Computers in Biology and Medicine 189 (2025) 109902

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

FEGGNN: Feature-Enhanced Gated Graph Neural Network for robust few-shot skin disease classification

Abdulrahman Noman, Zou Beiji, Chengzhang Zhu *, Mohammed Alhabib, Raeed Al-sabri

School of Computer Science and Engineering, Central South University, Changsha, 410083, China

ARTICLE INFO

Keywords:
Few-shot learning
Skin disease classification
Graph neural network
Gated recurrent units
Enhanced features

ABSTRACT

Accurate and timely classification of skin diseases is essential for effective dermatological diagnosis. However, the limited availability of annotated images, particularly for rare or novel conditions, poses a significant challenge. Although few-shot learning (FSL) methods in computer-aided diagnosis (CAD) can decrease the dependence on extensive labeled data, their efficacy is often diminished by these challenges, particularly the catastrophic forgetting defect during the sequence of few-shot tasks. To address these challenges, we propose a Feature Enhanced Gated Graph Neural Network (FEGGNN) framework to improve the few-shot classification of skin diseases. The FEGGNN leverages an efficient Asymmetric Convolutional Network (ACNet) to extract high-quality feature maps from skin lesion images, which are subsequently used to construct a graph where nodes represent feature vectors and edges indicate similarities between samples. The core of FEGGNN consists of multiple aggregation blocks within the Graph Neural Network (GNN) framework, which iteratively refine node and edge features. Each block updates node features by aggregating information from neighboring nodes, weighted by edge features, to capture contextual relationships. Simultaneously, Gated Recurrent Units (GRUs) model long-term dependencies across tasks, enabling effective knowledge transfer and mitigating catastrophic forgetting. The Efficient Channel Attention (ECA) mechanism further enhances edge feature updates by focusing on the most relevant feature channels, optimizing edge weight computation. This iterative refinement process enables FEGGNN to progressively enhance feature representations, ensuring robust performance in diverse few-shot classification tasks. FEGGNN's superior ability to generalize to unseen classes is demonstrated by its state-of-the-art performance, achieving 84.90% accuracy on Derm7pt and 95.19% on SD-198 in 2-way 5-shot settings.

1. Introduction

Skin diseases represent a significant global health burden, profoundly affecting the quality of life of millions worldwide [1]. The diverse range of dermatological conditions necessitates early and precise diagnosis to prevent disease progression and ensure optimal treatment outcomes. With the prevalence of skin conditions surpassing that of hypertension, obesity, and cancer combined, there is an urgent need for efficient and reliable diagnostic tools in dermatology. Recent advancements in computer-aided diagnosis (CAD) utilizing deep learning [2], particularly Convolutional Neural Networks (CNNs), have demonstrated notable success in automating medical image analysis, yielding impressive results in applications such as organ segmentation [3], tumor detection [4], and disease screening [5]. Despite these advancements, significant challenges remain, particularly concerning the classification of rare or novel skin diseases, largely due to the limited availability of annotated data for such conditions.

Conventional deep learning models, including CNNs, depend heavily on large-scale labeled datasets to achieve optimal performance, a requirement that poses considerable challenges for dermatological classification. In skin disease, obtaining annotated data is especially challenging due to patient privacy constraints, substantial labeling costs, and the inherently low prevalence of rare conditions. Consequently, this data scarcity leads to suboptimal classification performance, particularly in less prevalent conditions. Moreover, the complexity of skin disease manifestations, characterized by variations in texture, color, and appearance, further complicates accurate diagnosis. Few-Shot Learning (FSL) [6,7] has emerged as a promising approach to mitigate these challenges by enabling models to adapt to new classes with limited labeled examples. However, existing FSL approaches often face limitations, such as catastrophic forgetting, wherein models lose previously acquired knowledge when adapting to new tasks, thereby hindering effective generalization across sequential learning scenarios.

* Corresponding author.
E-mail address: anandawork@126.com (C. Zhu).

https://doi.org/10.1016/j.compbiomed.2025.109902
Received 9 October 2024; Received in revised form 25 January 2025; Accepted 19 February 2025
Available online 9 March 2025

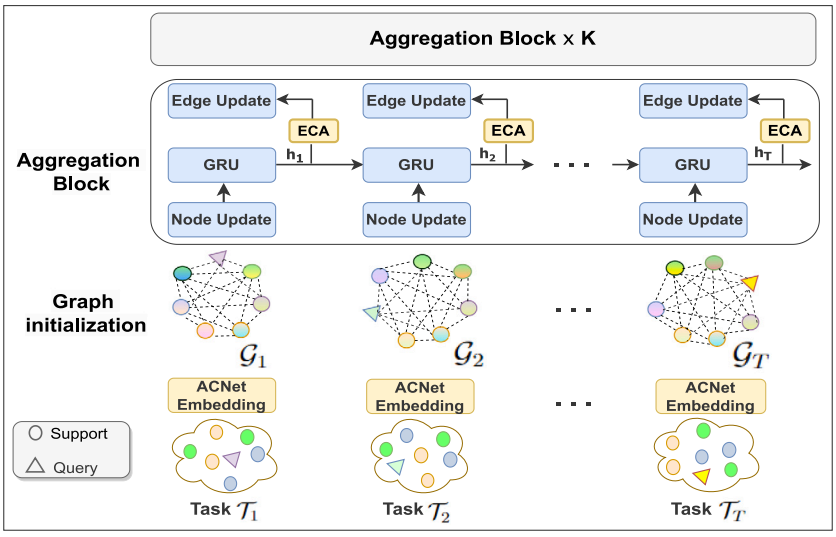

Fig. 1. Flowchart of the proposed FEGGNN model, demonstrating the process of graph feature updating and knowledge transfer across tasks. Each aggregation block consists of node updating followed by edge updating guided by an attention mechanism, while a GRU extracts and transfers task-relevant knowledge throughout the sequence.

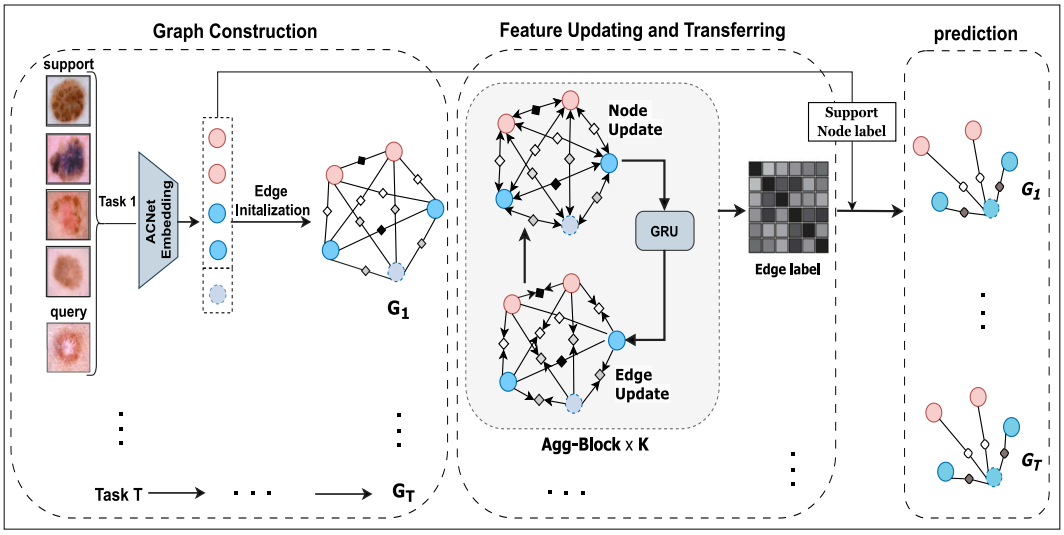

Fig. 2. Overview of the FEGGNN framework for a 2-way 2-shot classification task. The Asymmetric Convolutional Network (ACNet) converts input images into feature nodes, with each node representing a sample from two distinct classes, depicted in orange and blue. Solid-lined nodes denote the support set, while dotted-lined nodes indicate the query set. Edge feature strength is visually represented by color intensity, where white corresponds to a value of 0 and black to 1. For efficient learning, the training set D_{train} and testing set D_{test} are divided into multiple task sequences $\{G_t\}_{t=1}^{T}$, where T denotes the sequence length. Further methodological details are provided in Section 3.

Graph Neural Networks (GNNs) [8,9] have shown considerable potential in modeling complex relationships between data points, particularly in few-shot learning contexts for general computer vision. However, existing GNN-based methods for skin disease classification still face notable challenges in extracting detailed feature representations, as they frequently rely on conventional CNN architectures that may inadequately capture the complex and nuanced features necessary for effective dermatological image analysis.

To address the aforementioned challenges in few-shot skin disease classification, we propose the Feature-Enhanced Gated Graph Neural Network (FEGGNN), a novel framework specifically designed to enhance the accuracy and robustness of classification in data-limited scenarios. FEGGNN achieves this by integrating advanced feature refinement, cyclic feature updating, and efficient knowledge transfer mechanisms between tasks, as illustrated in Figs. 1 and 2. The framework begins with Asymmetric Convolutional Networks (ACNets) to extract highly discriminative features, effectively capturing the complex dermatological characteristics necessary for distinguishing various skin conditions. These extracted features are subsequently fed into a Graph Neural Network (GNN), wherein nodes represent individual sample features and edges denote similarity measures, enabling the model to comprehensively leverage inter-sample relationships. To tackle the issue of catastrophic forgetting and facilitate efficient knowledge transfer across sequential tasks, Gated Recurrent Units (GRUs) are employed within multiple aggregation blocks to cyclically update both node and edge features. Furthermore, an Efficient Channel Attention (ECA-Net) mechanism is incorporated to refine feature representations by emphasizing the most informative channels, thereby further enhancing the model's overall classification accuracy and robustness. In summary, the primary contributions of our model can be outlined as follows:

- FEGGNN presents a novel few-shot learning model that iteratively updates node and edge features, enhancing the accuracy and robustness of rare skin disease classification, thus improving computer-aided diagnosis in scenarios with limited labeled data.

- Our framework enhances skin disease classification by integrating ACNet with a CNN for discriminative feature extraction, and utilizing ECA-Net attention mechanisms within graph computations to refine feature similarity, resulting in improved classification accuracy and robustness.
- FEGGNN employs a Gated Recurrent Network (GRN) module to sequence few-shot tasks, enhancing knowledge transfer while mitigating catastrophic forgetting, thereby yielding a robust classifier with strong generalization to novel tasks.
- The proposed FEGGNN model, evaluated on benchmark datasets Derm7pt and SD-198, achieves state-of-the-art accuracies of 84.90% and 95.19%, respectively, outperforming existing methods and advancing automated skin disease diagnosis in limited data scenarios.

The remainder of this paper is organized as follows: Section 2 reviews the related work. Section 3 provides an in-depth description of the proposed FEGGNN framework. Section 4 outlines the experimental setup, including datasets and evaluation metrics, followed by a detailed discussion of the results and ablation studies. Finally, Section 5 presents the conclusion of our work and potential future directions.

2. Related works

2.1. Few-shot learning

Few-shot learning (FSL) leverages prior knowledge from previously learned tasks to improve model performance in recognizing new classes with limited labeled samples, enabling faster adaptation and stronger generalization to unseen categories. In the field of computer vision, FSL is divided into two primary groups: meta-learning-based methods and transfer-learning-based methods. Meta-learning approaches, which include optimization-based, generation-based, and metric-based methods, focus on extracting knowledge from batches of tasks and transferring it effectively to new tasks. Optimization-based methods [10,11] develop effective optimizers or well-initialized models that generalize quickly to novel classes, while generation-based methods [12,13] augment few-shot data by generating additional samples to mitigate overfitting. Metric-based methods [14–17] emphasize learning effective embeddings and suitable distance metrics to evaluate similarities between samples, enhancing classification performance in few-shot scenarios. Conversely, transfer learning [18–20] addresses the challenge of limited data by pre-training models on base classes with abundant training data and subsequently fine-tuning them on few-shot unseen classes, allowing the model to leverage learned representations for improved performance. To address catastrophic forgetting, Dynamic-FSL [21] facilitates rapid recognition and dynamic integration of novel categories. CML-BGNN [8] presents a Continual Meta-Learning framework using Graph Neural Networks, enhancing few-shot learning with memory-augmented knowledge transfer and addressing uncertainty through Bayesian edge inference. DenseNet [22] proposed a continuously evolving network using neural implants for new tasks, allowing integrated features from samples to be retained. bi-GEGRN [9] uses bidirectional-gated graph neural networks with edge-labeling to improve feature updates and information transfer in few-shot learning. While these methods significantly advance few-shot learning, they often encounter challenges such as catastrophic forgetting and fail to effectively capture the relational dependencies between samples. In contrast, our model FEGGNN overcomes these limitations by modeling FSL tasks as a sequence of tasks, thereby enhancing learning and facilitating knowledge transfer from training to testing phases. Specifically, FEGGNN converts FSL tasks into task graphs, as illustrated in Fig. 1, enabling the extraction and transfer of latent knowledge through aggregation blocks. This approach not only improves task-specific performance but also enhances the model's ability to generalize to new, unseen tasks.

2.2. Few-shot skin disease classification

In recent years, researchers have sought to integrate few-shot learning with skin disease classification to address the challenges posed by imbalanced categories and limited images for rare conditions. A prevalent approach within few-shot learning for classifying rare skin diseases is meta-learning, which trains models to acquire the ability to learn effectively from minimal data. In this domain, several methods have been developed to enhance performance based on the MAML framework [23]. DAML [24] enhances adaptation to novel classes by adjusting task weights, emphasizing challenging tasks while downweighting easier ones. MetaMed [25] employs sophisticated data augmentation techniques to improve the model's generalizability. Other approaches extend prototypical networks [15] to improve performance and adaptability, such as CGProNet [26] extends the Prototypical Network by leveraging graph neural networks and external memory to expand the feature space, thereby improving the performance of skin disease classification. QR loss [27], which addresses the limitations of cross-entropy loss in episodic training by introducing query relative loss, enabling better use of cross-sample information. PCN [28] refines Prototypical Networks by initializing disease classes using multiple prototypes, with each class consisting of a fixed number of sub-clusters. Recent advancements in few-shot learning for skin disease classification, exemplified by Meta-DermDiagnosis (MDD) [6] and SCAN [7], have made significant strides in addressing challenges associated with skin image classification by leveraging meta-learning methodologies and intra-class clustering techniques. Meta-DermDiagnosis employs meta-learning frameworks such as Reptile and Prototypical Networks to enhance diagnostic capabilities. However, it primarily focuses on learning transferable representations, often neglecting the structural relationships between samples. This limitation impedes its ability to capture the nuanced intra-class variability crucial for accurate classification. Similarly, SCAN utilizes a dual-branch framework to model intra-class sub-clusters through cluster functions, memory-based techniques, and purity loss functions. While effective in capturing some aspects of the class structure, it overlooks the importance of focusing on the specific relevant features of skin lesions and fails to model the relationships between query and support samples effectively. In contrast, our model FEGGNN enhances feature extraction by integrating the ACNet block into standard CNN architectures, enabling the extraction of rich and discriminative features. These feature vectors are then represented as graphs, allowing the model to capture and refine relationships between nodes effectively. By modeling interdependencies between features, FEGGNN leverages the Efficient Channel Attention (ECA) mechanism to refine feature similarity, focusing on the most relevant channels for classification. This combined approach not only improves feature representation but also facilitates efficient knowledge transfer across tasks through a graph-based framework. Consequently, FEGGNN excels at generalizing from well-established data to rare, unseen classes, making it particularly effective in few-shot skin disease classification, where labeled data is scarce.

2.3. Graph neural networks

Graph Neural Networks (GNNs) were initially developed to optimize the processing of graph-structured data, enabling neural networks to efficiently aggregate information from neighboring nodes. In recent years, GNNs have been integrated into few-shot learning methodologies to improve generalization and adaptability with limited data. The pioneering work of GNN [29] introduced the first application of GNNs in few-shot learning, leveraging graph structures to enhance task-specific performance. Building on this foundation, TPN [30] introduced a transductive approach that utilized a top-k graph to facilitate the transfer of labels between the support and query sets, thus improving model performance in few-shot scenarios. The EGNN [31] further advanced this concept by capitalizing on the intrinsic structure of directed graphs and

defining edge labels instead of node labels, thus enabling the model to leverage edge information. This approach allows for explicit modeling of intra-cluster similarity and inter-cluster dissimilarity, enhancing the clustering and generalization abilities of the model in few-shot learning tasks. In a meta-learning framework, methods like CML-BGNN [8] and bi-GEGRN [9] integrate edge-labeling graphs with graph recurrent network techniques to address challenges such as catastrophic forgetting, which is a common problem in sequential learning tasks. FEGGNN builds upon these foundational GNN-based methods but introduces several key innovations. Unlike EGNN, which relies on edge-labeling to model relationships, FEGGNN incorporates ACNet (Asymmetric Convolutional Networks) to enhance feature extraction, transforming sample features into graph structures for relational modeling. This approach allows FEGGNN to capture richer feature representations, which are then refined using the Efficient Channel Attention (ECA) mechanism. By selectively focusing on the most relevant features, FEGGNN improves classification performance, ensuring that key information is prioritized during the learning process. Additionally, FEGGNN addresses catastrophic forgetting, common in models like CML-BGNN and bi-GEGRN, by integrating Gated Recurrent Units (GRUs) in its aggregation blocks, enabling efficient knowledge transfer and better generalization to unseen classes. By combining ACNet for feature extraction, graph-based relational modeling, and attention-based refinement, FEGGNN offers a more robust solution for few-shot learning, especially in medical image analysis, where generalization from limited data is crucial, such as in skin disease classification.

2.4. ACNet-based feature enhancement

Feature enhancement is an essential component of improving few-shot learning (FSL) models. In this work, we leverage the Asymmetric Convolution Network (ACN) [32] and Efficient Channel Attention (ECA) [33] as primary techniques to enhance feature extraction and representation while preserving model efficiency. The progress of ACNet can be categorized into three primary directions: *Approximating Square-Kernel Convolutional Layers for Compression and Acceleration:* Asymmetric convolutions have been widely used to approximate standard square-kernel convolutional layers, enabling compression and acceleration. Early studies [34,35] demonstrated that a standard $d \times d$ convolutional layer could be decomposed into a sequence of $d \times 1$ and $1 \times d$ kernels, significantly reducing parameters and computational demands. However, directly applying these decompositions often leads to substantial information loss due to the higher intrinsic rank of learned kernels in practice [36]. *Optimizing Computational Efficiency:* Asymmetric convolutions have also been employed as architectural design elements aimed at optimizing computational efficiency. For example, Inception-v3 [37] replaced 7×7 convolutions with a sequence of 1×7 and 7×1 convolutions, achieving efficiency gains. However, this replacement proved suboptimal in low-level layers, where the loss of spatial information became more pronounced. Similarly, ENet [38] adopted the decomposition of 5×5 convolutions to expand the receptive field under constrained computational budgets, and EDANet [39] reduced the number of parameters and computations by 33% through decomposing 3×3 convolutions, although with minor performance degradation. *Modifying CNN architectures:* While many studies have sought to improve model performance by altering the structure of convolutional neural networks (CNNs), these methods often come with the cost of increased computational demands and resource consumption. For instance, the use of auxiliary classifiers [40] introduces deeper and wider networks, resulting in performance gains but requiring extensive hyperparameter tuning. Similarly, the design of convolutional kernels [41] has demonstrated improvements in feature representation and classification accuracy, but these improvements come with higher computational demands and increased parameter complexity. In contrast, our approach utilizes 1D asymmetric convolutions to enhance off-the-shelf models in an architecture-neutral manner, enriching the feature space during training without the need for specialized hyperparameter tuning or introducing additional computational overhead during inference. Within the FEGGNN framework, ACNet plays a pivotal role in feature extraction. It enhances the expressiveness of CNN layers, enabling the model to capture richer features while maintaining computational efficiency. By integrating ACNet with graph-based relational modeling and the ECA mechanism, FEGGNN achieves highly discriminative feature representations and effective generalization from limited labeled data. This makes FEGGNN particularly well-suited for complex tasks, such as medical image analysis, where both accuracy and computational efficiency are essential.

3. Methodology

3.1. Problem definition

Few-shot skin disease classification presents a critical challenge in machine learning, requiring the development of effective classifiers with a limited number of labeled samples. This challenge is particularly pronounced in medical contexts, where obtaining abundant annotated data for each possible skin disease is often impractical. The objective is to train classifiers that can generalize well to novel, unseen diseases. To address this, we distinguish between two datasets: the **training dataset** (D_{train}), which forms the foundation for model learning, and the **testing dataset** (D_{test}), which includes novel classes and is used to evaluate the model's generalization ability. These datasets are mutually exclusive, i.e., $D_{train} \cap D_{test} = \Phi$. Our approach utilizes an N-way, K-shot classification framework, as shown in Fig. 2, where N represents the number of classes per task, and K represents the number of samples per class. Each task $\mathcal{T} = (S, Q)$ consists of a **Support Set** (S), represented as $S = \{(x_i, y_i)\}_{i=1}^{N \times K}$, which contains the primary training examples, and a **Query Set** (Q), which contain N_q samples of unknown classes, serving as a validation subset, described by $Q = \{(x_i, y_i)\}_{i=N \times K+1}^{N \times K+N_q}$. The primary challenge lies in effectively leveraging node and edge distinctive features within this framework through a sequence of tasks T without forgetting to achieve accurate and reliable predictions.

3.2. Feature extraction

In the proposed FEGGNN framework for skin disease classification, Asymmetric Convolutional Networks (ACNets) are employed feature extraction for embedding, specifically designed to achieve efficient and effective feature representation. ACNets utilize a combination of asymmetric convolutional kernels, including 3×3, 3×1, and 1×3, to extract a diverse array of features from skin lesion images. This allows the network to capture both fine-grained details and broader structural patterns, which are essential for differentiating a variety of skin conditions. The feature extraction process for an input image x_i is represented as:

$$\mathbf{v}_i^0 = f_{\text{emb}}(x_i; \Theta_{\text{emb}}), \tag{1}$$

where $\mathbf{v}_i^0$ denotes the initial feature vector extracted from the image x_i, f_{emb} represents the feature extraction function implemented by the ACNet, and Θ_{emb} refers to the set of learnable parameters within the ACNet. During training, ACNet employs a parallel kernel configuration (3×3, 3×1, and 1×3) to enrich the feature space by capturing diverse spatial relationships at multiple scales. This design enhances the model's ability to extract intricate patterns and structural details without introducing additional hyperparameters beyond those of standard convolutional layers. Unlike methods such as auxiliary classifiers or kernel-specific designs [40,41], which modify the CNN architecture and require extensive hyperparameter optimization, ACNet achieves feature diversity in an *architecture-neutral manner*, seamlessly enhancing existing convolutional layers without adding complexity to model calibration. While the use of multiple kernels increases computational operations during training, this marginal overhead is offset

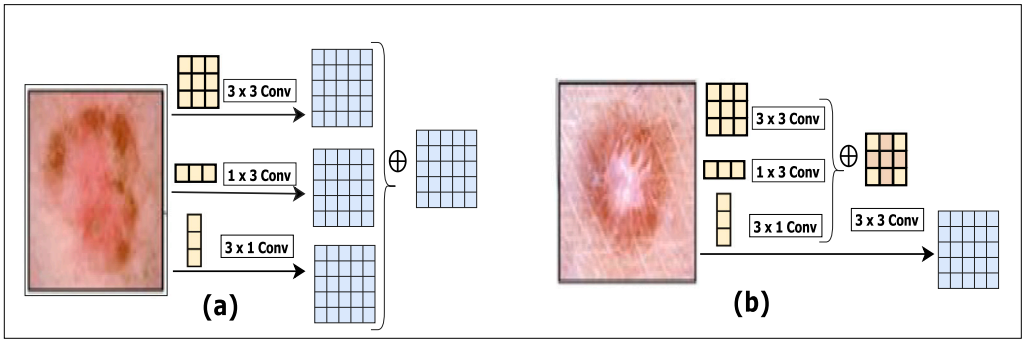

Fig. 3. Overview of ACNet: (a) Training stage, where each standard 3 × 3 convolution layer is replaced by an Asymmetric Convolution Block (ACB) consisting of three parallel layers with 3 × 3, 1 × 3, and 3 × 1 kernels, whose outputs are summed for enhanced feature extraction. (b) Testing stage, where the ACB is fused back into a standard 3 × 3 kernel, maintaining the original architecture with no additional computations required during inference.

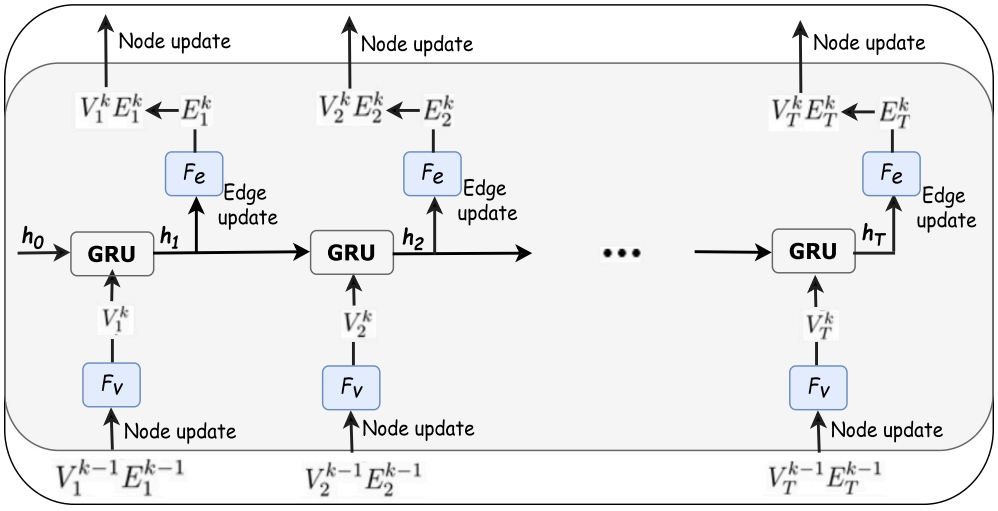

Fig. 4. The Aggregation Block processes a graph G_t by taking the input node set V_t^{k-1} and edge set E_t^{k-1}. It first updates the node features using the function F_v, followed by knowledge transfer via a GRU. The edge features are then updated using the function F_e. The block outputs the updated node set V_t^k and edge set E_t^k, where $t = 1, 2, \ldots, T$.

by significant improvements in feature richness and representation. Crucially, during inference, the **kernel fusion process** combines the outputs of the asymmetric kernels into a standard 3 × 3 kernel. This ensures that no additional computations or parameters are introduced during testing, preserving the efficiency of the original architecture while retaining the benefits of the enriched features learned during training. This streamlined design makes ACNet particularly suitable for *real-time clinical applications*, where computational efficiency is critical and resources are often limited. Fig. 3 presents an overview of ACNets, detailing (a) the training stage structure and (b) the testing stage configuration.

3.3. Graph construction

As illustrated in Fig. 2, the graph construction process converts few-shot learning tasks ($TFSL$) into a structured graph representation, suitable for classification. Each task $\mathcal{T} \in T$, comprising a support set S and a query set Q, is represented as a fully connected task graph within this framework. The graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}; \mathcal{T})$, where $\mathcal{V} = \{V_i\}_{i=1,\ldots,|\mathcal{T}|}$ represents the set of nodes, and $\mathcal{E} = \{E_{ij}\}_{i,j=1,\ldots,|\mathcal{T}|}$ denotes the set of edges. Let v_i and e_{ij} be the node feature of V_i and the edge feature of E_{ij}, respectively. $|\mathcal{T}| = N \times (K + N_q)$ represents the total number of samples in the task $\mathcal{T}$. From Eq. (1), the initial node features $\mathbf{v}_i^0 = f_{\text{emb}}(x_i; \Theta_{\text{emb}})$ are vectors extracted from samples using (ACNet). These vectors encapsulate the essential characteristics of the corresponding skin lesion images, thereby forming the fundamental units of the graph.

The edges, denoted by e_{ij}, capture the relationships between the nodes, reflecting the pairwise similarities between them. The edge feature $\mathbf{e}_{ij}^0$ is initialized in accordance with the edge-labeling graph framework:

$$\mathbf{e}_{ij}^0 = \begin{cases} 1, & \text{if } y_i = y_j \text{ and } x_i, x_j \in S, \\ 0, & \text{if } y_i \neq y_j \text{ and } x_i, x_j \in S, \\ 0.5, & \text{otherwise.} \end{cases} \tag{2}$$

This initialization scheme ensures that the graph accurately reflects the relationships within the dataset. Edge features are set to 1 for samples in the same class within the support set, indicating a strong connection, and to 0 for samples in different classes, indicating no connection. In all other cases, edge features are initialized to 0.5, representing a moderate connection. The initial node features and edge features form the initial node set $V_t^0 = \{v_i^0\}^{|\mathcal{T}|}$ and the initial edge set $E_t^0 = \{e_{ij}^0\}^{|\mathcal{T}| \times |\mathcal{T}|}$ respectively, where $t = 1, 2, \ldots, T$.

3.4. Aggregation block

Following graph construction and initialization, the initialized graphs are processed through aggregation blocks to perform feature inference and facilitate knowledge transfer. Each aggregation block comprises a node updating operation, Gated Recurrent Units (GRUs)

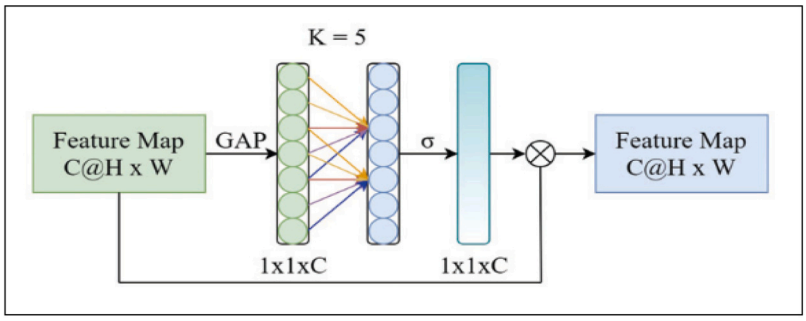

Fig. 5. Efficient Channel Attention (ECA) module, enhancing feature representation by applying global average pooling, 1D convolution, and sigmoid activation to emphasize important channels.

with shared parameters to capture long-term task dependencies, and an edge updating operation. As depicted in Fig. 4, in the kth aggregation block, the input consists of the node feature set V_t^{k-1} and the edge feature set E_t^{k-1} from graph G_t. The aggregation block then outputs the updated node feature set V_t^k and the updated edge feature set E_t^k.

3.4.1. Node update

The node v_i^k of graph G_t is updated by aggregating features from neighboring nodes in proportion to their edge features, followed by a transformation operation:

$$v_i^k = f_v^k\left(\left[\sum_{j\in N_i} \| e_{ij}^{k-1} v_j^{k-1} \| ; v_i^{k-1}\right]; \theta_v^k\right), \tag{3}$$

where $\|\cdot\|$ denotes normalization, $[\cdot;\cdot]$ represents concatenation, and f_v^k is the transformation block's neural network which includes *two convolutional layers, a LeakyReLU activation, and a dropout layer*, as illustrated in Fig. 7. The parameter set θ_v^k is learned during training. The node feature v_i^{k-1} is first concatenated with its aggregated neighboring features and then updated through f_v^k. The result is a new node set $V^k = \{v_1^k, v_2^k, \ldots, v_{|\mathcal{T}|}^k\}$.

3.4.2. Gated recurrent units GRU

Following the node update, V^k and the hidden state h_{t-1}^k from previous tasks are input into the GRU, facilitating the aggregation of prior knowledge and the extraction and transfer of the knowledge gained from the current task.

$$\begin{aligned} z_t &= \sigma(W_z \cdot [h_{t-1}^k, V_t^k]), \\ r_t &= \sigma(W_r \cdot [h_{t-1}^k, V_t^k]), \\ \tilde{h}_t^k &= \tanh(W \cdot [r_t \odot h_{t-1}^k, V_t^k]), \\ h_t^k &= (1 - z_t) \odot h_{t-1}^k + z_t \odot \tilde{h}_t^k, \end{aligned} \tag{4}$$

where W_z, W_r, and W are weight matrices, σ is the sigmoid activation function, and $\odot$ denotes element-wise multiplication. The input to the GRU consists of the hidden state h_{t-1}^k from the previous time step and the node features V_t^k. The hidden state h_0^k is initialized as a zero vector.

3.4.3. Edge update

The edge feature quantifies the degree of dependence between two adjacent nodes by measuring the distance between them. To enhance the model's understanding of these relationships, the edge update process incorporates the similarity between hidden states h_t^k and the edge features e_{ij}^{k-1} from previous aggregation blocks. The updated edge feature is calculated as:

$$e_{ij}^k = f_e^k\left(\|h_i^k - h_j^k\|; \theta_e^k\right) e_{ij}^{k-1}, \tag{5}$$

where $\|\cdot\|$ denotes a normalization operation that enhances the tractability of the similarity calculation. The function f_e^k refers to edge updating neural network responsible for transforming the similarity vector into a specific similarity score, as shown in Fig. 7. This network is further enhanced by integrating the Efficient Channel Attention (ECA) mechanism, which encodes feature maps to produce focused feature representations that optimize similarity measures.

3.4.4. ECA-net

The Efficient Channel Attention (ECA) module enhances Convolutional Neural Networks (CNNs) by selectively emphasizing important channels with minimal additional parameters. As illustrated in Fig. 5, ECA-Net starts with channel-wise global average pooling (GAP) to aggregate convolutional features. A fast 1D convolution with a kernel size K is then applied to generate channel weights, where K defines the scope of local cross-channel interactions. These channel weights are subsequently refined using a sigmoid function σ, ensuring efficient and accurate attention without reducing the dimensionality of the feature maps. This approach preserves the feature map's channel dimension C and spatial dimensions H and W, resulting in improved model performance while maintaining computational efficiency and detailed spatial information.

By integrating the ECA mechanism into the edge update process, the model achieves more precise and optimized feature representations, leading to enhanced performance in tasks requiring accurate similarity measures and classifications.

3.5. Prediction and loss function

Upon completing an aggregation block, the updated node feature set V_t^k and edge feature set E_t^k are propagated as inputs into the subsequent aggregation block. This iterative process allows the model to continuously refine its understanding of node relationships and enhance feature representations. The prediction for each node i within the graph is computed using a softmax function, which operates over the neighborhood of node i. The prediction is given by:

$$\hat{y}_i^k = \text{softmax}\left(\sum_{j\in N_i} e_{ij}^k \cdot \delta(y_j = C_n)\right), \tag{6}$$

where C_n represents the nth class in the classification task, and $\delta(y_j = C_n)$ is the Kronecker delta function, which returns 1 if $y_j = C_n$ and 0 otherwise. This mechanism ensures that the influence of each neighboring node j is weighted by the corresponding edge feature e_{ij}^k, enabling the model to produce a probability distribution over the classes for node i.

The model's performance is optimized by minimizing the binary cross-entropy loss function, which evaluates the discrepancy between the predicted probabilities $\hat{y}_i^k$ and the true labels y_i. The overall loss L is computed by summing the loss contributions across all query edges, aggregation blocks, and tasks. The loss function is formally defined as:

$$L = -\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{i=1}^{N_q}\left[y_i \log(\hat{y}_i^k) + (1 - y_j)\log(1 - \hat{y}_j^k)\right], \tag{7}$$

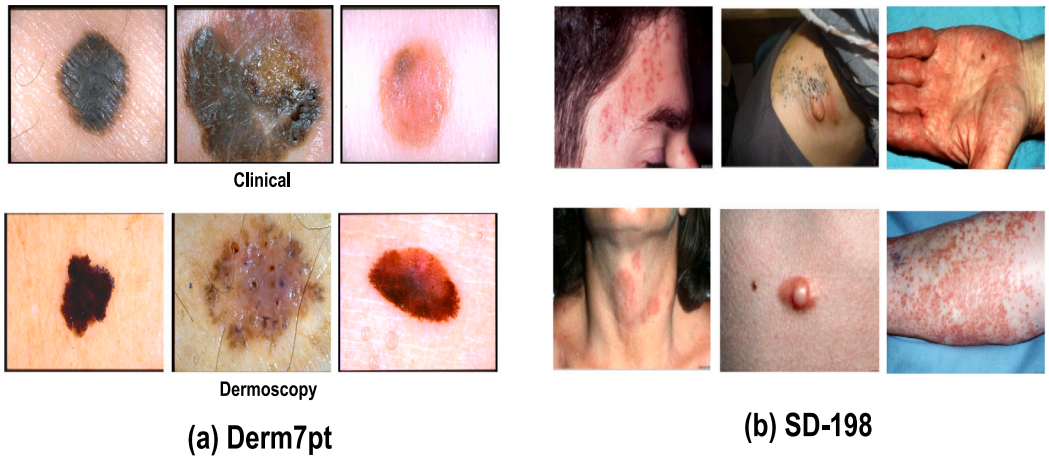

Fig. 6. Representative images from (a) Derm7pt and (b) SD-198 skin lesion datasets, showcasing lesion diversity for few-shot skin disease classification in the FEGGNN framework.

where T is the total number of tasks, K is the number of aggregation blocks, and N_q represents the number of query nodes. This loss function is minimized using backpropagation, allowing the model to iteratively adjust its parameters for better alignment with the training data. The details of our FEGGNN model, including the feature extraction, graph construction, aggregation block operations, and prediction, are summarized in the following algorithm 1.

Algorithm 1 FEGGNN Framework for Few-Shot Skin Disease Classification

Input: D_{train}, D_{test}, T (number of tasks), N (number of classes), K (shots per class), L (number of layers for aggregation blocks)
for $\mathcal{T} \leftarrow 1$ to T **do**
  Step 1: Initialize graph for task $\mathcal{T}$
    $S, Q \leftarrow$ `Create support set (S) and query set (Q)`
    $V_S^0 \leftarrow \text{ACNet}(S)$ ▷ *Extract features from S using ACNet*
    $V_Q^0 \leftarrow \text{ACNet}(Q)$ ▷ *Extract features from Q using ACNet*
    $G_\mathcal{T} \leftarrow$ Initialize graph with V_S^0, V_Q^0 ▷ *Construct graph for task* $\mathcal{T}$
  for $k \leftarrow 1$ to L **do**
    Step 2: Aggregation block processing
      $V_\mathcal{T}^k \leftarrow \text{Node update}(V_\mathcal{T}^{k-1}, E_\mathcal{T}^{k-1})$ ▷ *Update node features by aggregating neighbors' features*
      $H_\mathcal{T}^k \leftarrow \text{GRU}(V_\mathcal{T}^k, H_\mathcal{T}^{k-1})$ ▷ *Transfer knowledge across tasks via GRU*
      $E_\mathcal{T}^k \leftarrow \text{Edge update}(V_\mathcal{T}^k, H_\mathcal{T}^k)$ ▷ *Update edge features to refine relationships*
      $E_\mathcal{T}^k \leftarrow \text{ECA-Net}(E_\mathcal{T}^k)$ ▷ *Refine edge features with ECA-Net (channel attention)*
  end for
  Step 3: Prediction
    $\hat{Y}_Q \leftarrow \text{Softmax}(\text{Predict}(V_Q^L, E_Q^L))$ ▷ *Generate predictions using softmax on updated features*
  Step 4: Loss Calculation
    $L_\mathcal{T} \leftarrow \text{Binary Cross-Entropy Loss}(\hat{Y}_Q, Q)$ ▷ *Calculate loss based on predictions and true labels of query set*
end for
Return: $L = \sum_{\mathcal{T}=1}^{T} L_\mathcal{T}$ ▷ *Sum losses from all tasks for optimization*

4. Experiments

We evaluated our FEGGNN model in transductive settings, where all queries are processed and predicted concurrently, in order to fully evaluate its performance. Two popular benchmark datasets for few-shot skin lesion classification, *Derm7pt* [42] and *SD-198* [43].

4.1. Datasets

The ***Derm7pt*** dataset [42] comprises over 2000 clinical and dermoscopic images across 20 skin lesion types, originally sized at 768 × 512 pixels and resized to 224 × 224 pixels for this study. We used the standard train/test split, excluding the 'miscellaneous' and 'melanoma' classes due to insufficient data. Of the remaining 18 categories, 13 were used for training and 5 for testing, focusing on classes with limited data to assess model generalization in rare cases. The 18 categories were divided into a base set (13 categories, 1892 images) and a novel set (5 categories, 114 images) following previous studies [6,7]. All images were resized to 64 × 64 pixels and augmented with random cropping, rotation, and horizontal flipping. Fig. 6(a) displays sample images from the dataset.

The ***SD-198 dataset*** [43] comprises 6584 clinical images spanning 198 skin disease categories, including acne, dermatitis, and various malignant conditions. Collected from diverse patients and dermatologists using calibrated devices, the images vary in scale, color, exposure, and illumination. The dataset encompasses a wide range of patient demographics, including age, gender, skin color, and disease stages. For experimental consistency with prior studies [6,7], images originally sized at 1640 × 1130 pixels were resized to 224 × 224 pixels. A subset of the dataset was selected, with the evaluation focusing on 70 rarer disease classes with fewer than 20 images each, while 20 classes with 60 images each were used for training. Standard augmentations (scaling, rotation, flipping) were used to ensure robustness in the analysis. Sample images are shown in Fig. 6(b).

4.2. Experimental details

Network Architecture: To ensure a robust comparison with existing few-shot learning models, we utilized three common backbones for the feature embedding module f_{emb}: Conv4, Conv6, and Wide ResNet (WRN-28-10), as referenced in recent studies [7,15,31]. Conv4 is a 4-layer convolutional network, while Conv6 consists of 6 convolutional layers, a fully connected layer, batch normalization, and LeakyReLU activation. We enhanced feature extraction in Conv4, Conv6, and WRN-28-10 by incorporating Asymmetric Convolutional Networks (ACNets) blocks into the first layer of each architecture. The node update network f_v consists of two convolutional layers, LeakyReLU activation, and dropout, while the edge update network f_e includes four convolutional layers, batch normalization, ECA-Net for feature attention, LeakyReLU, and dropout. The overall architecture of our model is detailed in Fig. 7.

Parameter Settings: The proposed FEGGNN framework was implemented using the PyTorch framework and trained on an NVIDIA

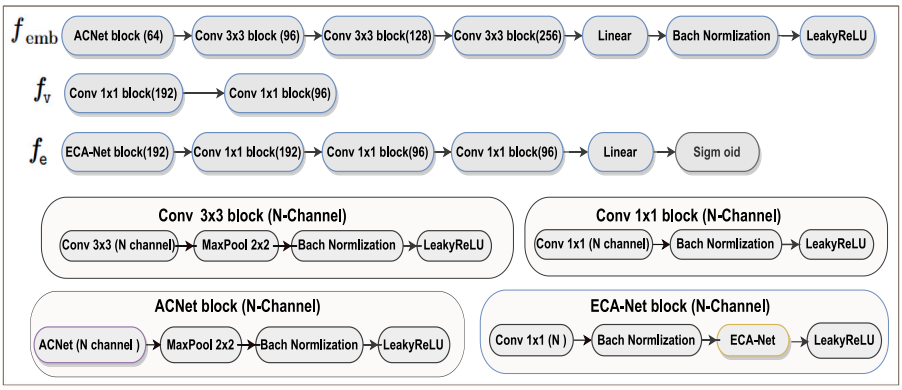

Fig. 7. Architectures of FEGGNN, where f_{emb} represents the ConvNet embedding module with the ACNet block, f_v denotes the node update module, and f_e indicates the edge update module incorporating the ECA-Net block.

Table 1
Performance comparison on Derm7pt for **2-way 1-shot** and **2-way 5-shot** classification, showing FEGGNN and baseline results across backbones (Conv4, Conv6, WRN-28-10) using Accuracy and F1-score, with FEGGNN achieving superior performance.

Method	Backbone	2-way 1-shot		2-way 5-shot	
		Accuracy	F1-score	Accuracy	F1-score
PCN [28]	Conv4	59.98%	58.54%	70.62%	71.85%
SCAN [7]		61.42%	61.90%	72.58%	74.05%
CGProNet [26]		66.13%	66.08%	75.25%	75.90%
SS-DCN [44]		67.32%	–	84.80%	–
FEGGNN		**74.93%**	**74.83%**	82.50%	82.42%
Meta-Derm [6]	Conv6	61.80%	–	76.90%	–
SCAN [7]		62.80%	63.75%	76.65%	73.60%
CGProNet [26]		68.19%	68.09%	79.65%	79.55%
FEGGNN		**75.57%**	**75.90%**	**84.90%**	**84.79%**

Table 2
Performance comparison on SD-198 for **2-Way 1-Shot** and **2-Way 5-Shot** classification, showing FEGGNN and baseline results across backbones (Conv4, Conv6, WRN-28-10) using Accuracy and F1-score, with FEGGNN achieving superior performance.

Method	Backbone	2-way 1-shot		2-way 5-shot	
		Accuracy	F1-score	Accuracy	F1-score
PCN [28]	Conv4	70.03%	70.78%	84.95%	85.87%
SCAN [7]		77.12%	78.00%	90.22%	91.01%
CGProNet [26]		77.89%	77.51%	86.05%	85.82%
SS-DCN [44]		78.52%	–	90.43%	–
FEGGNN		**87.96%**	**87.93%**	**92.64%**	**92.59%**
Meta-Derm [6]	Conv6	65.3%	–	83.7%	–
SCAN [7]		76.75%	77.64%	87.45%	88.28%
CGProNet [26]		77.30%	77.84%	87.60%	87.38%
FEGGNN		**89.02%**	**89.01%**	**93.90%**	**93.86%**
NCA [45]	WRN-28-10	71.27%	71.27%	83.30%	84.23%
Baseline [18]		75.72%	76.64%	88.95%	84.23%
S2M2-R [19]		76.42%	77.51%	90.32%	90.97%
NegMargin [46]		76.85%	77.98%	89.92%	90.65%
PT+NCM [47]		78.25%	78.86%	90.33%	90.90%
BEM [48]		78.32%	78.70%	90.48%	90.94%
EASY [49]		78.80%	79.44%	90.87%	91.43%
SCAN [7]		80.20%	81.21%	91.48%	92.08%
CGProNet [26]		80.23%	80.98%	92.51%	92.48%
FEGGNN		**89.10%**	**89.06%**	**95.19%**	**95.11%**

A100-SXM4-40 GB GPU. We adopted an episodic N-way K-shot learning strategy with 2-way 1-shot and 2-way 5-shot configurations, selecting 5 and 10 query samples per class for the Derm7pt and SD-198 datasets, respectively. The model was optimized using the Adam optimizer with an initial learning rate of 1×10^{-3}, weight decay of 1×10^{-5}, and a dropout rate of 0.1. Training was conducted for 3000 iterations, while testing involved 1000 iterations with a batch size of 16 tasks. The architecture consisted of a 3-layer graph neural network, where each layer incorporated an 8-cell GRU to capture task-specific dependencies and improve robustness in few-shot learning scenarios. We repeated all experiments 20 times to ensure consistency and reduce the impact of data variability, with 95% confidence intervals".

4.3. Results

This section presents the performance and analysis of FEGGNN, including a comparative evaluation of experimental results, significance analysis, and a detailed discussion on the model's computational complexity.

4.3.1. Experiment results

The performance of the proposed Feature-Enhanced Gated Graph Neural Network (FEGGNN) model, along with several others, is summarized in Tables 1 and 2, presenting accuracy and F1-scores for 2-way 1-shot and 2-way 5-shot tasks on the Derm7pt and SD-198 skin disease datasets. FEGGNN consistently outperforms other models across most settings and datasets. On the Derm7pt 2-way 1-shot task, FEGGNN achieves 74.93% accuracy, surpassing CGProNet's 66.13%. The 2-way 5-shot task, records 82.50%, slightly below SS-DCN (84.80%), but surpasses all with Conv6 backbone. On SD-198, FEGGNN leads with 87.96% (1-shot) and 92.64% (5-shot), outperforming SCAN's 77.12% and 90.22%.

We conducted a comprehensive comparison between FEGGNN and twelve state-of-the-art models, designed specifically for rare skin disease classification, utilizing either meta-learning or transfer learning for few-shot learning. Four of these models—PCN [28], Meta-Derm [6], SCAN [7], and CGProNet [26]—are based on meta-learning, while the remaining eight models—NCA [45], Baseline [18], S2M2-R [19], NegMargin [46], PT+NCM [47], BEM [48], EASY [49], and SS-DCN [44]—leverage transfer learning approaches. The results of these transfer learning models were reimplemented and reported in [7], except for SS-DCN.

The meta-learning-based models exhibit various strategies to enhance few-shot learning performance. PCN [28] utilizes Prototypical Clustering Networks, which divide each class into equal sub-clusters to capture intra-class variability and generalize well to novel examples. SCAN [7], on the other hand, employs dynamic clustering to manage varying sub-clusters across different classes, achieving state-of-the-art accuracy in few-shot learning tasks. Meta-Derm [6], built on Reptile and ProtoNets, leverages Group Equivariant Convolutions to improve feature learning, while CGProNet [26] expands feature space representation by integrating CNN and GNN to enhance class representations. The transfer learning-based models also employ unique strategies. NCA [45] applies neighbor component analysis to capture feature correlations, and Baseline [18] implements cross-entropy loss to train a foundational feature encoder. S2M2-R [19] uses self-supervision

and Manifold Mixup to enhance feature learning, while NegMargin [46] applies negative margin softmax loss to improve novel set performance. PT+NCM [47] approximates Gaussian distributions for better feature alignment, and BEM [48] reduces task bias through normalization techniques. EASY [49] introduces ensemble methods and image resizing for data augmentation, and SS-DCN [44] captures subtle inter-class differences, resulting in more discriminative feature representations.

In contrast to existing methods, FEGGNN introduces key innovations that significantly enhance classification performance in few-shot settings. By incorporating Asymmetric Convolutional Networks (ACNets) for precise feature extraction and Efficient Channel Attention (ECA) mechanisms for optimized feature focus, FEGGNN ensures superior feature representation. Additionally, the use of Gated Recurrent Units (GRU) within the graph neural network allows for efficient knowledge transfer across tasks while addressing catastrophic forgetting, a prevalent challenge in few-shot learning. These advancements contribute to FEGGNN's leading performance on both the Derm7pt and SD-198 datasets, making it particularly well-suited for complex and variable skin disease classification tasks. FEGGNN's robustness and accuracy are especially evident when utilizing the Conv6 backbone, where it demonstrates a significant performance advantage over competing models across all key metrics.

4.3.2. Statistical significance analysis

To validate the robustness of the comparative results, a statistical significance analysis was conducted using a paired t-test between FEGGNN and competing models SCAN and PCN. FEGGNN achieved a mean classification accuracy of 84.88% with a standard deviation of 0.18% and a 95% confidence interval (CI) ranging from 84.75% to 85.01%. The paired t-test between FEGGNN and SCAN yielded a p-value < 0.0001, while the comparison between FEGGNN and PCN also resulted in a p-value < 0.0001. Since both p-values are well below the significance threshold of 0.05, these results confirm that FEGGNN's superior performance is statistically significant when compared to both SCAN and PCN, providing strong evidence of its effectiveness in few-shot skin disease classification tasks.

4.3.3. Computational and space complexity

Let n denote the size of task $\mathcal{T}$, i.e., $n = |\mathcal{T}|$. Matrix inversion typically exhibits a time complexity of $O(n^3)$, which can become inefficient for larger task sizes. However, in our setting, where $n = N \times K + Nq$ (with 22 nodes for 1-shot and 30 for 5-shot tasks), the task sizes are relatively small, ensuring that such operations remain computationally feasible. The proposed FEGGNN achieves exceptional computational and space efficiency due to its streamlined graph architecture. This compact structure facilitates rapid information propagation while maintaining low computational and memory overhead. In a comprehensive evaluation on the Derm7pt dataset, as shown in Table 3, FEGGNN was benchmarked against SCAN and PCN models over 800 epochs using identical backbones, hyperparameters, and GPU configurations to ensure fairness. FEGGNN outperformed both competitors by requiring only 13.45 s per epoch, compared to 20.93 s for SCAN and 16.44 s for PCN. Additionally, FEGGNN achieved a lower floating-point operations (FLOPs) count of 31.56 GMac and utilized fewer model parameters (35.44 million) than SCAN (37.12 million) and PCN (36.60 million). These results underscore FEGGNN's superior design, effectively balancing high classification accuracy with optimized computational and space requirements, thereby establishing it as a highly efficient solution for few-shot skin disease classification.

4.4. Ablation studies

In this section, we conduct comprehensive ablation studies to evaluate the individual contributions of each component in the FEGGNN framework, as follows:

Table 3
Computational efficiency comparison of FEGGNN and baseline models based on time per epoch, FLOPs, and model parameters.

Method	Time per epoch	FLOPs	Model parameters
PCN [28]	16.44 s	32.81 GMac	36.60 M
SCAN [7]	20.93 s	32.81 GMac	37.12 M
FEGGNN (Ours)	13.45 s	31.56 GMac	35.44 M

Table 4
Performance of FEGGNN components using the Conv6 backbone in a 2-way, 5-shot setting on the SD-198 test dataset. Each row shows the effect of cumulatively adding one or more components to the baseline (Normal Model).

Configuration	Accuracy (%)	F1-score (%)
Normal model (Baseline)	76.76%	78.59%
+ ECA-Net	79.28%	79.85%
+ GRU	83.41%	84.17%
+ ACNet	89.57%	90.08%
+ All modules	**93.90%**	**93.86%**

4.4.1. Backbone architectures

Our FEGGNN model, after incorporating ACNet, demonstrates significant improvements in few-shot classification tasks across different backbone architectures. In our ablation study, FEGGNN with Conv4 achieved 74.93% and 82.50% accuracy on the Derm7pt dataset for 2-way 1-shot and 2-way 5-shot tasks, respectively. However, the deeper Conv6 and WRN-28-10 backbones, enhanced by ACNet, yielded even better results. FEGGNN with Conv6 reached 84.90% accuracy on Derm7pt and 93.90% on SD-198, while WRN-28-10 led to the highest performance with 95.19% accuracy on SD-198 in the 2-way 5-shot scenario. These results highlight the effectiveness of ACNet in improving feature extraction and classification accuracy, especially when paired with deeper architectures.

4.4.2. Effect of components

A 2-way 5-shot ablation study was conducted on the SD-198 dataset to evaluate the contributions of the FEGGNN model's various components during the 2000-iteration training phase, as illustrated in Fig. 8 (A). Referred to as the "Normal Model", the fundamental model is a conventional graph neural network that lacks any supplementary modifications. Accuracy enhancements are discernible as components are incrementally integrated. The ECA-Net module (+ECA-Net) is integrated to improve feature similarity through channel attention, which leads to a moderate performance increase. The GRU component (+GRU) provides an additional boost to accuracy by enabling the effective transmission of knowledge across tasks, which is notably advantageous in few-shot learning scenarios with limited data. The most substantial improvement in feature extraction capabilities is achieved when the ACNet module (+ACNet) is integrated into the backbone feature extractor. Ultimately, the model's learning capacity and overall performance are enhanced by the cumulative impact of each module, as the combination of all components (+All modules) achieves the utmost accuracy over the 2000 iterations. Table 4 summarizes these ablation study results on the SD-198 test dataset using the Conv6 backbone for a 2-way, 5-shot setting. Each row illustrates how incrementally adding ECA-Net, GRU, and ACNet affects both Accuracy and F1-score, highlighting the progressive gains in performance that culminate in the best results when all modules are combined.

4.4.3. Effect of GRU cell depth

To evaluate the impact of the number of hidden states in the GRU, we compared three different cell configurations of our model on the SD-198 dataset, presenting the validation curves in terms of node classification accuracy, as shown in Fig. 8(B). The GRU-2,4,8 cell configurations represent models utilizing unrolled gated recurrent units with 16-time steps (or tasks) to propagate information across graph layers. The GRU-8 configuration significantly outperforms those with

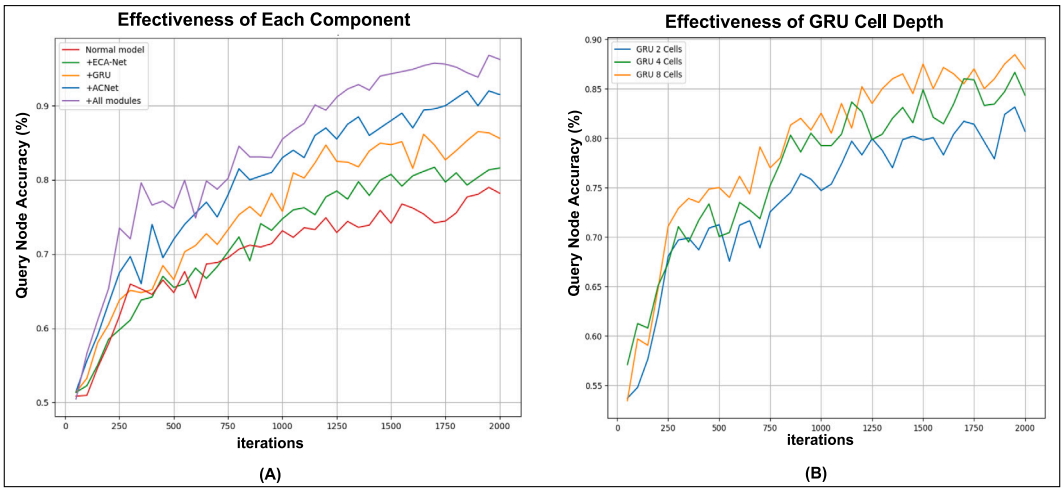

Fig. 8. The influence of (A) individual components and (B) the depth of GRU cells on the accuracy of FEGGNN in a 2-way 5-shot setting using the SD-198 dataset, was evaluated over 2000 iterations.

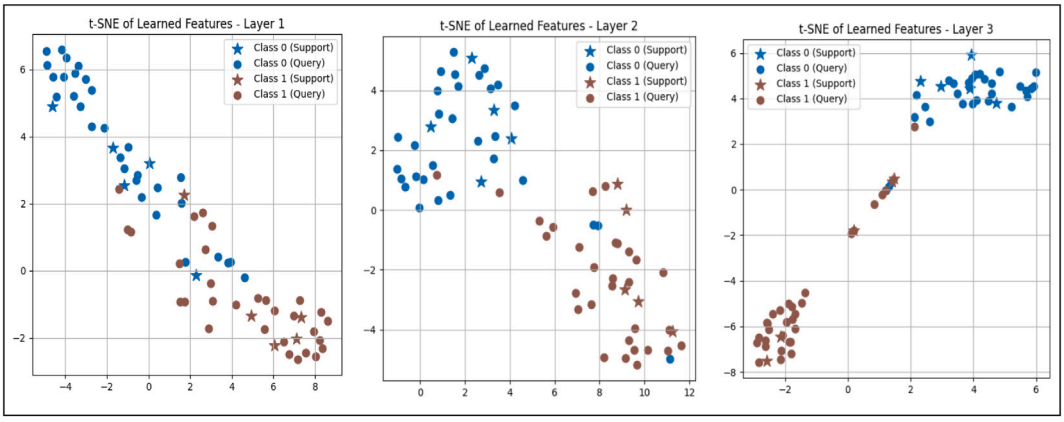

Fig. 9. t-SNE visualizations of feature propagation show progressive improvement in feature separability. As propagation advances, samples with different labels become increasingly separated, while similarly labeled samples form tighter, more distinct clusters.

fewer hidden states (GRU-2,4), confirming that the enhanced memory module improves node representation learning by leveraging prior task knowledge for more accurate predictions.

4.4.4. Effect of graph depth

The effect of graph depth on FEGGNN performance was evaluated by exploring architectures with up to five layers for 2-way 5-shot classification tasks on Derm7pt and SD-198, as detailed in Table 5. Shallow graph depths (L = 1 or L = 2) exhibited limited feature extraction capacity, leading to suboptimal performance. In contrast, increasing the depth to L = 3 substantially improved accuracy, reaching 82.50% on Derm7pt and 92.64% on SD-198. Further increasing the depth (L = 4 and L = 5) offered only marginal gains while incurring significantly higher computational costs and a heightened risk of overfitting, likely due to the limited training/testing dataset in few-shot settings. Hence, a depth of L = 3 was selected as the optimal configuration, where the node update and edge update modules cyclically interact three times to effectively balance accuracy and efficiency.

4.4.5. Visualization of feature propagation

To gain insights into how our model learns and propagates node features, we employed t-SNE to visualize the feature space at different stages of layer propagation. As shown in Fig. 9, both query and support samples with various labels are increasingly pulled apart as propagation progresses, while samples with the same labels are drawn closer together. This demonstrates that the model effectively captures and differentiates between features based on label information, thereby improving the overall classification performance.

Table 5
2-Way 5-Shot classification accuracy on derm7pt and SD-198 with varying FEGGNN depth using a Conv4 backbone, showing the impact of model depth on classification performance.

Dataset	Number of FEGGNN layers				
	L=1	L=2	L=3	L=4	L=5
Derm7pt	75.16%	80.65%	**82.50%**	82.33%	80.27%
SD-198	86.43%	90.77%	**92.64%**	92.59%	91.96%

4.4.6. Grad-CAM interpretations

Fig. 10 illustrates Grad-CAM visualizations on the Derm7pt and SD-198 datasets, showcasing the FEGGNN model's ability to localize diagnostically relevant regions in skin lesion images. Grad-CAM heatmaps highlight critical areas influencing the model's predictions, with warmer colors (red, yellow) indicating higher significance and cooler colors (blue) representing lower relevance. On the Derm7pt dataset, the heatmaps emphasize key features such as asymmetry and pigmentation irregularities, essential for identifying conditions like melanoma. Similarly, on the SD-198 dataset, Grad-CAM highlights central lesion areas linked to abnormal growth and inflammation, which are crucial for detecting rare skin diseases. These visualizations enhance clinical interpretability by correlating the model's predictions

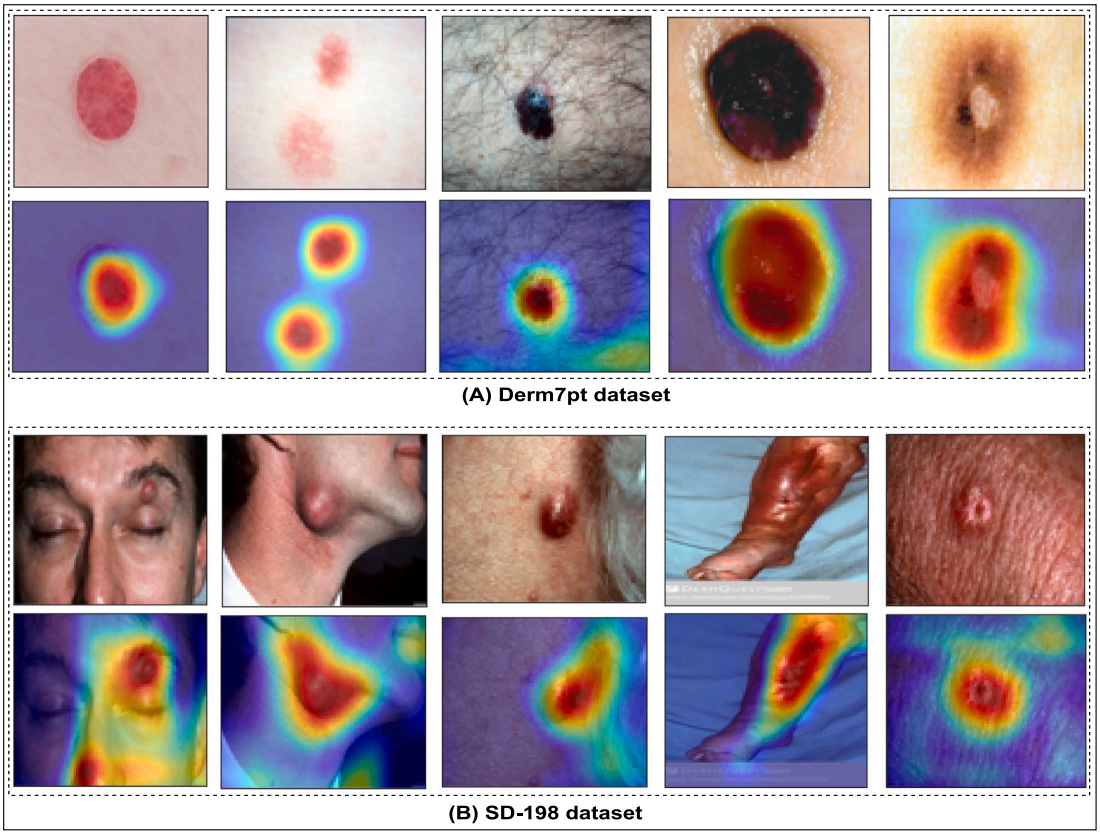

Fig. 10. Grad-CAM visualizations on (A) Derm7pt and (B) SD-198, highlighting diagnostically significant regions in skin lesion images from the Derm7pt and SD-198 datasets, showcasing the FEGGNN model's focus during classification.

with medically significant regions, improving diagnostic accuracy and building trust in few-shot learning contexts where labeled data is limited. The integration of Grad-CAM within FEGGNN provides both performance and explainability, positioning it as a reliable tool for computer-aided dermatological diagnosis (CAD).

4.5. Limitation and future work

While our model effectively enhances feature extraction, employs a focusing mechanism during updates, and facilitates knowledge transfer across few-shot tasks to improve skin disease classification, it is constrained by the scarcity of data in certain classes, with some containing fewer than 20 images. This limitation increases the risk of overfitting, particularly as the network depth grows within the graph structure. Additionally, the small dataset size restricts the scalability of training and testing tasks in the few-shot framework. Future work will focus on addressing these limitations by collecting larger, more diverse skin disease datasets and exploring advanced transfer learning techniques to improve model robustness and generalization, ensuring better performance and reliability in real-world clinical applications.

5. Conclusion

In this paper, we introduced the Feature-Enhanced Gated Graph Neural Network (FEGGNN), a novel framework designed to address the challenges of few-shot skin disease classification in data-limited environments. FEGGNN integrates Asymmetric Convolutional Networks (ACNet) for feature extraction, a Graph Neural Network (GNN) for relationship modeling, Gated Recurrent Units (GRUs) for cyclic feature updates, and an Efficient Channel Attention (ECA-Net) mechanism to refine feature representations, collectively enhancing the accuracy and robustness of classification. Evaluations on benchmark datasets, including Derm7pt and SD-198, demonstrated that FEGGNN outperforms state-of-the-art methods, achieving 84.90% and 95.19% accuracy, respectively, in 2-way 5-shot scenarios, underscoring its effectiveness in computer-aided dermatological diagnosis. Our model's ability to efficiently transfer knowledge across tasks also addresses the issue of catastrophic forgetting, contributing to its strong generalization capabilities for rare skin disease classes. Future work will aim to enhance FEGGNN's generalization potential by exploring advanced attention mechanisms and leveraging self-supervised learning with larger unlabeled datasets, as well as integrating additional data modalities to create a comprehensive CAD system for dermatological applications.

CRediT authorship contribution statement

Abdulrahman Noman: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Formal analysis, Data curation, Conceptualization. **Zou Beiji:** Writing – review & editing, Supervision, Project administration. **Chengzhang Zhu:** Writing – original draft, Supervision, Project administration. **Mohammed Alhabib:** Writing – review & editing, Validation. **Raeed Al-sabri:** Writing – review & editing, Validation.

Ethics statement

This study, adheres to ethical standards in research and publication. The work was conducted in compliance with ethical guidelines for medical and machine learning research. No private or identifiable patient data were used in this study; all data utilized were obtained from publicly available and ethically approved sources.

The research was supported by the National Key R&D Program of China (2018AAA0102100), the Key Research and Development Program of Hunan Province (2022SK2054), the National Natural Science

Foundation of China (No. 62376287), the International Science and Technology Innovation Joint Base of Machine Vision and Medical Image Processing in Hunan Province (2021CB1013), and the Natural Science Foundation of Hunan Province (No. 2022JJ30762). The funding agencies had no influence on the design, implementation, or outcomes of this research.

This paper has been submitted to the Computers in Biology and Medicine journal, and the authors have ensured that the research is conducted responsibly to promote fairness, transparency, and non-discrimination. The proposed methods aim solely to advance scientific understanding and improve healthcare outcomes.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Key R&D Program of China (2018AAA0102100), the Key Research and Development Program of Hunan Province (2022SK2054); the National Natural Science Foundation of China (No. 62376287); the International Science and Technology Innovation Joint Base of Machine Vision and Medical Image Processing in Hunan Province (2021CB1013); and the Natural Science Foundation of Hunan Province (No. 2022JJ30762).

References

[1] M.P. Fu, S.M. Merrill, M. Sharma, W.T. Gibson, S.E. Turvey, M.S. Kobor, Rare diseases of epigenetic origin: Challenges and opportunities, Front. Genet. 14 (2023) 1113086.
[2] S. Arun Kumar, S. Sasikala, Review on deep learning-based CAD systems for breast cancer diagnosis, Technol. Cancer Res. Treat. 22 (2023) 15330338231177977.
[3] C. Qu, T. Zhang, H. Qiao, Y. Tang, A.L. Yuille, Z. Zhou, et al., Abdomenatlas-8k: Annotating 8,000 CT volumes for multi-organ segmentation in three weeks, Adv. Neural Inf. Process. Syst. 36 (2024).
[4] L. Zhou, M. Wang, N. Zhou, Distributed federated learning-based deep learning model for privacy mri brain tumor detection, 2024, arXiv preprint arXiv:2404.10026.
[5] V. Rajinikanth, S. Yassine, S.A. Bukhari, Hand-sketchs based parkinson's disease screening using lightweight deep-learning with two-fold training and fused optimal features, Int. J. Math. Stat. Comput. Sci. 2 (2024) 9–18.
[6] K. Mahajan, M. Sharma, L. Vig, Meta-dermdiagnosis: Few-shot skin disease identification using meta-learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 730–731.
[7] S. Li, X. Li, X. Xu, K.-T. Cheng, Dynamic subcluster-aware network for few-shot skin disease classification, IEEE Trans. Neural Netw. Learn. Syst. (2023).
[8] Y. Luo, Z. Huang, Z. Zhang, Z. Wang, M. Baktashmotlagh, Y. Yang, Learning from the past: continual meta-learning with bayesian graph neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 5021–5028.
[9] Q. Wang, B. Zhang, P. Li, Z. Li, Y. Shi, C. Zhao, C. Zhao, H. Ling, Bidirectional gated edge-labeling graph recurrent neural network for few-shot learning, IEEE Trans. Cogn. Dev. Syst. (2022).
[10] J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, S. Ahn, Bayesian model-agnostic meta-learning, Adv. Neural Inf. Process. Syst. 31 (2018).
[11] C. Finn, K. Xu, S. Levine, Probabilistic model-agnostic meta-learning, Adv. Neural Inf. Process. Syst. 31 (2018).
[12] Y.-X. Wang, R. Girshick, M. Hebert, B. Hariharan, Low-shot learning from imaginary data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7278–7286.
[13] S. Gidaris, N. Komodakis, Generating classification weights with gnn denoising autoencoders for few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 21–30.
[14] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, Adv. Neural Inf. Process. Syst. 29 (2016).
[15] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, Adv. Neural Inf. Process. Syst. 30 (2017).
[16] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H. Torr, T.M. Hospedales, Learning to compare: Relation network for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1199–1208.
[17] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, J. Luo, Revisiting local descriptor based image-to-class measure for few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7260–7268.
[18] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C.F. Wang, J.-B. Huang, A closer look at few-shot classification, 2019, arXiv preprint arXiv:1904.04232.
[19] P. Mangla, N. Kumari, A. Sinha, M. Singh, B. Krishnamurthy, V.N. Balasubramanian, Charting the right manifold: Manifold mixup for few-shot learning, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 2218–2227.
[20] L. Zhang, X. Chang, J. Liu, M. Luo, Z. Li, L. Yao, A. Hauptmann, TN-ZSTAD: Transferable network for zero-shot temporal activity detection, IEEE Trans. Pattern Anal. Mach. Intell. 45 (3) (2022) 3848–3861.
[21] S. Gidaris, N. Komodakis, Dynamic few-shot visual learning without forgetting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4367–4375.
[22] Y. Lifchitz, Y. Avrithis, S. Picard, A. Bursuc, Dense classification and implanting for few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9258–9267.
[23] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135.
[24] L. Wang, Q. Dou, P.T. Fletcher, S. Speidel, S. Li, Medical image computing and computer assisted intervention—Miccai 2022, in: Proceedings of the 24th International Conference, Strasbourg, France, Vol. 12901, 2021, pp. 109–119.
[25] R. Singh, V. Bharti, V. Purohit, A. Kumar, A.K. Singh, S.K. Singh, MetaMed: Few-shot medical image classification using gradient-based meta-learning, Pattern Recognit. 120 (2021) 108111.
[26] A. Noman, Z. Beiji, C. Zhu, M. Alhabib, R. Al-Sabri, CgProNet: Continual graph-based prototypical networks for few-shot skin disease classification, 2024.
[27] W. Zhu, H. Liao, W. Li, W. Li, J. Luo, Alleviating the incompatibility between cross entropy loss and episode training for few-shot skin disease classification, in: Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VI 23, Springer, 2020, pp. 330–339.
[28] V. Prabhu, A. Kannan, M. Ravuri, M. Chaplain, D. Sontag, X. Amatriain, Few-shot learning for dermatological disease diagnosis, in: Machine Learning for Healthcare Conference, PMLR, 2019, pp. 532–552.
[29] T.T. Thein, Y. Shiraishi, M. Morii, Few-shot learning-based malicious IoT traffic detection with prototypical graph neural networks, IEICE Trans. Inf. Syst. 106 (9) (2023) 1480–1489.
[30] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S.J. Hwang, Y. Yang, Learning to propagate labels: Transductive propagation network for few-shot learning, in: 7th International Conference on Learning Representations, 2019.
[31] J. Kim, T. Kim, S. Kim, C.D. Yoo, Edge-labeling graph neural network for few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11–20.
[32] X. Ding, Y. Guo, G. Ding, J. Han, Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1911–1920.
[33] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-Net: Efficient channel attention for deep convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11534–11542.
[34] E.L. Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus, Exploiting linear structure within convolutional networks for efficient evaluation, Adv. Neural Inf. Process. Syst. 27 (2014).
[35] M. Jaderberg, A. Vedaldi, A. Zisserman, Speeding up convolutional neural networks with low rank expansions, 2014, arXiv preprint arXiv:1405.3866.
[36] J. Jin, A. Dundar, E. Culurciello, Flattened convolutional neural networks for feedforward acceleration, 2014, arXiv preprint arXiv:1412.5474.
[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
[38] A. Paszke, A. Chaurasia, S. Kim, E. Culurciello, Enet: A deep neural network architecture for real-time semantic segmentation, 2016, arXiv preprint arXiv:1606.02147.
[39] S.-Y. Lo, H.-M. Hang, S.-W. Chan, J.-J. Lin, Efficient dense modules of asymmetric convolution for real-time semantic segmentation, in: Proceedings of the 1st ACM International Conference on Multimedia in Asia, 2019, pp. 1–6.
[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[41] Z. Sun, M. Ozay, T. Okatani, Design of kernels in convolutional neural networks for image classification, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part VII 14, Springer, 2016, pp. 51–66.

[42] J. Kawahara, S. Daneshvar, G. Argenziano, G. Hamarneh, Seven-point checklist and skin lesion classification using multitask multimodal neural nets, IEEE J. Biomed. Heal. Informatics 23 (2) (2018) 538–546.

[43] X. Sun, J. Yang, M. Sun, K. Wang, A benchmark for automatic visual classification of clinical skin disease images, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11-14, 2016, Proceedings, Part VI 14, Springer, 2016, pp. 206–222.

[44] J. Xiao, H. Xu, D. Fang, C. Cheng, H. Gao, Boosting and rectifying few-shot learning prototype network for skin lesion classification based on the internet of medical things, Wirel. Netw. 29 (4) (2023) 1507–1521.

[45] Z. Wu, A.A. Efros, S.X. Yu, Improving generalization via scalable neighborhood component analysis, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 685–701.

[46] B. Liu, Y. Cao, Y. Lin, Q. Li, Z. Zhang, M. Long, H. Hu, Negative margin matters: Understanding margin in few-shot classification, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16, Springer, 2020, pp. 438–455.

[47] Y. Hu, V. Gripon, S. Pateux, Leveraging the feature distribution in transfer-based few-shot learning, in: International Conference on Artificial Neural Networks, Springer, 2021, pp. 487–499.

[48] Y. Hu, V. Gripon, S. Pateux, Leveraging the feature distribution in transfer-based few-shot learning, in: International Conference on Artificial Neural Networks, Springer, 2021, pp. 487–499.

[49] Y. Bendou, Y. Hu, R. Lafargue, G. Lioi, B. Pasdeloup, S. Pateux, V. Gripon, Easy—ensemble augmented-shot-y-shaped learning: State-of-the-art few-shot classification with simple components, J. Imaging 8 (7) (2022) 179.