

SAG-ViT: A Scale-Aware, High-Fidelity Patching Approach with Graph Attention for Vision Transformers

Shravan Venkatraman Jaskaran Singh Walia Joe Dhanith P R
Vellore Institute of Technology, Chennai, India

shravan.venkatraman18@gmail.com

karanwalia2k3@gmail.com

joedhanith.pr@vit.ac.in

Abstract

Image classification is a computer vision task where a model analyzes an image to categorize it into a specific label. Vision Transformers (ViT) improve this task by leveraging self-attention to capture complex patterns and long-range relationships between image patches. However, a key challenge for ViTs is efficiently incorporating multi-scale feature representations, which is inherent in CNNs through their hierarchical structure. In this paper, we introduce the Scale-Aware Graph Attention Vision Transformer (SAG-ViT), a novel framework that addresses this challenge by integrating multi-scale features. Using EfficientNet as a backbone, the model extracts multi-scale feature maps, which are divided into patches to preserve semantic information. These patches are organized into a graph based on spatial and feature similarities, with a Graph Attention Network (GAT) refining the node embeddings. Finally, a Transformer encoder captures long-range dependencies and complex interactions. The SAG-ViT is evaluated on benchmark datasets, demonstrating its effectiveness in enhancing image classification performance. Our code and weights are publicly available at <https://github.com/shravan-18/SAG-ViT>.

1. Introduction

The field of image classification has experienced significant advancements with the introduction of deep learning architectures. CNNs have long been the foundation for image classification tasks due to their proficiency in capturing local spatial hierarchies through convolutional operations [9]. However, their inherent limitations in modeling long-range dependencies restrict their ability to fully exploit global contextual information within images [12]. The introduction of Vision Transformers (ViT) [4, 18] has opened new avenues by leveraging self-attention mechanisms to model

global relationships within images. ViTs treat images as sequences of patches (tokens) and have demonstrated competitive performance compared to traditional CNNs. Despite their success, ViTs often require large-scale datasets for effective training and may overlook fine-grained local details due to their fixed-size patch tokenization [17].

Recent research has highlighted the importance of multi-scale feature representations in enhancing ViTs' performance across various vision tasks [1]. Multi-scale approaches enable models to capture objects and patterns of varying sizes, providing a more comprehensive understanding of the image content. While CNNs inherently capture multi-scale features through hierarchical layers, integrating this capability efficiently into Transformer-based models remains a challenge.

To handle this challenge, we propose a novel Transformer-based framework called Scale-Aware Graph Attention Vision Transformer (SAG-ViT). Our model begins by extracting rich, multi-scale feature maps from input images using a pre-trained EfficientNet backbone [16]. We then divide these feature maps into patches, preserving high-level semantic information and reducing information loss compared to raw image patching. We then construct graphs where each node represents a feature map patch, and edges are established based on spatial adjacency and feature similarity using a k-connectivity scheme. This graph captures both local and global relationships among image regions. A Graph Attention Network (GAT) [19, 24] processes the graph, dynamically focusing on the most relevant patches. The enriched node embeddings are then passed through a Transformer encoder, which captures long-range dependencies and complex interactions.

Our contributions are summarized as follows:

- We introduce a patching mechanism that operates on CNN-derived feature maps, retaining rich semantic information and efficiently capturing multi-scale features.
- A k-connectivity and similarity-based edge weighting

scheme is developed in the proposed Transformer architecture to construct graphs that model intricate spatial relationships between patches.

- We employ a GAT Network to process the information-rich graph embeddings to effectively model both local and global dependencies within images.
- We validate our method on multiple benchmark datasets across different domains, demonstrating higher performance compared to other transformer-based approaches.

The remainder of this paper is organized as follows: Section 2 reviews related work on graph transformers, attention mechanisms, multi-scale feature embedding, and their integration in image classification. Section 3 details our proposed method, including the architecture and graph construction process. Section 4 presents the experimental setup, datasets, and evaluation metrics. Section 5 discusses the results, and Section 6 concludes the paper.

2. Literature Survey

In this section, we review relevant literature on vision transformers, multi-scale feature representation, and graph neural networks for image classification.

2.1. Vision Transformers for Image Classification

Transformer-based models have gained significant attention in computer vision, initially popularized by the Vision Transformer (ViT), which treats images as sequences of patches and uses self-attention to capture global dependencies, achieving competitive results with CNNs for image classification [24]. However, ViT models often require large datasets and substantial computational resources, limiting their accessibility.

To improve data efficiency, DeiT leverages distillation and data augmentation, enabling ViTs to perform well on smaller datasets [10]. T2T-ViT [25] introduces a Tokens-to-Token transformation to better capture local structures, addressing ViT’s limitation of naive tokenization. The Perceiver model uses an asymmetric attention mechanism to distill large inputs into a compact latent space, allowing it to scale effectively for high-dimensional data [8]. Similarly, PVT and CvT incorporate pyramid-like structures into transformers, merging CNN-like multi-scale processing with transformer advantages for richer feature extraction [23].

The Swin Transformer introduces a shifting window approach to self-attention, efficiently capturing both local and global contexts while maintaining manageable complexity, especially for dense tasks like segmentation and detection [11]. These models highlight a growing trend toward integrating multi-scale representations to improve vision transformers’ ability to capture both fine-grained details and long-range dependencies.

2.2. Multi-Scale Feature Representation

Multi-scale feature representations are critical for recognizing objects and patterns at varying scales [1]. CNNs naturally capture multi-scale features through their hierarchical layers and receptive fields [9]. Techniques such as feature pyramid networks [10] and multi-branch architectures [2] have been proposed to enhance multi-scale learning in CNNs.

In the context of transformers, incorporating multi-scale features remains challenging due to the fixed-size patch tokenization. CrossViT [1] introduces a dual-branch transformer architecture that processes image patches of different sizes in separate branches, fusing them using cross-attention mechanisms. This approach effectively captures both fine-grained details and global context.

2.3. Graph Neural Networks for Image Classification

Graph Neural Networks have gained attention for their ability to model relational data. In image classification, representing images as graphs allows for capturing spatial relationships between different regions [24]. Nodes can represent super pixels or patches, and edges encode similarities or spatial connections. Constructing graphs directly from raw images can lead to information loss due to the reduction in spatial resolution [26]. By constructing graphs from CNN-derived feature maps, richer semantic information can be retained [5]. This approach enhances the modeling of complex spatial dependencies crucial for accurate classification.

Graph Attention Networks extend the concept of attention mechanisms to graph-structured data [19]. GATs compute attention coefficients for neighboring nodes, allowing the network to focus on the most relevant connections. This dynamic weighting improves the learning of node representations by emphasizing important relationships. Incorporating GATs in image classification enables the modeling of both local and non-local dependencies [22]. When combined with multi-scale feature representations, GATs can effectively capture intricate patterns within images.

2.4. Hybrid Models

Recent studies suggest that combining transformer and convolutional layers into a hybrid architecture can harness the strengths of both approaches. BoTNet [15] modifies self-attention in the final three blocks of ResNet to integrate both architectures. The CMT [7] block incorporates depthwise convolutional layers for local feature extraction, alongside a lightweight transformer block. CvT [11] places pointwise and depthwise convolutions before the self-attention mechanism to enhance performance. LeViT [6] replaces the patch embedding block with a convolutional stem, enabling faster inference for image classification. MobileViT [13]

combines Transformer blocks with the MobileNetV2 [14] block to create a lightweight vision transformer. MobileFormer [3] bridges CNNs and transformers in a bidirectional manner to capitalize on both global and local features.

3. Method: SAG-ViT

In this section, we detail our proposed approach to enhance transformer performance for image classification through a multiscale feature embedding and high-fidelity graph attention-based patching. During graph construction in graph transformers, spatial hierarchies are often lost or insufficiently represented, especially as redundant or less relevant areas dilute the image’s contextual representation. To overcome this limitation, we propose a novel framework that captures both local and global dependencies while preserving rich semantic information. Specifically, we begin by outlining our high-fidelity feature map patching strategy (§3.1). We then detail the graph construction methodology based on k -connectivity and feature similarity (§3.2). Finally, we explain the integration of Graph Attention Networks with Transformer encoders (§3.3). Figure 1 illustrates the network architecture of our proposed Scale-Aware Vision Transformer with Graph Attention (SAG-ViT).

3.1. High-Fidelity Feature Map Patching

We initiate the processing pipeline by extracting high-fidelity patches from feature maps generated by a lightweight convolutional backbone. By operating on feature maps rather than raw images, we retain higher-level semantic information. We process the input image $I \in \mathbb{R}^{H \times W \times C}$ through a deep CNN to exploit its compound multiscale feature scaling for receptive fields and efficient convolution paths, yielding a feature map $F \in \mathbb{R}^{H' \times W' \times D}$, where $H' = \frac{H}{s}$, $W' = \frac{W}{s}$, and D denotes the depth of the feature channels with stride s .

To preserve detailed and multi-scale semantic information, we partition the feature map F into non-overlapping patches $P_{i,j} \in \mathbb{R}^{k \times k \times D}$, where k is the spatial dimension of each patch. Formally, the patch extraction is defined as:

$$P_{i,j} = F[i \cdot k : (i+1) \cdot k, j \cdot k : (j+1) \cdot k, :], \quad (1)$$

for all $i \in \{0, \dots, \frac{H'}{k} - 1\}$ and $j \in \{0, \dots, \frac{W'}{k} - 1\}$.

This operation can be represented using an unfolding operator \mathcal{U}_k :

$$\mathcal{U}_k(F) = \{P_{i,j} \mid P_{i,j} = F[i \cdot k : (i+1) \cdot k, j \cdot k : (j+1) \cdot k, :]\}. \quad (2)$$

for all $i = 0, \dots, \frac{H'}{k} - 1$ and $j = 0, \dots, \frac{W'}{k} - 1$.

where $\mathcal{U}_k : \mathbb{R}^{H' \times W' \times D} \rightarrow \mathbb{R}^{\frac{H'}{k} \times \frac{W'}{k} \times k \times k \times D}$. Each patch $P_{i,j}$ is then vectorized into a feature vector $p_{i,j} \in$

$\mathbb{R}^{k^2 D}$ by flattening the spatial and channel dimensions:

$$p_{i,j} = \text{vec}(P_{i,j}). \quad (3)$$

This results in a collection of patch vectors:

$$\mathcal{P} = \bigcup_{i=0}^{\frac{H'}{k}-1} \bigcup_{j=0}^{\frac{W'}{k}-1} \{p_{i,j}\}. \quad (4)$$

By extracting patches directly from the feature map F , we leverage the high-level abstractions learned by the CNN. This approach ensures that each patch $P_{i,j}$ encapsulates rich semantic information, capturing both local patterns and contextual relationships within the image. Moreover, extracting patches from the reduced spatial dimensions $H' \times W'$ leads to fewer patches \mathcal{P} , decreasing computational complexity while maintaining essential information.

The vectorized patches $p_{i,j}$ serve as nodes in the subsequent graph construction phase. The high-dimensional feature vectors facilitate the capture of intricate relationships between patches when constructing edges based on similarity measures. Additionally, the non-overlapping nature of patch extraction ensures that each patch maintains its spatial locality within the feature map, preserving the inherent spatial structure essential for accurate image classification.

This mathematical formulation ensures that the patch extraction process is both systematic and scalable, facilitating efficient downstream processing in the graph-based classification pipeline.

3.2. Graph Construction Using k -Connectivity and Similarity-Based Edges

Once the patches $\mathcal{P} = \{p_{i,j}\}$ are extracted, we construct a graph $G = (V, E)$ to model the spatial and feature-based relationships among them. Here, $V = \{v_{i,j}\}$ represents the set of nodes corresponding to patches, and E denotes the set of edges connecting these nodes. Each node $v_{i,j} \in V$ is associated with a feature vector $x_{i,j} = p_{i,j} \in \mathbb{R}^{Cp^2}$, where each patch of size (p, p) is vectorized into a Cp^2 -dimensional feature vector. After extracting all patches, we organize them into a matrix

$$X_V = [x_1, x_2, \dots, x_{|\mathcal{V}|}]^T \in \mathbb{R}^{|\mathcal{V}| \times Cp^2},$$

where $|\mathcal{V}|$ is the number of patches (nodes) in the graph.

Next, we define the edges $e_{u,v} \in E$ based on k -connectivity and feature similarity. For each patch $p_i \in V$, we consider its neighboring patches, which are spatially adjacent to it within the feature map. A patch p_i is connected to its neighboring patches p_j , where $p_j \in \mathcal{N}(p_i)$ represents the set of neighbors of patch p_i . The neighborhood $\mathcal{N}(p_i)$ is determined by the spatial adjacency of patches, considering a fixed local window size k around each patch. The adjacency matrix $A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ is defined as:

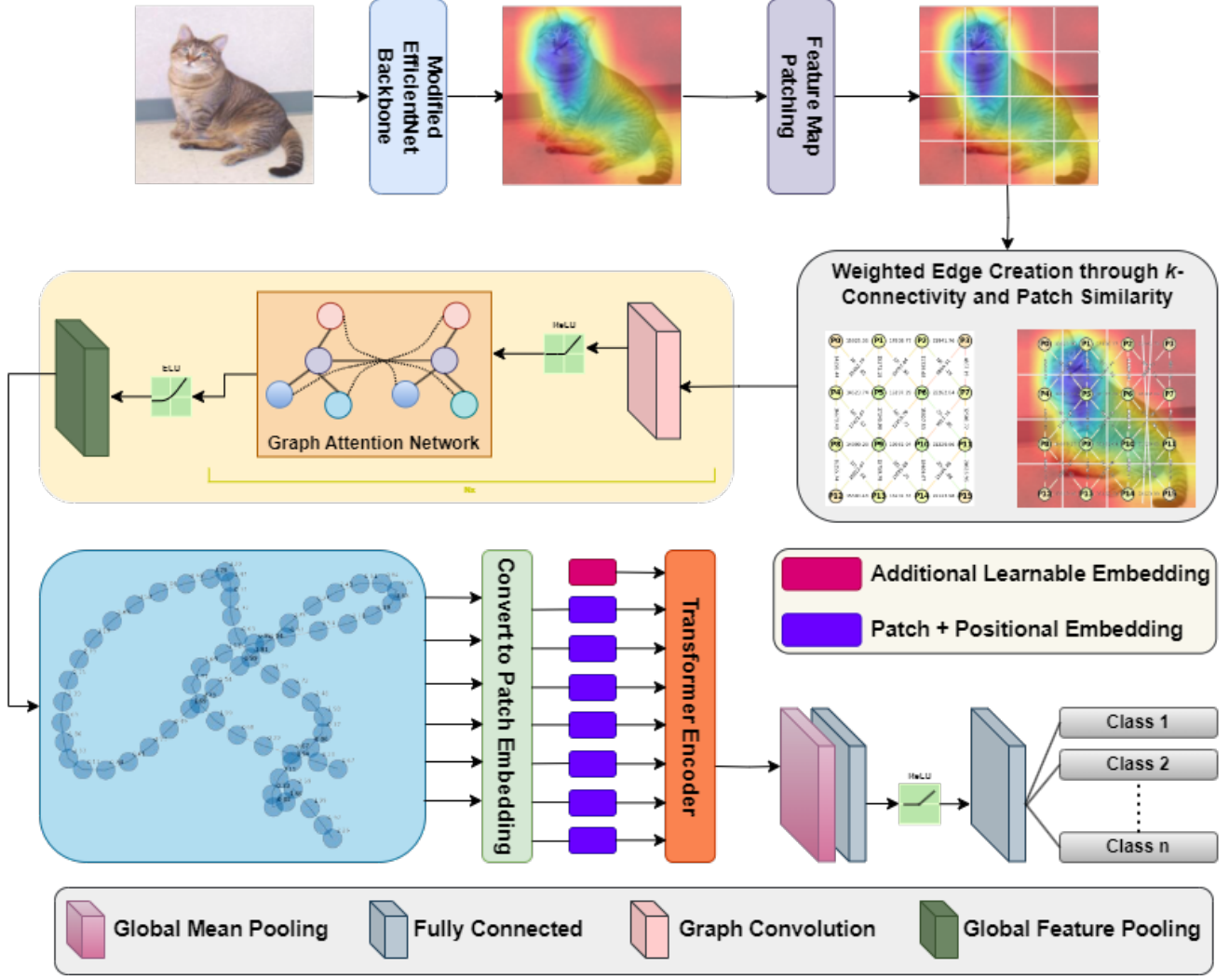


Figure 1. An illustration of our proposed SAG-ViT architecture for learning scale-aware, high-fidelity features with graph attention.

$$A_{u,v} = \begin{cases} \exp\left(-\frac{\|x_u - x_v\|_2^2}{\sigma^2}\right) & \text{if } v \in \mathcal{N}_k(u), \\ 0 & \text{otherwise,} \end{cases} \quad (5)$$

$$\mathcal{N}_k(u) = \left\{ v \in V \mid \begin{array}{l} \|\phi(u) - \phi(v)\|_2 \\ \text{is among the } k \text{ smallest distances} \end{array} \right\}. \quad (6)$$

where $\mathcal{N}_k(u)$ denotes the set of the k -nearest spatial neighbors of node u , and σ is a hyperparameter controlling the decay of the similarity function. To formalize the k -connectivity, we define the neighborhood function $\mathcal{N}_k(u)$ based on the Euclidean distance in the spatial grid, where $\phi(u)$ maps node u to its spatial coordinates (i, j) :

3.3. Integration of Graph Attention Networks (GAT) with Transformer Encoders

After constructing the graph $G = (V, E)$, we employ a Graph Attention Network (GAT) to process the node features and capture fine-grained dependencies among patches. Integrating GAT with transformer encoders facilitates the modeling of both local and global interactions, enhancing the discriminative power of the feature representations. The attention mechanism in GAT dynamically assigns weights

to neighboring nodes to emphasize more relevant connections.

For a given node u , the attention coefficient $\alpha_{u,v}$ with its neighbor v is computed as:

$$\alpha_{u,v} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}\mathbf{x}_u \parallel \mathbf{W}\mathbf{x}_v]))}{\sum_{k \in \mathcal{N}(u)} \exp(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}\mathbf{x}_u \parallel \mathbf{W}\mathbf{x}_k]))} \quad (7)$$

where $\mathbf{W} \in \mathbb{R}^{F' \times D}$ is a learnable linear transformation matrix, $\mathbf{a} \in \mathbb{R}^{2F'}$ is a learnable attention vector, and $\mathcal{N}(u)$ represents the set of neighbors of node u .

The updated feature \mathbf{x}'_u for node u is obtained by aggregating the transformed features of its neighbors weighted by the attention coefficients using a non-linear activation function (ELU):

$$\mathbf{x}'_u = \text{ELU} \left(\sum_{v \in \mathcal{N}(u)} \alpha_{u,v} \mathbf{W}\mathbf{x}_v \right) \quad (8)$$

To capture diverse relational patterns and stabilize the learning process, we employ multi-head attention. For h attention heads, the concatenated output for weight matrices $\mathbf{W}^{(i)}$ and attention coefficients $\alpha_{u,v}^{(i)}$ for the i -th head is given by:

$$\text{GAT}(\mathbf{x}) = \parallel_{i=1}^h \sigma \left(\sum_{v \in \mathcal{N}(u)} \alpha_{u,v}^{(i)} \mathbf{W}^{(i)} \mathbf{x}_v \right) \quad (9)$$

where σ denotes a non-linear activation function (e.g., ELU), and \parallel represents the concatenation operation across the attention heads.

These node embeddings $\mathbf{X}' = \{\mathbf{x}'_u\}_{u \in V}$ produced by the GAT are subsequently fed into a Transformer Encoder to model high-level interactions and long-range dependencies across all patches. Before integration, we apply positional encoding \mathbf{p}_u to each node embedding to retain spatial information:

$$\mathbf{X}'' = \mathbf{X}' + \mathbf{P} \quad (10)$$

where $\mathbf{P} = \{\mathbf{p}_u\}_{u \in V}$ represents the positional encodings.

The Transformer encoder processes the sequence of node embeddings using multi-head self-attention mechanisms. For a query vector $\mathbf{Q} = \mathbf{X}'' \mathbf{W}_Q^{(h)}$, a key vector $\mathbf{K} = \mathbf{X}'' \mathbf{W}_K^{(h)}$, and a value vector $\mathbf{V} = \mathbf{X}'' \mathbf{W}_V^{(h)}$, with $\mathbf{W}_Q^{(h)}, \mathbf{W}_K^{(h)}, \mathbf{W}_V^{(h)} \in \mathbb{R}^{d_{\text{model}} \times d_k}$ being learnable weight matrices for the h -th head, the self-attention operation for each head h is defined as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} \right) \mathbf{V} \quad (11)$$

The combination of GAT and Transformer encoders can be formalized as a two-stage feature transformation:

$$\text{Graph Attention Stage: } \mathbf{X}' = \text{GAT}(\mathbf{X}, \mathbf{A}) \quad (12)$$

$$\text{Transformer Encoding Stage: } \mathbf{X}''' = \text{Transformer}(\mathbf{X}' + \mathbf{P}) \quad (13)$$

We use this hierarchical processing to ensure that the model first refines patch embeddings through graph-based attention, captures localized relationships, and then leverages Transformer-based self-attention to integrate these refined embeddings into a cohesive global representation. After the Transformer encoder, we apply a global mean pooling operation to aggregate the sequence of embeddings into a single vector $\mathbf{z} \in \mathbb{R}^{d_{\text{model}}}$:

$$\mathbf{z} = \frac{1}{|V|} \sum_{u \in V} \mathbf{x}'''_u \quad (14)$$

Finally, we pass this pooled representation through a Multi-Layer Perceptron (MLP) to produce the final classification logits:

$$\hat{\mathbf{Y}} = \text{softmax}(\mathbf{W}_{\text{out}} \mathbf{z} + \mathbf{b}_{\text{out}}) \quad (15)$$

where $\mathbf{W}_{\text{out}} \in \mathbb{R}^{C \times d_{\text{model}}}$ and $\mathbf{b}_{\text{out}} \in \mathbb{R}^C$ are the weight matrix and bias vector of the output layer, respectively, and C is the number of target classes.

3.4. Ablation Study

To rigorously evaluate the contributions of each component in our proposed architecture, we conducted a comprehensive ablation study. This analysis aims to discern the individual impact of the EfficientNet backbone, the Graph Attention Network (GAT), and the Transformer encoder on the model's overall performance. By systematically removing or altering components, we can quantify their significance and validate the theoretical underpinnings of our design choices.

3.4.1 Experimental Setup

We designed three ablation experiments on the CIFAR-10 dataset to isolate the effects of each component:

1. **Backbone + GAT (No Transformer):** In this configuration, we exclude the Transformer encoder, allowing us to assess the role of the Transformer in capturing global dependencies. The model processes the feature embeddings extracted by the EfficientNet backbone through the GAT, generating class predictions directly from the aggregated node representations.
2. **Backbone + Transformer (No GAT):** Here, we omit the GAT to evaluate its contribution in modeling local dependencies and refining node features. The feature

embeddings from the backbone are fed into the Transformer encoder, which attempts to learn both local and global relationships without the explicit attention mechanism provided by the GAT.

3. **GAT + Transformer (No Backbone):** In this scenario, we remove the EfficientNet backbone to determine its impact on feature representation. Randomly initialized embeddings are used as input to the GAT and Transformer, highlighting the importance of high-quality feature extraction.

4. Results

In this section, we present a comprehensive evaluation of our proposed model across five diverse benchmark datasets: CIFAR-10, GTSRB, NCT-CRC-HE-100K, NWPU-RESISC45, and PlantVillage. These datasets encompass a wide range of domains, including natural images, traffic sign recognition, histopathological images, remote sensing data, and agricultural imagery. The diversity of these datasets allows us to thoroughly assess the effectiveness and generalization capability of our model across different types of image data.

4.1. Overall Performance

Our proposed model demonstrates superior performance compared to state-of-the-art architectures across all evaluated datasets. Table 1 summarizes the F1 scores achieved by our model and various baseline models utilizing different backbones.

Analyzing the results, our proposed model consistently outperforms the baseline models across all datasets. On CIFAR-10, our model achieves an F1 score of 0.9574, surpassing the next best model (ResNet-based within our architecture) by approximately 4.02%. This significant improvement underscores the effectiveness of integrating EfficientNet as the backbone in our model. EfficientNet’s compound scaling strategy optimizes network depth, width, and resolution, providing richer feature embeddings that enhance the model’s ability to capture intricate patterns in natural images when processed through our graph-based approach.

On the GTSRB dataset, which involves recognizing traffic signs under various challenging conditions, our model attains an F1 score of 0.9958. This is a notable improvement over the DenseNet201-based variant, which achieves 0.9862. The 0.96% increase, though seemingly modest due to the high baseline performance, demonstrates our model’s superior ability to capture subtle variations in traffic signs, crucial for real-world traffic sign recognition tasks.

For the NCT-CRC-HE-100K dataset, consisting of histopathological images for colorectal cancer classification, our model achieves an F1 score of 0.9861, outperforming the ResNet-based variant’s score of 0.9478 by approximately 3.83%. This substantial improvement indicates that

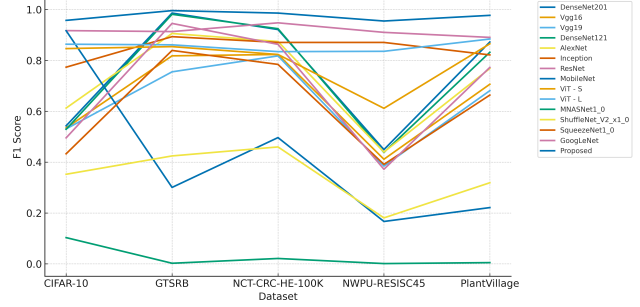


Figure 2. Scores of the proposed model compared to existing architectures across various datasets.

the EfficientNet backbone, combined with our graph-based processing, effectively captures complex tissue structures, enhancing the model’s discriminative power in medical image analysis.

On the NWPU-RESISC45 dataset, which includes remote sensing images from various land-use scenes, our model achieves an F1 score of 0.9549, outperforming the ResNet-based variant by 4.46%. This result demonstrates the model’s ability to capture spatial relationships and patterns inherent in remote sensing data more effectively than other backbones within our architecture.

Lastly, on the PlantVillage dataset, our model records an F1 score of 0.9772, significantly higher than the ResNet-based variant’s score of 0.8905, marking an improvement of approximately 8.66%. This considerable enhancement underscores the effectiveness of our model in agricultural imagery, particularly in detecting and classifying plant diseases where subtle visual cues are critical.

We also evaluated our model’s performance on the underwater trash dataset [21], benchmarking it against state-of-the-art algorithms (excluding backbone models) such as YOLOv8, RCNN, Fast-RCNN, and Mask-RCNN. Our model consistently performed better with a validation F1 of 0.96, exceeding the benchmark results of these models [20].

Comparing our model with standalone Vision Transformers (ViT-S and ViT-L), which do not incorporate our graph-based enhancements, we observe that while ViT models perform competitively on some datasets, they generally lag behind our proposed model. For instance, on CIFAR-10, ViT-L achieves an F1 score of 0.8637, which is 9.35% lower than our model’s performance. This comparison highlights the advantage of our approach in integrating EfficientNet for feature extraction with graph attention mechanisms and Transformer encoding, providing a more comprehensive understanding of the data. Figure 2 graphically compares the F1 Scores of all models across the five datasets, demonstrating the proposed model’s superior performance compared to existing architectures.

The consistent superiority of our proposed model across

Table 1. F1 Scores of Different Models on Benchmark Datasets

Backbone	CIFAR-10	GTSRB	NCT-CRC-HE-100K	NWPU-RESISC45	PlantVillage
DenseNet201	0.5427	0.9862	0.9214	0.4493	0.8725
Vgg16	0.5345	0.8180	0.8234	0.4114	0.7064
Vgg19	0.5307	0.7551	0.8178	0.3844	0.6811
DenseNet121	0.5290	0.9813	0.9247	0.4381	0.8321
AlexNet	0.6126	0.9059	0.8743	0.4397	0.7684
Inception	0.7734	0.8934	0.8707	0.8707	0.8216
ResNet	0.9172	0.9134	0.9478	0.9103	0.8905
MobileNet	0.9169	0.3006	0.4965	0.1667	0.2213
ViT - S	0.8465	0.8542	0.8234	0.6116	0.8654
ViT - L	0.8637	0.8613	0.8345	0.8358	0.8842
MNASNet1.0	0.1032	0.0024	0.0212	0.0011	0.0049
ShuffleNet_V2_x1.0	0.3523	0.4244	0.4598	0.1808	0.3190
SqueezeNet1.0	0.4328	0.8392	0.7843	0.3913	0.6638
GoogLeNet	0.4954	0.9455	0.8631	0.3720	0.7726
Proposed	0.9574	0.9958	0.9861	0.9549	0.9772

diverse datasets can be attributed to several key factors:

- **Efficient Feature Extraction:** The EfficientNet backbone within our architecture provides high-quality feature embeddings due to its balanced scaling of network depth, width, and resolution. This results in richer and more discriminative features compared to other CNN backbones.
- **Graph-Based Representation:** By constructing k-connectivity graphs from feature map patches, our model effectively models spatial and semantic relationships between image regions, capturing both local and global dependencies.
- **Attention Mechanisms:** The Graph Attention Network assigns adaptive weights to neighboring nodes, emphasizing relevant regions and enhancing local feature representation. The Transformer encoder further captures long-range dependencies and global context, which is particularly beneficial for complex images where global interactions are crucial for accurate classification.

4.2. Hardware Efficiency

We also evaluated the hardware efficiency of our proposed model in terms of RAM and GPU VRAM usage. Table 2 details the resource consumption of each model across the different datasets.

Our proposed model demonstrates competitive resource utilization, especially considering its superior performance. On CIFAR-10, our model uses 7.24% RAM, which is lower than several other variants using different backbones, such as the VGG16-based version that consumes 11.5% RAM. This indicates that incorporating EfficientNet in our archi-

tecture not only enhances performance but also improves hardware efficiency.

Regarding GPU VRAM usage, our model maintains moderate consumption. For example, on GTSRB, it uses 36.38% GPU VRAM, which is slightly higher than some CNN-based variants but significantly lower than the ViT-L model’s 81.87% VRAM usage on CIFAR-10. Despite the additional components of the GAT and Transformer encoder, the efficient feature extraction of EfficientNet and the sparsity of the k-connectivity graphs contribute to keeping resource usage reasonable within our architecture. Figure 3 illustrates the RAM and GPU VRAM usage of various models, highlighting the proposed model’s resource efficiency.

4.3. Ablation Study

To evaluate the contribution of each component in our proposed model, we conducted an ablation study on the CIFAR-10 dataset. The results are summarized in Table 3.

When the model includes the EfficientNet backbone with the GAT but without the Transformer encoder, the F1 score drops to 0.7785. This significant decrease underscores the crucial role of the Transformer encoder in capturing global dependencies and enhancing classification accuracy. The self-attention mechanism in the Transformer allows the model to weigh the importance of all patches relative to each other, facilitating a holistic understanding of the image.

Conversely, using the EfficientNet backbone with the Transformer encoder but without the GAT results in an F1 score of 0.7593. This emphasizes the importance of the

Table 2. Hardware RAM and GPU (VRAM) Consumption (%) of Different Models on Benchmark Datasets. The minimum values in each column are highlighted in blue and the second minimum values in green.

Backbone	CIFAR		GTSRB		NCT-CRC-HE-100K		PlantVillage		NWPU-RESISC45	
	RAM	GPU	RAM	GPU	RAM	GPU	RAM	GPU	RAM	GPU
DenseNet201	9.7	24.32	11	19.15	11.1	19.14	11.3	17.97	11	18.77
Vgg16	11.5	42.19	11.3	29.76	12.1	29.77	11.6	30.58	11.1	29.27
Vgg19	11.8	55.93	11.1	29.27	11.8	29.27	11.4	30.24	11.2	29.77
DenseNet121	8.1	28.8	10.9	16.81	11.2	16.81	11.2	17.52	10.8	16.88
AlexNet	13.1	12.71	11.4	16.34	12.2	16.34	13.4	17.46	12	16.34
Inception	10.2	29.4	10.5	22.1	11.4	24.7	11.5	24.2	10.8	20.32
ResNet	9.5	23.1	10.7	18.9	11.1	20.3	11.2	21.2	10.9	18.92
MobileNet	12.41	22.5	11	21.63	7.6	14.08	11.3	19.39	10.9	21.63
MNASNet1.0	7.8	18.07	10.9	15.16	7.5	10.68	11.2	16.13	10.9	15.16
ShuffleNet_V2_x1.0	9.8	12.13	10.8	15.14	8.7	11.5	11.3	16.06	10.9	15.1
SqueezeNet1.0	11.72	19.37	7.5	10.34	7.7	10.62	11.4	16.02	11	15.21
GoogLeNet	8.1	19.06	10.7	15.07	9.4	16.12	11.3	16.05	10.8	15.08
ViT - S	11.37	30.62	12.3	29.7	12.1	24.28	12.5	27.32	13.2	29.54
ViT - L	15.17	81.87	13.6	33.72	14.8	33.21	13.5	35.98	13.9	34.96
Proposed	7.24	33.12	9.2	36.38	7.6	37.32	8.2	39.32	10.72	11.62

Table 3. Ablation Study Results on CIFAR-10 Dataset

Model	F1	RAM (GB)	GPU (VRAM) (GB)	Time per Epoch
Backbone + GAT (no transformer)	0.7785	5.6	4.9	14 mins 30 sec
Backbone + transformer (no GAT)	0.7593	3.1	4.5	16 mins 7 sec
GAT + Transformer (no Backbone)	0.5032	4.3	5.3	1 hour 33 mins

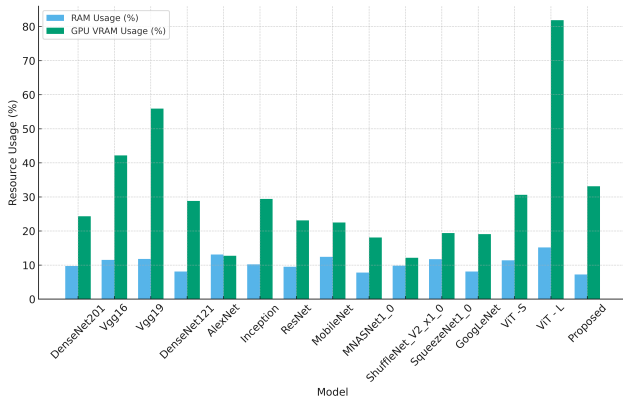


Figure 3. Comparison of RAM and GPU VRAM usage (%) across models

GAT in refining local feature representations before global processing. The GAT enhances node features by aggregating

information from immediate neighbors, effectively capturing local structural information essential for accurate classification.

When the model comprises the GAT and Transformer encoder without the EfficientNet backbone, the F1 score drops drastically to 0.5032. This significant decline highlights the importance of the EfficientNet backbone in providing rich and discriminative feature embeddings necessary for effective graph construction and subsequent processing. The RAM and GPU VRAM usage of various models are illustrated in Figure 4.

These observations confirm that each component of our proposed architecture is essential and contributes uniquely to the model’s overall performance. The EfficientNet backbone generates high-quality feature embeddings; the GAT captures local dependencies through attention mechanisms; and the Transformer encoder models global relationships, enabling the model to understand complex patterns that span different regions of the image.

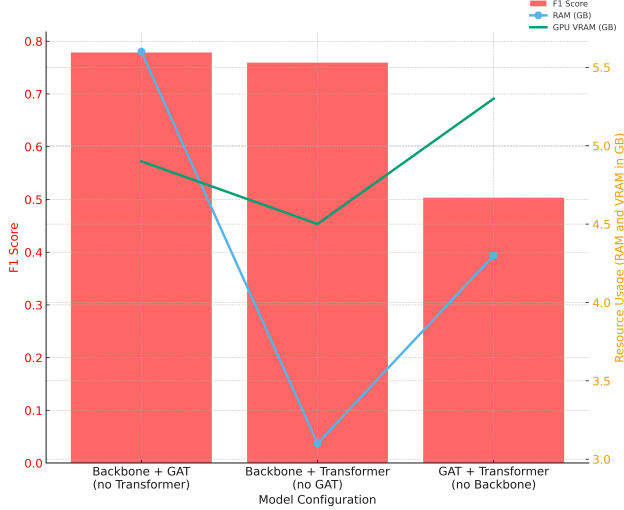


Figure 4. Ablation study of model configurations, showing F1 Score, RAM, and GPU VRAM usage (GB) for different combinations of backbone, GAT, and Transformer components.

4.4. Discussion

The results validate our hypothesis that integrating efficiently scaled feature embeddings with graph-based attention mechanisms and Transformer encoders significantly enhances model performance across diverse datasets. The EfficientNet backbone within our architecture provides superior feature representations, which, when used in our graph construction, enhance the model’s ability to capture both local and global dependencies effectively.

Our comparisons are designed to showcase the critical role that EfficientNet plays within our proposed model. The substantial improvements over models with other backbones highlight that the enhancements result from the synergistic integration of EfficientNet with our graph-based approach, rather than from the backbone alone. This integration allows for richer feature representations that, when processed through the GAT and Transformer encoder, lead to improved classification accuracy.

The ablation study confirms that the removal of any component leads to a significant drop in performance, establishing that the improvements are due to the cohesive integration of these elements within our architecture. By quantitatively demonstrating the impact of each component, we validate the architectural choices grounded in principles of deep learning, graph theory, and attention mechanisms.

Moreover, the inclusion of diverse datasets such as GTSRB, NCT-CRC-HE-100K, NWPU-RESISC45, and PlantVillage demonstrates the robustness and generalization ability of our model across different domains. These datasets present various challenges, including fine-grained classification, medical image analysis, remote sensing, and

agricultural disease detection. Our model’s consistent superiority across these datasets emphasizes its versatility and effectiveness in handling complex and varied image data.

In conclusion, our proposed model, integrating EfficientNet as the backbone within a graph-based framework enhanced by attention mechanisms, significantly outperforms existing models. The improvements stem from the unique combination of high-quality feature extraction, graph-based representation of spatial relationships, attention mechanisms for local dependencies, and Transformer encoding for global context. This holistic approach ensures that our model effectively captures and utilizes the rich semantic information necessary for accurate image classification across diverse datasets.

5. Conclusion

This paper presents the Scale-Aware Graph Attention Vision Transformer (SAG-ViT), a novel framework designed to address the challenge of multi-scale feature representation in Vision Transformers. By utilizing EfficientNet for feature extraction and organizing image patches into a graph, SAG-ViT effectively captures both local and global relationships in images. The incorporation of a Graph Attention Network (GAT) refines the node embeddings, while the Transformer encoder captures long-range dependencies and complex interactions. Experimental evaluations on benchmark datasets, including CIFAR-10, GTSRB, NCT-CRC-HE-100K, NWPU-RESISC45, and PlantVillage, demonstrate the model’s effectiveness, showing significant improvements in image classification performance. Additionally, an ablation study provides insights into the importance of each component in the SAG-ViT framework, helping to understand their individual contributions to the overall performance. This work highlights the potential of integrating multi-scale features and graph-based attention mechanisms to enhance the capabilities of Transformer-based models in computer vision.

References

- [1] Chun-Fu Chen, Quan Fan, and Rajeev Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. 2021. 1, 2
- [2] Yinpeng et al. Chen. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3434–3443. IEEE, 2019. 2
- [3] Yinpeng et al. Chen. Mobile-former: Bridging mobilenet and transformer, 2021. 3
- [4] Alexey et al. Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. 2020. 1
- [5] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. Large-scale learnable graph convolutional networks. 2018. 2
- [6] Benjamin et al. Graham. Levit: a vision transformer in convnet’s clothing for faster inference, 2021. 2

- [7] Jiang et al. Guo. Cmt: Convolutional neural networks meet vision transformers. 2021. 2
- [8] Andrew et al. Jaegle. Perceiver: General perception with iterative attention. 2021. 2
- [9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. Available: <http://code.google.com/p/cuda-convnet/>. 1, 2
- [10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2016. 2
- [11] Ze et al. Liu. Swin transformer: Hierarchical vision transformer using shifted windows. 2021. 2
- [12] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. 2017. 1
- [13] Sachin Mehta and Mohammad Rastegari. Mobilevit: Lightweight, general-purpose, and mobile-friendly vision transformer, 2021. 2
- [14] Mark et al. Sandler. Mobilenetv2: Inverted residuals and linear bottlenecks, 2018. 3
- [15] Aravind et al. Srinivas. Bottleneck transformers for visual recognition, 2021. 2
- [16] Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training. 2021. 1
- [17] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. 2020. 1
- [18] Ashish et al. Vaswani. Attention is all you need. 2017. 1
- [19] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. 2017. 1, 2
- [20] Jaskaran Singh Walia and Pavithra L K. Deep learning innovations for underwater waste detection: An in-depth analysis, 2024. 6
- [21] Jaskaran Singh Walia and Karthik Seemakurthy. Optimized custom dataset for efficient detection of underwater trash. In *Towards Autonomous Robotic Systems*, pages 292–303, Cham, 2023. Springer Nature Switzerland. 6
- [22] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803. IEEE, 2018. 2
- [23] Haiping et al. Wu. Cvt: Introducing convolutions to vision transformers. 2021. 2
- [24] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2019. 1, 2
- [25] Li et al. Yuan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. 2021. 2
- [26] Yixin et al. Zhu. A survey on graph structure learning: Progress and opportunities. 2021. 2