. Nodes establish connections based on their inherent spatial relationships. Moreover, although it is more expensive computationally, the incorporation of complete edges augments the graph with a global perspective allowing nodes to be linked. This strategy captures long-range dependencies addressing the limitation of the traditional approach, which tends to focus on local relationships. This methods is especially useful for layers using attention, because it allows to assign the weights for each edges and focus only on most significant connections.

[1] `https://github.com/fastai/imagenette`

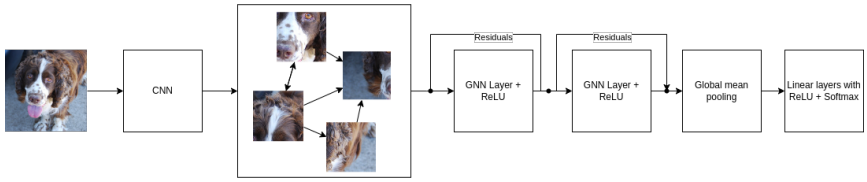


Figure 3: ViG architecture with residual connections.

One of the methods of solving vanishing gradient problem in classical CNNs was the usage of residual connections. It enabled also the creation of deeper architectures. Inspired by this success residual connections were added also in GNNs ([3]). Although, we are not building a deep architecture, in our work we have also used them with graph convolutional layers, which is depicted in Figure 3. This modification of ViG model not only solved the training problems, but it improved model performance as well.

Specification of edges E in a graph G is equivalent with building a graph adjacency matrix A. In this matrix 0 represents no connection between nodes and 1 represents an existing edge. Our next improvement of ViG's architecture assumes that adjacency matrix can be trainable. Its elements can have any value from interval $[0, 1]$ and thus can be treated as edge weights. To achieve that we incorporate an additional CNN block followed by element-wise sigmoid function responsible for dynamic generation of that adjacency matrix. This architecture is presented in Figure 4. To avoid situations where all the elements of A are equal to 0 or 1, we have modified the loss function adding to standard cross-entropy loss a regularization term. This term was equal to $-\lambda \cdot \sigma(A)$ where σ denotes a standard deviation and λ is a regularization coefficient set to 0.25 experimentally. The trained, in this way, edge weights are used by modified graph convolutional layers - every message sent through an edge is multiplied by corresponding weight. Thanks to that we dynamically (depending on the input image) control the influence of different nodes (image patches) on each other.

3. Experiments and results

All our experiments were conducted using three different graph convolutional layers: GraphSAGE (SAmple and aggreGatE) ([4]), Graph Attention Networks (GAT) ([5]) and Graph Transformer ([6]), each model consisted of two GNN lay-

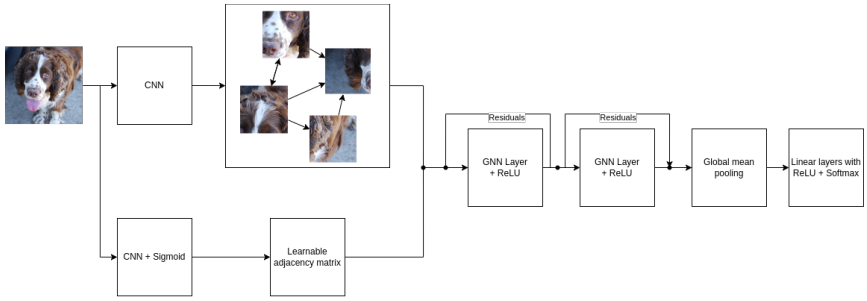


Figure 4: ViG architecture with trainable edge weights.

Table 1: Results of experiments (C - complete edges, N - neighbor edges, R - residuals, TE - trainable edges).

Model	**Accuracy**	**Model**	**Accuracy**
Ours SAGE C-R-TE	0.871 ± 0.003	Ours Trans C	0.848 ± 0.006
Ours Trans C-R-TE	0.867 ± 0.006	Ours GAT N	0.845 ± 0.013
Ours GAT N-R	0.866 ± 0.008	Ours GAT C	0.845 ± 0.009
Ours GAT C-R	0.865 ± 0.009	ViG	0.840 ± 0.004
Ours GAT C-R-TE	0.863 ± 0.005	CNN	0.832 ± 0.021
Ours SAGE C-R	0.860 ± 0.004	Ours Trans N	0.825 ± 0.017
Ours SAGE N-R	0.857 ± 0.012	Ours SAGE N	0.825 ± 0.007
Ours Trans C-R	0.856 ± 0.003	Ours SAGE C	0.822 ± 0.009
Ours Trans N-R	0.852 ± 0.011	ViT	0.746 ± 0.005

ers. ViG and ViT models were reproduced as in the original publications. CNN model consisted of two classic convolutional layers instead of GNN layers. All results are the average of the three trials. Moreover, each group of trials was initialized with the same set of seeds. Every experiment was trained for a maximum of 100 epochs with early stopping on validation accuracy with patience of 20 epochs. Then, the epoch with the best validation accuracy was used for testing. As Imagenette only contains a train and validation dataset, the train dataset of Imagenette was split with fixed seed into train (8500 samples) and validation (969 samples) datasets, and the original validation dataset was used as the test (3925 samples) dataset.

The results presented in Table 1 reveal several notable findings. Firstly, our proposed modifications to the ViG model, particularly those incorporating complete trainable edges (C-R-TE), have led to significant improvements in accuracy compared to traditional convolutional neural networks (CNN) and the Vi-

sion Transformer (ViT) on the Imagenette dataset. This suggests that leveraging graph-based structures and integrating them into convolutional architectures can effectively enhance performance in image classification tasks. Furthermore, the performance of different graph convolutional layers varied, but all the layers indicated the efficacy of this approach in capturing global graph structures for image feature extraction. On the contrary, the ViT model exhibited comparatively lower accuracy on the Imagenette dataset. We hypothesize that this inferior performance may be attributed to the model's reliance on self-attention mechanisms, which may struggle to effectively capture spatial information in smaller datasets like Imagenette. Future investigations on larger datasets will be essential to validate this hypothesis and gain deeper insights into the effectiveness and robustness of different model architectures. In summary, our experiments not only validate the efficacy of graph-based models, particularly those incorporating complete learnable edges, but also highlight the importance of structural information in capturing relationships for image classification tasks. These findings provide valuable insights for the development of more advanced and effective models in computer vision.

4. Summary

Presented results demonstrate the efficacy of proposed modifications to the ViG model, surpassing both traditional CNN and ViT in accuracy on the Imagenette dataset. Graph-based models, particularly those incorporating complete learnable edges, exhibit superior performance, highlighting their potential in image classification tasks. Furthermore, we hypothesize that the inferior performance of the ViT model may be attributed to the small size of the Imagenette dataset. In future investigations, we will validate all methods on larger datasets to gain deeper insights into their effectiveness and robustness.

References

[1] Han, K., Wang, Y., Guo, J., Tang, Y., and Wu, E. Vision gnn: An image is worth graph of nodes. *Advances in Neural Information Processing Systems*, 35:8291–8303, 2022.

[2] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszko-

reit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

[3] Li, G., Müller, M., Thabet, A. K., and Ghanem, B. Can GCNs go as deep as CNNs? *CoRR*, abs/1904.03751, 2019. URL http://arxiv.org/abs/1904.03751.

[4] Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017. URL http://arxiv.org/abs/1706.02216.

[5] Brody, S., Alon, U., and Yahav, E. How attentive are graph attention networks? *CoRR*, abs/2105.14491, 2021. URL https://arxiv.org/abs/2105.14491.

[6] Shi, Y., Huang, Z., Wang, W., Zhong, H., Feng, S., and Sun, Y. Masked label prediction: Unified massage passing model for semi-supervised classification. *CoRR*, abs/2009.03509, 2020. URL https://arxiv.org/abs/2009.03509.