Taxonomy-Guided Routing in Capsule Network for Hierarchical Multi-Label Image Classification

Khondaker Tasrif Noor[a,*], Wei Luo[a], Antonio Robles-Kelly[a], Leo Yu Zhang[a,b], Mohamed Reda Bouadjenek[a]

[a]*School of Information Technology, Deakin University, Waurn Ponds, VIC, 3216, Australia*
[b]*School of ICT, Griffith University, Gold Coast, QLD, 4222, Australia*

Abstract

Hierarchical multi-label classification in computer vision presents significant challenges in maintaining consistency across different levels of class granularity while capturing fine-grained visual details. This paper presents HT-CapsNet, a novel capsule network architecture that explicitly incorporates taxonomic relationships into its routing mechanism to address these challenges. Our key innovation lies in a taxonomy-aware routing algorithm that dynamically adjusts capsule connections based on known hierarchical relationships, enabling more effective learning of hierarchical features while enforcing taxonomic consistency. Through the integration of hierarchical agreement mechanisms and taxonomy-guided routing, our model effectively captures the spatial relationships and interdependencies among labels, facilitating improved representation learning. Extensive experiments on six benchmark datasets, including Fashion-MNIST, Marine-Tree, CIFAR-10, CIFAR-100, CUB-200-2011, and Stanford Cars, demonstrate that HT-CapsNet significantly outperforms existing methods across various hierarchical classification metrics. The taxonomy-guided routing mechanism significantly improves both classification accuracy and hierarchical consistency, showcasing the robustness and effectiveness of our approach in handling complex hierarchical multi-label classification tasks.

1. Introduction

Image classification presents a fundamental challenge in computer vision, particularly when dealing with real-world scenarios where images exhibit complex semantic relationships. While traditional classification approaches assign single labels to images, many practical applications require understanding multiple levels of abstraction simultaneously. Hierarchical Multi-Label Classification (HMC) emerges as a critical paradigm that addresses these complexities by enabling images to be classified across multiple semantic levels while respecting predefined taxonomic relationships [1, 2]. Unlike standard multi-label classification, where labels are treated independently[3], HMC explicitly

*Corresponding author
Email addresses: `k.noor@research.deakin.edu.au` (Khondaker Tasrif Noor), `wei.luo@deakin.edu.au` (Wei Luo), `antonio.robles-kelly@deakin.edu.au` (Antonio Robles-Kelly), `leo.zhang@deakin.edu.au`; `leo.zhang@griffith.edu.au` (Leo Yu Zhang), `reda.bouadjenek@deakin.edu.au` (Mohamed Reda Bouadjenek)

models the intrinsic parent-child relationships between classes[4], creating a structured prediction framework that mirrors natural object categorization, making it particularly valuable in domains such as image recognition, document categorization [5], protein function prediction [6], and fine-grained image classification [7]. For instance, in visual recognition tasks, an image might be classified as “vehicle” at the coarsest level, “land vehicle” at an intermediate level, and “car” at the finest level, with each level providing increasingly specific information [8]. This hierarchical approach offers several distinct advantages over alternative methods. First, it enables more nuanced and interpretable predictions by capturing the natural taxonomy of visual concepts [9]. Second, it allows for flexible querying and retrieval at different levels of granularity, making it particularly valuable for applications like content-based image retrieval and visual search [10]. Third, by leveraging hierarchical relationships, these systems can potentially achieve better generalization, especially for fine-grained categories with limited training data [7]. These capabilities have made HMC increasingly relevant across diverse domains, from fine-grained object recognition to medical image analysis [11].

Despite its practical importance, developing effective HMC systems presents several significant challenges. A fundamental difficulty lies in maintaining hierarchical consistency, which requires ensuring that predictions respect the parent-child relationships in the label hierarchy [12, 13]. Traditional deep learning approaches, while powerful for flat classification and multi-label classification, often struggle to maintain these hierarchical constraints, potentially predicting incompatible label combinations that violate the underlying taxonomy. Additionally, most existing methods treat the hierarchical structure as a post-processing constraint rather than integrating it directly into the learning process [14, 15], leading to suboptimal use of taxonomical information. The inherent complexity of simultaneously modelling multiple hierarchical levels while preserving label dependencies increases computational demands and model complexity [14, 16, 17]. These challenges are further compounded in real-world applications where the label hierarchy can be deep and complex [18], with varying numbers of classes at different levels and intricate inter-level relationships. The critical nature of modelling hierarchical feature dependencies is visually demonstrated in Figure 1, which illustrates Class Activation Maps (CAMs) across different hierarchical levels. These visualizations reveal how visual attention patterns should naturally evolve from coarse to fine semantic levels during classification. For example, when classifying vehicles, effective hierarchical models should first attend to general shape and structure at coarse levels (e.g., “transport”), then progressively focus on more specific discriminative features at finer levels (e.g., “automobile” vs “truck”). However, as shown in the figure, traditional approaches often fail to maintain this hierarchical consistency in feature attention, leading to fragmented or inconsistent feature localization across levels. This inconsistency can result in reduced interpretability and reliability of classifications, particularly in fine-grained scenarios where subtle feature differences determine class membership [10]. The importance of coherent feature relationships across hierarchical levels is highlighted as a significant challenge that current methods have not adequately addressed.

Capsule Networks (CapsNets), introduced by Hinton *et al*. in [20], represent a significant advancement in deep learning architecture design. Unlike traditional convolutional neural networks (CNNs) that rely solely on scalar-valued feature maps [21], CapsNets employ groups of neurons called capsules that output vectors representing entity properties and their instantiation parameters. The key innovation of CapsNets lies in their dynamic routing-by-agreement mechanism [20], which enables parts-to-whole relationships to be learned through iterative refinement of connections between capsules at different levels. This architectural characteristic makes CapsNets inherently suitable for capturing hierarchical relationships [13], as they naturally model the compositional nature of features and their hierarchical organization.

However, existing capsule network architectures have not been fully optimized for hierarchical multi-label classification tasks. While the routing-by-agreement mechanism shows promise for hierarchical learning, current approaches do not explicitly incorporate label taxonomy information into the routing process [22, 23]. This limitation results in routing decisions that may not align with known hierarchical relationships between classes. Furthermore, existing methods often treat each level of the hierarchy independently during the routing process [13, 19], missing opportunities to leverage cross-level dependencies and enforce consistency constraints.

To address these limitations, we propose Hierarchical Taxonomy-aware Capsule Network (HT-CapsNet), a novel architecture that explicitly incorporates taxonomical information into the capsule routing process. Our approach introduces a taxonomy-guided routing mechanism that dynamically adjusts routing weights based on known hierarchical relationships between classes. This is achieved through a specialized routing algorithm that combines traditional routing-by-agreement with a taxonomy-aware attention mechanism, ensuring that capsule connections respect the natural hierarchy of the classification task. HT-CapsNet employs a multi-level architecture where each level corresponds to a different granularity in the label hierarchy, with bidirectional information flow enabling both top-down and bottom-up refinement of predictions. The architecture features hierarchical consistency regularization that enforces parent-child relationships during training, and adaptive routing coefficients that automatically adjust based on the hierarchical level and local taxonomic structure. The main contributions of this work can be summarized as: i) We propose an end-to-end capsule network architecture for hierarchical multi-label classification that naturally captures label dependencies through its capsule structure while explicitly incorporating the hierarchical taxonomy information into the network design. ii) We introduce a novel hierarchical routing algorithm that enhances the traditional dynamic routing mechanism by incorporating taxonomy-awareness, enabling more effective learning of hierarchical features while maintaining taxonomical consistency across different levels of the hierarchy. iii) Through extensive experiments on multiple benchmark datasets,

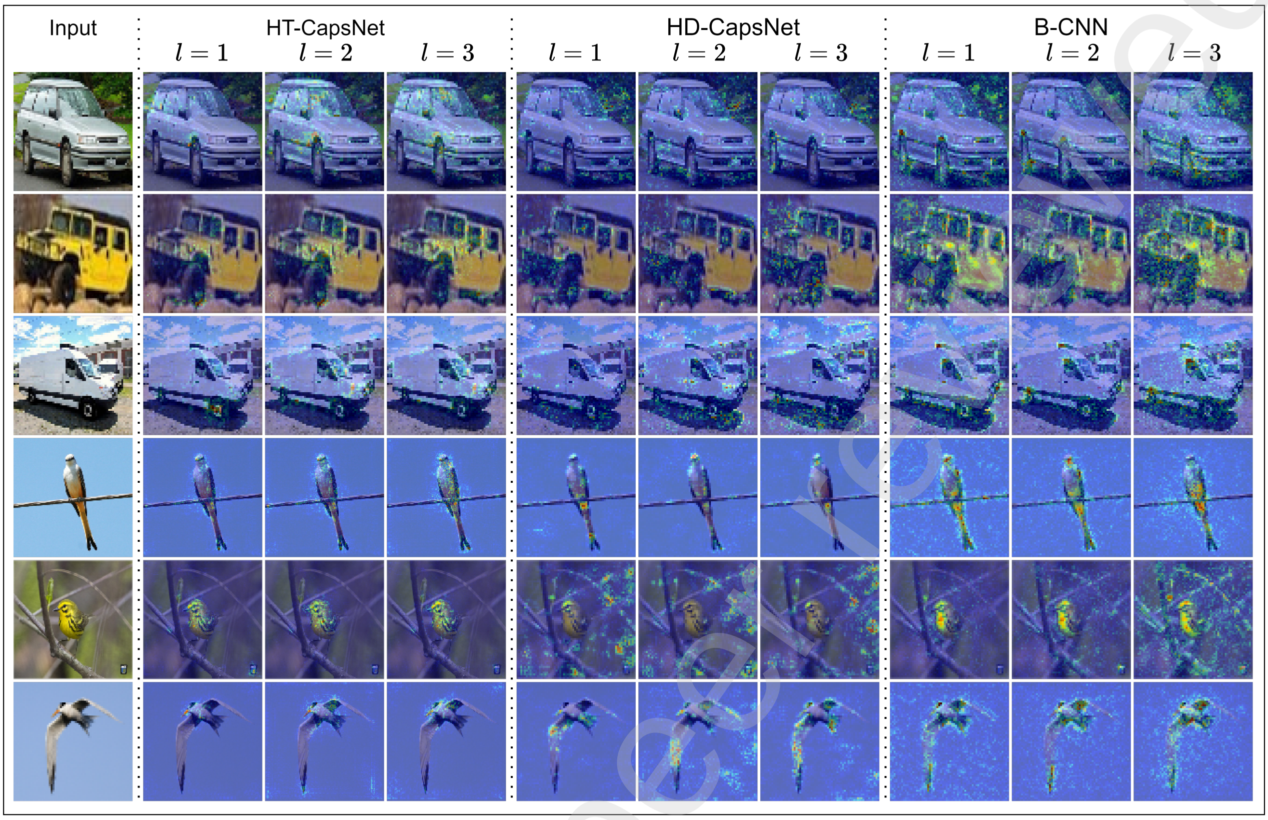

Figure 1: Class Activation Maps (CAMs) for our proposed HT-CapsNet, capsule based HD-CapsNet [19] and convolution based B-CNN [16] baseline models across different hierarchical levels ($l = 1, 2, 3$). Each row shows a different image, with columns showing the input image and corresponding CAMs at each level. HT-CapsNet demonstrates more focused and coherent attention patterns that progressively refine from coarse to fine levels, maintaining hierarchical consistency. For instance, in vehicle images (rows 1-3), attention begins with focused discriminative regions at level 1, gradually expanding to capture broader contextual features at level 3. Similarly, for animal images (rows 4-6), the attention patterns progress from precise focal points to more comprehensive feature regions, demonstrating HT-CapsNet's ability to leverage both fine-grained and holistic features across the hierarchy. This hierarchical attention pattern is notably more coherent in HT-CapsNet compared to the baseline models, which show less structured progression across levels.

we demonstrate that, HT-CapsNet achieves superior performance compared to existing methods across various hierarchical classification metrics. The taxonomy-guided routing mechanism significantly improves both classification accuracy and hierarchical consistency. Our approach maintains computational efficiency while handling complex hierarchical relationships.

The remainder of this paper is organized as follows: Section 2 reviews related work in deep neural networks for hierarchical classification and capsule networks. Section 3 presents our proposed HT-CapsNet architecture and taxonomy-aware routing mechanism in detail. Section 4 describes our experimental setup and results. Section 5 discusses the implications and limitations of our approach, and Section 6 concludes the paper with final remarks and future directions.

2. Related Works

The evolution of deep learning approaches for HMC represents a critical intersection of structured prediction and representation learning. While significant advances have been made in both hierarchical classification methodologies and neural network architectures, the challenge of effectively modelling complex taxonomic relationships while maintaining computational efficiency remains at the forefront of computer vision research [24]. This section examines two streams of research that inform our work: deep neural networks for hierarchical classification and developments in capsule network architectures. We first analyze how deep learning approaches have progressively addressed the challenges of hierarchical classification, highlighting both their contributions and limitations. We then explore the evolution of capsule networks, focusing particularly on their potential for modelling hierarchical relationships and the current gaps in their application to taxonomic learning tasks.

2.1. Deep Neural Networks for HMC

Hierarchical multi-label classification has seen significant developments with the advent of deep learning approaches. Early work in this domain focused on adapting traditional neural networks to handle hierarchical relationships [14, 16, 25], primarily through modified loss functions [26] and output layer structuring [27]. These initial approaches, while innovative, often struggled with maintaining consistency across hierarchical levels. The emergence of convolutional neural networks (CNNs) marked a significant advancement in hierarchical image classification. Several pioneering works proposed architectures that leverage the inherent hierarchical nature of CNN feature maps [15]. A notable approach introduced branched architectures [16, 25], where different network branches specialized in different levels of the hierarchy. These branched architectures address the varying granularity requirements across hierarchical levels by maintaining separate feature extraction pathways, allowing each branch to focus on features relevant to its specific level of abstraction. This architectural pattern proved particularly effective in capturing both coarse-grained features necessary for high-level categorization and fine-grained details required for specific classification. The approach was further enhanced by methods that incorporated attention mechanisms to dynamically weigh features based on their relevance to different hierarchical levels [28]. These attention-enhanced models demonstrated improved performance by learning to focus on discriminative features specific to each level while maintaining overall hierarchical consistency. The success of these approaches highlighted the importance of level-specific feature learning in hierarchical classification tasks, though challenges remained in efficiently coordinating information flow between different branches and maintaining consistent predictions across levels.

Recent developments have focused on more sophisticated approaches to handling hierarchical relationships. One significant line of research explores graph-based neural networks [29, 30], where class hierarchies are explicitly modelled as graphs, allowing the network to learn relationships between different levels directly. Another promising direction involves transformer-based architectures [31] that leverage self-attention mechanisms to capture long-range dependencies across hierarchical levels. Several approaches have been proposed to address the challenge of maintaining hierarchical consistency. These include hierarchical loss functions [26, 19], which explicitly penalize violations of taxonomic constraints, and regularization techniques [32] that encourage feature sharing between related classes across different levels. More recent work has explored probabilistic approaches [7] that model the uncertainty in hierarchical predictions. Despite these advances, several challenges remain. Most existing approaches treat hierarchical relationships as static constraints rather than learnable structures [14, 16, 33]. Additionally, many methods struggle with the trade-off between global hierarchical consistency and local classification accuracy [22, 17]. The computational complexity of these approaches also remains a significant concern, particularly for deep hierarchies with many classes.

2.2. Capsule Networks

Capsule Networks represent a fundamental shift in deep learning architecture design. Since their introduction by Sabour *et al.* [20], they have offered a novel perspective on building more robust and interpretable neural networks. The core innovation of capsules lies in their ability to encode entity properties in vector form, allowing for better preservation of hierarchical relationships and spatial information compared to traditional neural networks [34, 13, 19]. The dynamic routing-by-agreement mechanism, a key component of CapsNets, has seen several important developments. Initial work focused on improving the routing algorithm's efficiency and stability [34, 35]. Subsequent research introduced variations such as self-routing [36], SDA-routing [37] and attention-based routing [38, 39], each offering different approaches to establishing connections between capsules.

Several studies have explored modifications to the basic capsule architecture to enhance its capabilities. These include approaches for handling varying architecture sizes [40], methods for incorporating spatial relationships more effectively [41], and techniques for improving the network's scalability to larger datasets [42]. Recent work has also investigated the integration of modern deep learning concepts such as self-attention mechanisms [39] and residual connections [13] into the capsule framework. In the context of hierarchical classification, capsule networks have shown promising potential. Their ability to model part-whole relationships naturally aligns with hierarchical structure learning [13, 13]. Some approaches have explored using capsules for multi-level feature representation [22, 13], while others have focused on adapting the routing mechanism to handle hierarchical relationships.

However, existing capsule-based approaches for hierarchical classification face several limitations. Most notably, they typically don't explicitly incorporate known taxonomic relationships into the routing process [13, 19]. Additionally, the computational complexity of routing algorithms often limits their application to deeper hierarchies [40]. Our work addresses these limitations by introducing a taxonomy-aware routing mechanism that explicitly incorporates hierarchical relationships while maintaining the computational efficiency necessary for practical applications. This represents a significant advance in both capsule network architecture and hierarchical classification methodology.

3. Method

We consider the problem of learning when the labels follow a hierarchical taxonomy structure with multiple levels, where each level represents a different granularity of classification. Let $X = \{x_i\}_{i=1}^{N}$ denote a training dataset with N samples. For each sample, we have labels at L different hierarchical levels, denoted as $Y = \left\{\{y_i^l\}_{l=1}^{L}\right\}_{i=1}^{N}$ where $y_i^l \in \{0,1\}^{K_l}$ is a one-hot encoded vector subject to $\sum_{k=1}^{K_l} y_{i,k}^l = 1$. Here, K_l denotes the number of classes at level l, typically $K_L > K_{L-1} > \ldots > K_1$. The label y_i^l represents the label for sample x_i at level l. The hierarchical relationships between classes at adjacent levels are encoded in a taxonomy matrix T^l for each level l. Here, $T^l \in \{0,1\}^{K_l \times K_{l+1}}$ for $l = 1, \ldots, L-1$. Each entry $T_{i,j}^l$ indicates whether class j at level $l+1$ is a child of class i at level l, such that,

$$T_{i,j}^l = \begin{cases} 1, & \text{if } j \in \{\text{children of class } i\} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

For any sample x_i, the consistency constraint can be expressed as:

$$y_i^l = y_i^{l+1} \left(T^l\right)^T; \quad \forall\, l \in \{1, \ldots, L-1\} \tag{2}$$

This ensures that if the sample belongs to a class at level $l+1$, it must belong to the corresponding parent class at level l. This hierarchical consistency is crucial for maintaining logical relationships in the prediction hierarchy. To address this hierarchical classification problem, we propose HT-CapsNet, a novel capsule network architecture that explicitly incorporates taxonomical relationships into its architecture and routing mechanism.

3.1. Hierarchical Taxonomy-aware Capsule Network

In this work we propose Hierarchical Taxonomy-aware Capsule Network (HT-CapsNet[1]), that explicitly incorporates class taxonomy information into the routing mechanism of capsule networks. Our architecture leverages the hierarchical structure of class labels while enforcing taxonomic consistency through a specialized routing algorithm. The overall architecture of HT-CapsNet is illustrated in Figure 2, which consists of three primary components: a feature extraction backbone, multiple primary capsule layers (P_l), and multiple taxonomy-aware secondary capsule layers (S_l) for l^{th} hierarchical level.

The feature extraction block in our network is responsible for extracting high-level features from the input data. We employ a standard convolutional neural network (CNN) architecture for this purpose. Let $\phi(x_i \mid \theta_{\mathcal{B}}) \in \mathbb{R}^{H \times W \times C}$ denote the feature maps extracted from input x_i through a convolutional backbone network $\phi(\cdot \mid \theta_{\mathcal{B}})$:

$$F = \phi(x_i \mid \theta_{\mathcal{B}}) \in \mathbb{R}^{H \times W \times C} \tag{3}$$

[1]Our implementation of HT-CapsNet is available at https://github.com/tasrif-khondaker/HT-CapsNet

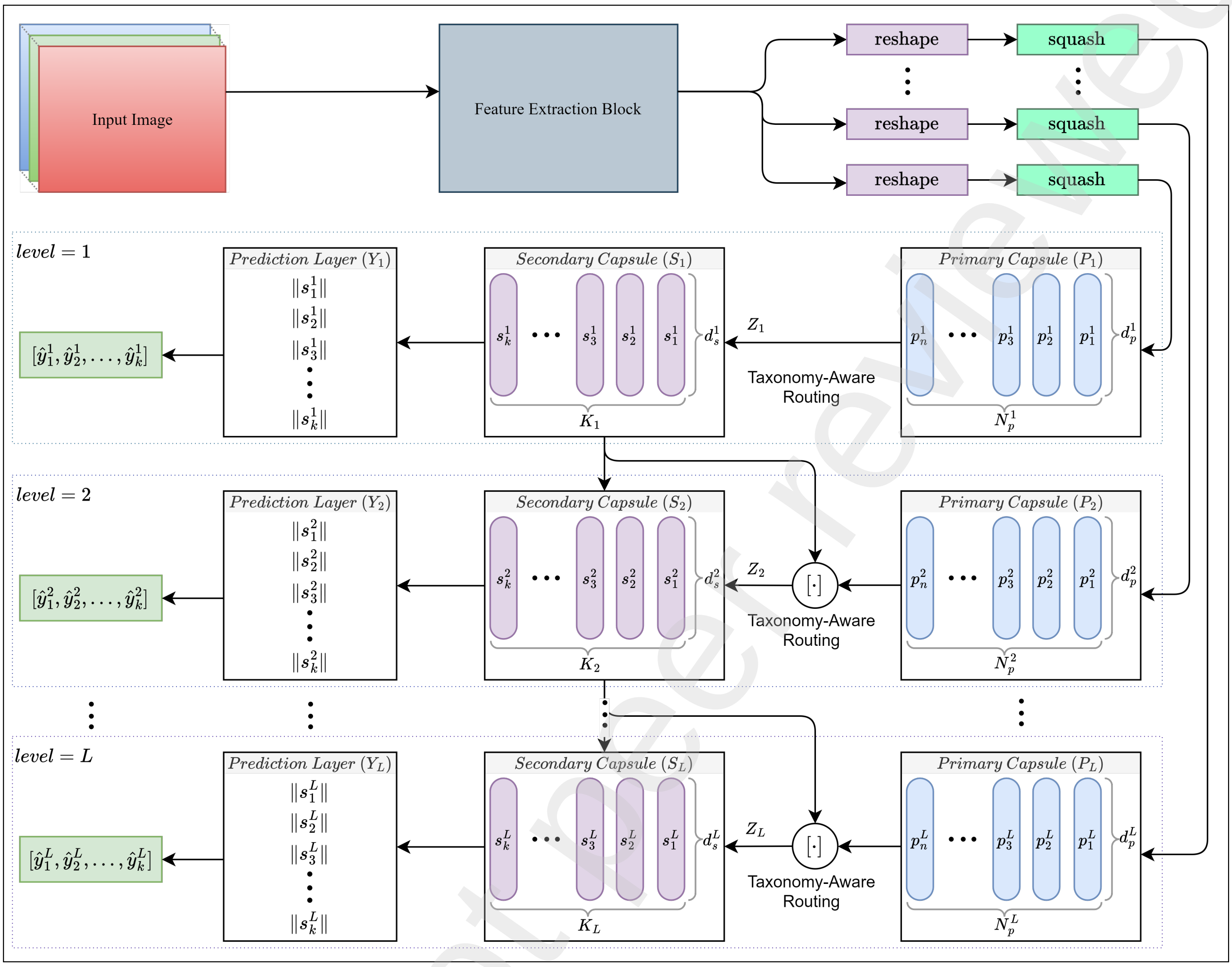

Figure 2: Architecture of the proposed Hierarchical Taxonomy-aware Capsule Network (HT-CapsNet). The network consists of a feature extraction backbone, multiple primary capsule layers (P_l), and multiple taxonomy-aware secondary capsule layers (S_l) for each hierarchical level l. The primary capsules are reshaped from the feature maps extracted by the backbone network, while the secondary capsules are formed based on the predictions from the previous level and the primary capsules. The routing process between primary and secondary capsules, as well as between consecutive secondary capsule layers, is guided by the proposed taxonomy-aware routing mechanism in algorithm 1 to enforce hierarchical consistency. The final predictions are obtained by computing the normalized lengths of the secondary capsule vectors. The network is trained end-to-end using a multi-level loss function that incorporates both classification and hierarchical consistency constraints.

where H, W are the spatial dimensions of the feature maps, C is the number of channels and $\theta_{\mathcal{B}}$ represents the parameters of the backbone network.

In the primary capsule layer (P), as outlined in [20, 34], an essential process is undertaken to transform the feature maps F into capsule vectors. The primary capsule layer is formed by reshaping these features into a set of N_p^l primary capsules, where each capsule is represented by a d_p^l-dimensional vector:

$$P_l = \text{squash}\left(\text{reshape}\left(F\right)\right) \in \mathbb{R}^{N_p^l \times d_p^l} \tag{4}$$

where $N_p^l = \frac{H \times W \times C}{d_p^l}$ represents the number of primary capsules after reshaping the feature maps into capsules of dimension d_p^l. Each primary capsule is denoted as:

$$p_i^l \in \mathbb{R}^{d_p^l}, \quad i \in \{1, \ldots, N_p^l\} \tag{5}$$

The squash function in Equation 4 is a non-linear activation function that ensures the length of each capsule vector is within the range $[0, 1]$, while preserving its orientation. It is defined as:

$$v_o = \text{squash}(v_{in}) = \frac{||v_{in}||^2}{1 + ||v_{in}||^2} \frac{v_{in}}{||v_{in}||} \tag{6}$$

where v_{in} and v_o represent the input and output capsule vectors, respectively.

The secondary capsule layers (S_l) in HT-CapsNet are constructed to capture hierarchical relationships across multiple levels. For each hierarchical level l, there is a taxonomy-aware secondary capsule layer that processes information from two sources: the level-specific primary capsules and, for levels beyond the first, the predictions from the previous level. This dual-input structure enables both feature preservation and hierarchical information propagation. Each secondary capsule layer S_l contains K_l capsules, corresponding to the number of classes at level l. Each capsule represents a distinct class and is characterized by a d_s^l-dimensional vector that encodes the instantiation parameters of that class:

$$S_l = \left\{ s_k^l \in \mathbb{R}^{d_s^l} \right\}_{k=1}^{K_l} \tag{7}$$

where s_k^l represents the capsule vector associated with class k at level l. The connections between these capsules are governed by our novel taxonomy-aware routing mechanism (detailed in Section 3.2), which plays a crucial role in enforcing hierarchical consistency while allowing flexible learning of part-whole relationships. This specialized routing algorithm incorporates the predefined class taxonomy to guide the routing process, ensuring that capsule agreements respect the known hierarchical structure while maintaining the network's ability to discover and learn meaningful hierarchical patterns in the data. The input to each secondary capsule layer is carefully structured to preserve both low-level feature representations and hierarchical context. For each level l, the input Z_l is initially formed as follows:

$$Z_l = \begin{cases} P_l, & \text{if } l = 1 \\ ([P_l; S_{l-1}]\ ,\ S_{l-1}), & \text{if } l > 1 \end{cases} \tag{8}$$

where $[;]$ denotes concatenation along the capsule dimension, and in our implementation, we ensure $d_p^l = d_s^{l-1}$ for $l > 1$ to maintain dimensional compatibility during concatenation. This formulation ensures that while higher levels incorporate predictions from lower levels, they maintain access to the primary feature representations through P_l, preventing information loss in deeper levels of the hierarchy.

The final predictions at each level are obtained by computing normalized lengths of the secondary capsule vectors. For each level l, the prediction layer Y_l transforms the secondary capsule representations into class probabilities:

$$Y_l = \{\hat{y}_k^l\}_{k=1}^{K_l}, \tag{9}$$

where $\hat{y}_k^l$ represents the probability of class k at level l. The class probabilities are computed as follows:

$$\hat{y}_k^l = \frac{\exp\left(\left\|s_k^l\right\|\right)}{\sum_{j=1}^{K_l} \exp\left(\left\|s_j^l\right\|\right)} \tag{10}$$

where $\left\|s_k^l\right\|$ denotes the Euclidean norm of the capsule vector s_k^l. The softmax normalization ensures a proper probability distribution over the classes at each level.

While the architectural design of HT-CapsNet provides the foundation for hierarchical learning, the key innovation lies in how information flows through these components via our proposed taxonomy-aware routing mechanism. Unlike conventional routing mechanisms for capsule networks that overlook hierarchical relationships, our approach explicitly incorporates taxonomic constraints into the routing process, ensuring that the network learns meaningful hierarchical patterns while maintaining taxonomic consistency. This specialized routing algorithm guides the flow of information between capsules, enabling the network to capture both local and global hierarchical relationships in the data.

3.2. Taxonomy-Aware Routing

The key innovation in HT-CapsNet lies in our taxonomy-aware routing algorithm, which explicitly incorporates hierarchical class relationships into the routing process to enforce taxonomic consistency. This mechanism ensures that the capsule agreements align with the known hierarchical structure of the classes, while maintaining the flexibility to learn novel hierarchical patterns. The routing process occurs between primary capsules and each level of secondary capsules, as well as between consecutive levels of secondary capsules, ensuring taxonomic consistency throughout the network. Our approach modifies the routing coefficients based on the predefined taxonomy matrix while maintaining the network's ability to learn flexible part-whole relationships.

The taxonomy-aware routing mechanism operates by integrating three key components: vote generation, taxonomy-guided coefficient computation, and hierarchical agreement calculation. These components work together to ensure that the routing process respects hierarchical relationships while maintaining flexibility in learning part-whole relationships. For each level l, the routing process begins with the computation of prediction vectors (votes) through learnable transformation matrices. Given an input capsule $z_i^l \in Z_l$, the vote for secondary capsule k is computed as:

$$v_{i,k}^l = W_{i,k}^l z_i^l \tag{11}$$

where $W_{i,k}^l \in \mathbb{R}^{d_s^l \times d_p^l}$ is a learnable transformation matrix that maps the input capsule to the prediction vector space of level l.

The taxonomy-aware routing algorithm introduces a fundamentally new approach to routing in capsule networks by incorporating explicit hierarchical relationships into the agreement mechanism. This routing process adaptively guides the flow of information between capsules while enforcing taxonomic consistency across hierarchical levels. The routing coefficients $c_{i,k}^l$ between input capsule i and secondary capsule k at level l are computed as:

$$c_{i,k}^l = \begin{cases} \frac{\exp\left(b_{i,k}^l\right)}{\sum_{j=1}^{K_l} \exp\left(b_{i,j}^l\right)}; & \text{if } l = 1 \\ \frac{\exp\left(\tau_l b_{i,k}^l \cdot m_{i,k}^l\right)}{\sum_{j=1}^{K_l} \exp\left(\tau_l b_{i,j}^l \cdot m_{i,j}^l\right)}; & \text{otherwise} \end{cases} \tag{12}$$

where τ_l is a temperature parameter that controls the sharpness of the routing distribution, $b^l_{i,k}$ is the pre-routing logit, and $m^l_{i,k}$ is a taxonomy-derived mask. For the first level ($l = 1$), standard softmax routing is used since there are no parent-child relationships to consider. For higher levels, the routing coefficients are modulated by the taxonomy mask to enforce hierarchical consistency. The mask $m^l_{i,k}$ is defined as:

$$m^l_{i,k} = (\beta_h - \beta_l) \cdot \sigma\left(\lambda_T\left(T^l_{i,k}\left\|s^{l-1}_{\mathbf{p}(k)}\right\| - \mu_c\right)\right) + \beta_l \tag{13}$$

where β_h and β_l are high and low threshold values that bound the masking effect, effectively creating a soft gating mechanism that allows some flexibility in the routing process while still enforcing taxonomic constraints. The parameters λ_T controls the concentration of the taxonomy influence, $\sigma(\cdot)$ is the sigmoid function, μ_c is the center value, and $T^l_{i,k}$ is the taxonomy matrix value. $\left\|s^{l-1}_{\mathbf{p}(k)}\right\|$ represents the activation strength of the parent capsule, ensuring that routing decisions are influenced by the parent class's confidence.

For levels beyond the first ($l > 1$), we introduce a hierarchical agreement mechanism that ensures consistency between consecutive levels. This mechanism processes both the primary capsule information and the predictions from the previous level's secondary capsules. The hierarchical agreement score $h^l_{i,k}$ for a vote $v^l_{i,k}$ is computed as:

$$h^l_{i,k} = \sigma\left(\sum_{j=1}^{K_{l-1}} g^l_{k,j}\left\langle v^l_{i,k}, W^l_h s^{l-1}_j\right\rangle\right) \tag{14}$$

where $g^l_{k,j} \in \mathbb{R}^{K_l \times K_{l-1}}$ is a hierarchical gate that controls information flow between classes at adjacent levels, $W^l_h \in \mathbb{R}^{d^l_s \times d^{l-1}_s}$ is a dimension transformation matrix that aligns the dimensionality of capsules between levels, and s^{l-1}_j represents the secondary capsule outputs from the previous level. The hierarchical gates $g^l_{k,j}$ and the transformation matrix W^l_h are learned parameters initialized to bias connections according to the taxonomy structure, allowing the network to adaptively refine these relationships during training. The agreement scores are then used to modify the vote vectors, ensuring that routing decisions at higher levels are influenced by the established hierarchical relationships:

$$v^l_{i,k} \leftarrow h^l_{i,k};\ \forall l > 1 \tag{15}$$

This hierarchical agreement term ensures that the routing process at higher levels is influenced by hierarchically-aware representations based on the previous level's predictions, maintaining hierarchical consistency throughout the network.

The final secondary capsule vectors are computed through an iterative routing process that integrates the taxonomy-guided routing coefficients, hierarchical agreements, and attention mechanisms. The initial capsule updates are computed through a two-stage process. First, for each secondary capsule $\hat{s}^l_k$ at level l, based on the routing coefficients $c^l_{i,k}$ and votes $v^l_{i,k}$, an intermediate representation is determined:

$$\hat{s}^l_k = \text{squash}\left(\sum_{i=1}^{N_l} c^l_{i,k} v^l_{i,k}\right) \tag{16}$$

where N_l is the total number of input capsules at level l. The squash function ensures the capsule vectors have unit length while preserving their orientation. After each iteration, the routing logits are updated based on the agreement between the transformed vote vectors $v_{i,k}^l$ (which are the votes after applying hierarchical agreement) and current capsule outputs:

$$b_{i,k}^l \leftarrow b_{i,k}^l + \langle v_{i,k}^l, \hat{s}_k^l \rangle \tag{17}$$

Following the routing iterations, the intermediate capsule representations are refined through level-specific attention mechanisms. For the first level ($l = 1$), self-attention [43] is applied to capture intra-level relationships. Similarly, for higher levels ($l > 1$), multi-head attention [43] is used to capture both local and global hierarchical dependencies. The final capsule representations are obtained through layer normalization:

$$s_k^l = \left\| \hat{s}_k^l + A_l \right\|_n \tag{18}$$

where A_l represents the attention output, and $\|\cdot\|_n$ denotes vector normalization operation that preserves dimensionality. The normalization process standardizes the capsule vectors, ensuring they maintain consistent magnitudes while preserving their directional information. This process ensures that the final capsule vectors are robust and well-calibrated, capturing both local and global hierarchical relationships in the data. This three-stage process involving routing, attention, and normalization creates a sophisticated mechanism for learning hierarchical representations. These process allows the network to maintain taxonomic consistency, capture hierarchical dependencies, and discover complex patterns in the data while ensuring stable learning. Further, the interaction between the taxonomy-guided routing coefficients and hierarchical agreements creates a powerful mechanism that can simultaneously respect class hierarchies while discovering novel patterns in the data. This adaptive routing process allows the network to learn robust hierarchical representations while maintaining consistency with the known taxonomic structure.

The complete routing algorithm integrates these components into an iterative process that progressively refines capsule representations while maintaining both hierarchical consistency and taxonomic relationships. Algorithm 1 provides a detailed step-by-step description of this process, showing how the taxonomy-aware routing mechanism coordinates the flow of information across different levels of the hierarchy while enforcing taxonomic constraints.

3.3. Loss Function

Training HT-CapsNet requires a loss function that effectively handles both the hierarchical nature of the classification task and the capsule-based architecture. Our loss function combines margin-based objectives across different hierarchical levels while ensuring consistency with the taxonomic structure.

For each hierarchical level l, we employ a margin-based loss that operates directly on the capsule lengths. Given the predicted capsule vectors s_k^l and their corresponding lengths $\|s_k^l\|$ from Equation 10, the level-specific loss is defined as:

$$\begin{aligned} \mathcal{L}_l = \sum_{k=1}^{K_l} y_k^l \max\left(0, m^+ - \left\|s_k^l\right\|\right)^2 \\ + \lambda\left(1 - y_k^l\right) \max\left(0, \left\|s_k^l\right\| - m^-\right)^2 \end{aligned} \tag{19}$$

where y_k^l represents the ground truth for class k at level l, m^+ and m^- are margin parameters that define the desired bounds for capsule lengths, and λ is a down-weighting coefficient for absent classes.

Algorithm 1: Hierarchical Taxonomic-Aware Routing (HTR)

Input: Input capsules Z_l, Taxonomy matrix T^l, Level l, Previous level outputs S_{l-1} (if $l > 1$), Number of routing iterations R, Routing Hyper Parameters: $\tau_l, \lambda_T, \beta_h, \beta_l, \mu_c$

Output: Secondary capsule vectors $S_l = \{s_k^l\}_{k=1}^{K_l}$

Procedure `HTR`$(Z_l, T^l, l, S_{l-1}, R)$:
- **forall** $k \in \{1, ..., K_l\}$ *and* $i \in \{1, ..., N_l\}$ **do** ▷ *N_l and K_l are the number capsules in Z_l and S_l*
  - $b_{i,k}^l = 0$ ▷ *Initialize routing logits*
  - $v_{i,k}^l = W_{i,k}^l z_i^l$ ▷ *Generate votes for each pares*
- **for** $r \leftarrow 0$ **to** R **do**
  - **forall** $k \in \{1, ..., K_l\}$ *and* $i \in \{1, ..., N_l\}$ **do**
    - **if** $l > 1$ **then** /* *Process higher-level routing with taxonomy and hierarchical information* */
      - $m_{i,k}^l =$ `TaxonomyGuidedRouting`(T^l, k, S_{l-1}) ▷ *Taxonomy-guided mask for routing*
      - $h_{i,k}^l =$ `HierarchicalAgreement`$(v_{i,k}^l, S_{l-1})$ ▷ *Hierarchcial Agreement*
      - $v_{i,k}^l \leftarrow h_{i,k}^l$ ▷ *Update votes with hierarchical agreement*
      - $c_{i,k}^l = \frac{\exp\left(\tau_l b_{i,k}^l \cdot m_{i,k}^l\right)}{\sum_{j=1}^{K_l} \exp\left(\tau_l b_{i,j}^l \cdot m_{i,j}^l\right)}$
    - **else** /* *Process first-level routing without taxonomy* */
      - $c_{i,k}^l = \frac{\exp\left(b_{i,k}^l\right)}{\sum_{j=1}^{K_l} \exp\left(b_{i,j}^l\right)}$
    - $\hat{s}_k^l = \text{squash}\left(\sum_{i=1}^{N_l} c_{i,k}^l v_{i,k}^l\right)$
    - $b_{i,k}^l \leftarrow b_{i,k}^l + \left\langle v_{i,k}^l, \hat{s}_k^l \right\rangle$ ▷ *Update routing logits*
- **if** $l > 1$ **then**
  - $A_l =$ `MHAttention`$(\mathit{query} = \hat{s}_k^l, \mathit{value} = S_{l-1}, \mathit{key} = S_{l-1})$ ▷ *Standard multi-head attention [43]*
- **else**
  - $A_l =$ `SelfAttention`$(\hat{s}_k^l)$ ▷ *For the first level standard self-attention [43] is used*
- $s_k^l = \left\|\hat{s}_k^l + A_l\right\|_n$ ▷ *Normalization process[44] with default parameters [45]*
- **return** $\left\{s_k^l\right\}_{k=1}^{K_l}$

Function `TaxonomyGuidedRouting`(T^l, k, S_{l-1}):
- $s_{\mathbf{p}(k)}^{l-1} \in S_{l-1} = \left\{s_j^{l-1}\right\}_{j=1}^{K_{l-1}}, \quad \forall k \in \{1, ..., K_l\}$ ▷ *$s_{\mathbf{p}(k)}^{l-1}$ is the parent capsule of s_k^l*
- $m = (\beta_h - \beta_l) \cdot \sigma\left(\lambda_T\left(T_{i,k}^l \left\|s_{\mathbf{p}(k)}^{l-1}\right\| - \mu_c\right)\right) + \beta_l$ ▷ *taxonomic mask*
- **return** m

Function `HierarchicalAgreement`$(v_{i,k}^l, S_{l-1})$:
- $h = \sigma\left(\sum_{j=1}^{K_{l-1}} g_{k,j}^l \left\langle v_{i,k}^l, W_h^l s_j^{l-1}\right\rangle\right)$ ▷ $s_j^{l-1} \in S_{l-1} = \{s_j^{l-1}\}_{j=1}^{K_{l-1}}$
  ▷ *$W_h^l \in \mathbb{R}^{d_s^l \times d_s^{l-1}}$; $g_{k,j}^l \in \mathbb{R}^{K_l \times K_{l-1}}$ are learnable parameters*
- **return** h

To effectively handle the varying complexity across hierarchical levels, we introduce level-specific weights that account for the class distribution. These weights are initialized based on the relative complexity of each level:

$$\omega_l^{init} = \frac{1 - K_l / \sum_{j=1}^{L} K_j}{\sum_{i=1}^{L} \left(1 - K_i / \sum_{j=1}^{L} K_j\right)} \tag{20}$$

where K_l represents the number of classes at level l, and L is the total number of hierarchical levels. The level weights are dynamically adjusted during training to adapt to the model's performance:

$$\omega_l^{(t)} = (1 - \gamma) \frac{\rho_l^{(t)}}{\sum_{i=1}^{L} \rho_i^{(t)}} \tag{21}$$

where $\rho_l^{(t)} = \left(1 - \text{acc}_l^{(t)}\right) \cdot \omega_l^{init}$ represents the error-weighted initial weight at training iteration t, $\text{acc}_l^{(t)}$ is the classification accuracy at level l, and γ is a hyperparameter that controls the balance between initial and dynamic weighting.

The final loss function combines the weighted losses from all hierarchical levels:

$$\mathcal{L}_{total} = \sum_{l=1}^{L} \omega_l^{(t)} \mathcal{L}_l \tag{22}$$

This loss formulation serves multiple purposes in our architecture. First, the margin-based component encourages the network to learn discriminative capsule representations by enforcing separation between present and absent classes. Second, the hierarchical weighting scheme helps balance the learning process across levels of varying complexity. Finally, the dynamic weight adjustment mechanism allows the network to adaptively focus on challenging levels while maintaining stable training across the entire hierarchy. The loss function works in concert with the taxonomy-aware routing mechanism (Section 3.2) to ensure that the learned representations respect both the hierarchical structure of the classes and the part-whole relationships encoded in the capsule architecture.

4. Experiments

In this section, we present a comprehensive overview of the experiments conducted to evaluate the performance of HT-CapsNet in hierarchical multi-label classification tasks. In order to rigorously assess the efficacy of our proposed HT-CapsNet alongside other classifiers delineated within existing scholarly literature, we have employed six distinct image datasets: Fashion-MNIST [46], Marine-Tree [47], CIFAR-10 [48], CIFAR-100 [48], Caltech-UCSD Birds-200-2011 (CUB-200-2011) [49], and Stanford Cars [50]. Moreover, we have performed a comparative assessment of the effectiveness of our proposed HT-CapsNet in relation to the flat classification techniques and hierarchical methods found in the literature. For the flat classification method, we utilized the CapsNet framework described in [20], as well as VGG16 in [51], VGG19 in [51], ResNet-50 in [52], and EfficientNetB7 in [53]. These flat classification techniques focus solely on the most granular class levels and overlook the hierarchical approaches. It is important to mention that the baseline CapsNet in [20] employs a capsule-based architecture combined with the dynamic routing algorithm. In terms of hierarchical classification methods, we have made comparisons with both convolution-based and capsule-based networks. For the convolution-based category, we considered the CNN-based branch hierarchical classifier (B-CNN) from [16], the hierarchical convolutional neural network (H-CNN) in [25], and the Condition-CNN method in [54]. For the capsule-based approaches, we examined ML-

CapsNet in [22], BUH-CapsNet in [23], the H-CapsNet approach in [13], and the HD-CapsNet method in [19]. The experiments are structured to rigorously evaluate the model's ability to capture label correlations and uphold the hierarchical organization of the data. We will detail the benchmark datasets utilized, the experimental setup, and the evaluation metrics employed to measure the performance of HT-CapsNet against existing state-of-the-art HMC methods. Through these experiments, we aim to demonstrate the robustness and superiority of our proposed method.

4.1. Datasets

As mentioned previously, we have utilized six separate image datasets characterized by diverse class quantities and hierarchical relationships throughout our experimental framework. The specifics of the datasets are outlined below:

The *Fashion-MNIST* dataset constitutes a collection comprising $70,000$ grayscale images that represent 10 distinct categories of fashion merchandise. This dataset is systematically partitioned into $60,000$ images designated for training purposes and $10,000$ images allocated for testing. Each image is characterized by dimensions of 28×28 pixels. The dataset exhibits a balanced distribution, with each category containing $6,000$ images. The original dataset lacks any hierarchical arrangement. Consequently, we have established a hierarchical framework for the dataset by organizing the categories into two supplementary levels, as detailed in [25]. The first level includes two main categories, while the second level contains six unique categories. In this hierarchical structure, the first level categories act as parent categories to the second level categories, and the second level categories serve as parent categories to those at the next corresponding level tied to the grouped categories. Thus, the categories in the hierarchical arrangement create a parent-child relationship dynamic.

The *Marine-Tree* dataset comprises a collection of $160,000$ color images depicting marine organisms, categorized into tropical, temperate, and combined subsets. This dataset offers a hierarchical architecture consisting of five distinct levels. In the course of our experiment, we have implemented the settings pertaining to the combined subsets, which encompass 2 classes at the first level, 10 classes at the second level, 38 classes at the third level, 46 classes at the fourth level, and 60 classes at the fifth level. For the purpose of ensuring consistency, we have utilized the initial three levels of the hierarchical structure when conducting comparisons with the benchmark models, while employing all levels for the HT-CapsNet. Additionally, we have standardized the image dimensions to 64×64 pixels to facilitate simplicity.

In a similar manner, the *CIFAR-10* and *CIFAR-100* datasets represent two distinct collections comprising $60,000$ coloured images categorized into 10 and 100 child classes, respectively, with CIFAR-100 being further classified into 20 parent categories. The datasets are partitioned into $50,000$ images designated for training and $10,000$ images allocated for testing purposes. Each image exhibits dimensions of 32×32 pixels. In order to establish a three-level hierarchical framework, we have incorporated 2 supplementary levels for the CIFAR-10 dataset and 1 supplementary level for the CIFAR-

100 dataset, adhering to the methodology outlined by [16]. Consequently, within the CIFAR-10 dataset, the initial supplementary level encompasses 2 classes, while the second supplementary level comprises 7 classes; conversely, in the CIFAR-100 dataset, the initial supplementary level is constituted of 8 classes.

The *CUB-200-2011* dataset comprises color images representing 200 distinct bird species, while the *Stanford Cars* dataset encompasses color images of 196 unique automotive models. We have adhered to the hierarchical framework delineated in [27] for both datasets in order to implement a 3-level hierarchical organization, wherein the training, validation, and testing subsets contain $5,944$, $2,897$, and $2,897$ images for the CUB-200-2011 dataset, and $8,144$, $4,020$, and $4,021$ images for the Stanford Cars dataset, respectively. The first, second, and third tiers comprise 39, 123, and 200 categories for the CUB-200-2011 dataset and 13, 113, and 196 categories, respectively, for the Stanford Cars dataset. In the course of our experiments, we have designated the image dimensions as 64×64 pixels for both datasets.

4.2. Experimental Setup

In our experiments, we have consistently applied a uniform approach to data preprocessing and augmentation across all datasets involved in our experiments. Specifically, we utilized the Standard Scaler for data processing during the training phase of all models. This method ensures that the features of the dataset are normalized, allowing for improved convergence during the training process. To enhance the diversity and robustness of our training data, we implemented the Mix-Up data augmentation technique as introduced in [55]. Mix-Up is a straightforward yet powerful approach that creates new training samples by performing linear interpolation between pairs of randomly selected instances from the training set. This process involves calculating a weighted average of the two chosen samples along with their corresponding labels. The weights used for this interpolation are drawn from a beta distribution characterized by a parameter, denoted as α_m. In our experiments, we fixed the value of α_m at 0.2, which has been shown to effectively balance the trade-off between the original samples and the newly generated ones.

For model optimization, we employed the *Adam* optimizer, which is known for its efficiency and effectiveness in handling sparse gradients. Additionally, we incorporated an exponential decay learning rate scheduler to fine-tune the learning process. Experimentally, we found that setting the initial learning rate to a higher value (0.001) strikes a balance between rapid convergence and the risk of overshooting the minimum. As training progresses, fine-tuning the model parameters becomes crucial to hone in on the optimal solution. To further refine the training, we established a decay rate of 0.95, which is applied after initial 10 epochs throughout all our experiments. This systematic approach to learning rate adjustment aids in stabilizing the training process and enhances the model's performance over time.

As outlined earlier in Section 3.1, the feature extraction module in our HT-CapsNet employs a convolutional backbone network $\phi(\cdot \mid \theta_{\mathcal{B}})$ to extract high-level features from the input data. In all experiments conducted, we utilized the EfficientNetB7 model, as detailed in [53], excluding the fully-connected layer located at the top of the network. Additionally, we carried out pre-training using ImageNet weights $\theta_{\mathcal{B}}$ to set the initial parameters for the backbone of the feature extractor. Throughout all the experiments we conducted, we set the size of the primary capsules d_p^l to 8 for the initial level $l = 1$, and for levels $l > 1$, we specified $d_p^l = d_s^{l-1}$ to ensure compatibility during the concatenation phase. The size of the secondary capsules d_s^l was established at 64 for the first level $l = 1$, and then progressively reduced for the subsequent levels in line with the decay formula $d_s^l = 64 \times 2^{-(l-1)}$ for $\forall l > 1$ and $d_s^l \geq 1$. As a result, the number of primary capsules N_p^l depended on the dimensions of the input image. For the purpose of training the HT-CapsNet model, we employed the taxonomy-aware routing algorithm as outlined in Section 3.2. The routing iterations, referred to as r, were uniformly set at 3 across all the hierarchical tiers. The temperature parameter τ_l, as described in equation 12, was initialized to a value of 0.5. The high and low threshold parameters, β_h and β_l, were consistently maintained at 0.99 and 0.1, respectively. The concentration parameter λ_T was designated a value of 0.5, and the central value μ_c was established as 0.5 in equation 13 throughout all experimental procedures. Furthermore, upper and lower margin values m^+ and m^- were set to 0.9 and 0.1, respectively, for the margin-based loss function in equation 19. The down-weighting coefficient λ was maintained at 0.5 to balance the loss function. We obtained these vales through a series of preliminary experiments to ensure optimal performance.

The foundational CapsNet architecture was trained utilizing the identical hyperparameters delineated in [20], wherein the primary capsules possess dimensions of 8 and the secondary capsules exhibit dimensions of 16, employing dynamic routing for a total of 2 iterations across all datasets. In a similar manner, the models VGG16, VGG19, ResNet-50, and EfficientNetB7 were trained with the identical hyperparameters outlined in their respective research papers as described in [51], [52], and [53]. In the context of the B-CNN architecture, we have implemented the base-B model as described in [16], which does not incorporate pretrained weights. All additional hyperparameters were maintained in accordance with the specifications provided by Zhu and Bain in [16]. Likewise, we adopted the same hyperparameters as articulated in [25] for the H-CNN model, as well as those specified in [54] for the Condition-CNN architecture. For the ML-CapsNet, BUH-CapsNet, H-CapsNet and HD-CapsNet models, we employed the identical hyperparameters as referenced in [22], [23], [13], and [19], respectively, while ensuring that the capsule dimensions remained consistent with those of the HT-CapsNet model to facilitate a fair comparative analysis. Additionally, we conducted extensive training of the models across all datasets for a total of 200 epochs. This rigorous approach ensures a fair and consistent comparison of performance metrics, allowing us to evaluate the effectiveness and robustness of each model under uniform conditions. By maintaining this standard across the various datasets, we aim to eliminate any potential biases that could arise from differing training durations or conditions, thereby enhancing the validity of our comparative analysis.

Traditional evaluation metrics, including accuracy, precision, recall, and F1-score, prove inadequate for hierarchical classification models [1] as they overlook the hierarchical structure inherent in datasets. In complex class configurations, where instances may be classified across multiple levels, these metrics fail to accurately capture the model's adeptness in navigating and rendering precise predictions. The misclassification of labels at higher hierarchical levels is markedly more consequential than at lower levels. However, conventional metrics equate all misclassification, thus neglecting the critical nature of hierarchical interrelations. To rigorously evaluate the HT-CapsNet model, we employ both traditional and hierarchical metrics. Beyond standard per-level accuracy, we compute the hierarchical mean accuracy $\hat{\text{Acc}}@k$, which considers the top-k predictions at each level. Specifically, $\hat{\text{Acc}}@1$ represents the harmonic mean of accuracies across all levels considering only the top prediction, while $\hat{\text{Acc}}@5$ considers the top-5 predictions, providing insight into the model's ability to rank correct labels highly even when the top prediction is incorrect. Additionally, we utilize specialized hierarchical metrics including hierarchical precision (hP), recall (hR), F1-score (hF1), consistency (Cons), and exact match score (EM) following the footsteps of [13] to provide a comprehensive evaluation of the model's performance in hierarchical classification tasks. Hierarchical Precision quantifies the ratio of accurately predicted labels to all labels predicted, while Hierarchical Recall measures the proportion of correctly predicted true labels against all true labels. The Hierarchical F1-score integrates these metrics into a singular evaluative measure, encapsulating the model's efficacy in hierarchical classification contexts. Similarly, the Consistency score serves as a metric indicating the extent to which test instances align with the hierarchical structure, independent of their accuracy. This score is represented as a percentage, reflecting the proportion of aligned test instances. The Exact Match score assesses the percentage of predictions that entirely correspond to the ground truth at each hierarchical level, offering insights into the accuracy with which the predictions conform to the actual dataset.

4.3. Results

Now we turn our attention to the outcomes produced by our proposed HT-CapsNet model in relation to the current standard hierarchical multi-label classification techniques. We provide an in-depth examination of the performance metrics achieved across the six benchmark datasets, emphasizing the model's proficiency in effectively capturing hierarchical relationships and label correlations. We begin by assessing the performance of the HT-CapsNet model against the basic flat baseline models, namely CapsNet, VGG16, VGG19, ResNet-50, and EfficientNetB7, before moving on to a comparative assessment with the hierarchical models, which include B-CNN, H-CNN, Condition-CNN, ML-CapsNet, BUH-CapsNet, H-CapsNet, and HD-CapsNet. Following this, we evaluate the performance of HD-CapsNet in comparison to its ablation versions, as outlined in Section 4.4. The results of our experiments are presented in Tables 1, 2, and 3, which provide a comprehensive overview of the performance metrics achieved by the HT-CapsNet model and the benchmark models across the six benchmark datasets. Our experimental results demonstrate consistently superior performance of HT-CapsNet across all evaluated datasets, with particularly notable improvements in complex fine-grained classification tasks. The performance advantages become more pronounced as the hierarchical structure deepens and the classification task becomes more challenging.

HT-CapsNet exhibits robust performance across all hierarchical levels, with the most significant improvements observed in deeper levels where traditional methods typically struggle. This pattern suggests that our taxonomy-aware routing mechanism effectively leverages hierarchical relationships to maintain classification accuracy even at finer granularities. The performance gap between HT-CapsNet and baseline models widens as task complexity increases, indicating better scalability to challenging scenarios. In studies involving less complex datasets such as Fashion-MNIST, while HT-CapsNet demonstrates certain enhancements, the extent of the advantage remains relatively limited owing to the straightforward hierarchical architecture, as evidenced in Table 1. Conversely, as the complexity of the dataset escalates, the advantages conferred by our methodology become increasingly evident. In the case of Marine-tree, the performance benefits augment significantly at deeper hierarchical levels, indicating a superior capacity for managing intricate hierarchical relationships.

The results on the CIFAR datasets presented in Table 2 reveal a similar trend, with CIFAR-100's more complex hierarchy highlighting HT-CapsNet's superior hierarchical learning capabilities. The most striking improvements appear in fine-grained classification challenges for the CUB-200-2011 and Stanford Cars datasets, as illustrated in Table 3. Here, HT-CapsNet significantly outperforms existing methods, showcasing its ability to capture subtle hierarchical relationships and fine-grained distinctions. This pattern suggests that our taxonomy-aware routing mechanism is particularly adept at differentiating nuanced features while preserving hierarchical consistency.

The hierarchical metrics reveal several interesting patterns. First, HT-CapsNet maintains higher consistency scores across all datasets, indicating better preservation of hierarchical relationships. The improvements in hierarchical precision and recall become more pronounced as the taxonomy becomes more complex, suggesting that our model better captures intricate class relationships. The exact match scores show particularly significant improvements in fine-grained datasets, indicating better complete path prediction capability. For traditional flat classification approaches (VGG16, VGG19, ResNet-50, EfficientNetB7, and CapsNet), we used the predictions at the finest level to derive predictions for parent levels, as these models do not inherently utilize the hierarchical structure of the taxonomy [1]. While this approach ensures prediction consistency by definition, it results in substantially lower overall performance across all hierarchical metrics, highlighting the importance of explicitly modelling hierarchical relationships during the learning process.

Table 1: Performance evaluation on Fashion-MNIST [46] and Marine-tree [47] datasets, comparing HT-CapsNet against baseline methods. The results present accuracy at different hierarchical levels and include hierarchical metrics. The level-wise accuracy demonstrates a progressive improvement as the classification progresses from coarse to fine-grained levels. Meanwhile, the hierarchical metrics evaluate the model using hierarchical information throughout the classification process. The best and second-best results are highlighted in ■ and ■ colors, respectively.

Dataset	Models	Level Wise Accuracy (%)			Hierarchical Metrices (%)						
		Level 1	Level 2	Level 3	$\hat{\text{Acc}}$ @ 1	$\hat{\text{Acc}}$ @ 5	hP	hR	hF1	Cons	EM
Fashion-MNIST	VGG16 [51]	99.76	94.96	89.78	94.66	98.31	94.83	96.83	95.82	–	89.78
	VGG19 [51]	99.64	93.25	89.22	93.84	96.35	93.14	95.54	94.32	–	89.22
	ResNet-50 [52]	99.57	95.23	90.31	94.89	97.49	95.04	95.04	95.04	–	90.31
	EfficientNetB7 [53]	98.90	91.92	84.91	91.55	95.92	91.91	91.91	91.91	–	84.91
	CapsNet [20]	99.62	95.89	91.90	95.70	97.80	91.90	91.90	91.90	–	91.90
	B-CNN [16]	99.63	95.44	92.33	95.71	99.89	95.77	96.48	96.07	96.73	90.44
	H–CNN [25]	99.79	96.76	93.16	96.49	99.95	96.55	96.79	96.65	98.88	92.58
	Condition-CNN [54]	99.78	96.65	93.42	96.55	99.33	96.65	96.84	96.73	99.16	92.85
	ML-CapsNet [22]	99.70	95.89	92.10	95.80	99.74	95.85	96.19	95.99	98.35	91.31
	BUH-CapsNet [23]	99.89	97.53	94.75	97.34	99.46	97.38	97.41	97.40	99.80	94.68
	H-CapsNet [13]	99.73	97.06	93.95	96.86	99.86	96.86	97.36	97.07	97.60	92.69
	HD-CapsNet [19]	99.92	97.78	94.83	97.47	99.44	97.51	97.54	97.52	99.84	94.70
	HT-CapsNet	99.93	97.79	94.98	97.52	99.65	98.01	98.26	98.14	99.90	95.90
	HT-CapsNet†	97.92	92.72	88.94	93.05	96.66	95.07	95.32	95.19	97.90	90.89
	HT-CapsNet‡	96.45	90.53	86.38	90.93	91.83	90.32	90.55	90.43	96.45	88.77
Marine-tree	VGG16[51]	88.81	75.71	46.50	65.25	80.00	73.67	73.67	73.67	–	46.50
	VGG19 [51]	88.92	76.90	48.12	66.62	80.09	73.82	73.82	73.82	–	48.12
	ResNet-50 [52]	87.40	73.05	50.76	66.92	77.19	70.40	70.40	70.40	–	50.76
	EfficientNetB7 [53]	86.70	71.55	48.01	64.74	75.38	68.75	68.75	68.75	–	48.01
	CapsNet [20]	86.36	70.34	46.73	63.56	74.52	46.73	46.73	46.73	–	46.73
	B-CNN [16]	88.28	75.88	54.48	69.99	93.22	72.69	77.03	74.42	80.63	47.29
	H–CNN [25]	88.25	75.14	49.99	67.20	90.73	70.66	75.21	72.47	78.13	44.72
	Condition-CNN [54]	88.75	76.64	53.99	70.03	92.14	72.91	76.46	74.34	82.66	49.10
	ML-CapsNet [22]	86.62	68.21	37.06	56.40	76.24	62.91	66.79	64.45	79.92	34.30
	BUH-CapsNet [23]	88.48	76.49	52.33	68.99	92.39	72.35	73.17	74.07	91.78	52.53
	H-CapsNet [13]	88.38	77.49	52.44	69.30	95.81	72.93	80.97	76.74	83.07	54.85
	HD-CapsNet [19]	89.88	77.50	57.15	72.24	92.15	75.02	76.04	75.44	94.47	55.59
	HT-CapsNet	90.76	81.19	61.12	75.58	93.67	77.49	78.26	77.80	95.88	60.19
	HT-CapsNet†	85.12	74.18	53.37	68.24	88.98	73.62	74.35	73.91	90.88	54.19
	HT-CapsNet‡	83.77	71.20	50.54	65.54	87.11	72.07	72.78	72.36	88.88	52.19

† Denotes the HT-CapsNet without the taxonomy guided routing (taxonomy-based masking) in the routing process.

‡ Denotes the HT-CapsNet without the hierarchcial agreement between the capsules in different levels of the taxonomy.

The t-SNE visualizations in Figure 3 provide compelling evidence of HT-CapsNet's superior representation learning capabilities compared to baseline models. The visualizations elucidate several piv-

Table 2: Performance evaluation on CIFAR-10 [48] and CIFAR-100 [48] datasets, comparing HT-CapsNet against baseline methods. The results present accuracy at different hierarchical levels and include hierarchical metrics. The level-wise accuracy demonstrates a progressive improvement as the classification progresses from coarse to fine-grained levels. Meanwhile, the hierarchical metrics evaluate the model using hierarchical information throughout the classification process. The best and second-best results are highlighted in ■ and ■ colors, respectively.

Dataset	Models	Level Wise Accuracy (%)			Hierarchical Metrices (%)						
		Level 1	Level 2	Level 3	Acc @ 1	Acc @ 5	hP	hR	hF1	Cons	EM
CIFAR-10	VGG16 [51]	96.22	86.89	75.36	85.30	95.42	89.49	90.49	89.99	–	75.36
	VGG19 [51]	95.58	87.13	76.45	85.67	80.59	89.30	89.31	89.31	–	76.45
	ResNet-50 [52]	92.00	72.88	65.01	75.05	89.20	76.63	76.63	76.63	–	65.01
	EfficientNetB7 [53]	86.23	52.28	41.68	54.83	81.18	60.06	60.06	60.06	–	41.68
	CapsNet [20]	93.19	76.53	70.42	78.95	90.60	70.42	70.42	70.42	–	70.42
	B-CNN [16]	96.08	87.13	84.54	88.98	96.40	89.26	91.48	90.18	89.72	78.99
	H–CNN [25]	96.01	86.71	81.29	87.59	99.49	87.89	89.90	88.72	90.21	76.88
	Condition-CNN [54]	95.86	83.78	79.74	85.94	99.62	86.56	88.36	87.30	91.30	75.30
	ML-CapsNet [22]	97.95	90.03	86.78	91.35	99.16	91.38	92.24	91.74	95.47	85.24
	BUH-CapsNet [23]	98.72	93.81	90.84	94.35	99.63	94.41	94.59	94.48	99.06	90.56
	H-CapsNet [13]	97.61	92.58	91.12	93.69	99.28	93.92	94.60	94.74	91.24	86.65
	HD-CapsNet [19]	98.79	94.28	91.22	94.66	99.08	94.74	94.89	94.80	99.18	90.95
	HT-CapsNet	99.10	95.20	91.80	95.27	99.40	95.64	95.73	95.68	99.45	91.50
	HT-CapsNet†	96.17	89.27	84.75	89.82	95.42	91.81	91.90	91.86	96.45	85.50
	HT-CapsNet‡	94.80	87.24	82.87	88.03	93.44	89.90	89.99	89.94	94.44	83.39
CIFAR-100	VGG16 [51]	71.71	59.14	37.67	52.26	63.11	58.51	58.51	58.51	–	37.67
	VGG19 [51]	71.52	60.15	38.41	52.97	61.69	59.33	58.33	58.83	–	38.41
	ResNet-50 [52]	58.26	45.11	33.82	43.54	52.43	45.73	45.73	45.73	–	33.82
	EfficientNetB7 [53]	51.35	38.13	27.65	36.64	46.03	39.04	39.04	39.04	–	27.65
	CapsNet [20]	56.53	45.06	34.93	43.79	53.17	34.93	34.93	34.93	–	34.93
	B-CNN [16]	71.08	61.99	56.38	62.58	90.25	64.41	73.42	67.93	56.87	38.90
	H–CNN [25]	74.00	67.27	51.40	62.72	88.82	64.23	71.67	67.14	60.27	40.49
	Condition-CNN [54]	73.38	61.27	47.91	59.03	86.32	61.07	67.18	63.45	65.01	39.50
	ML-CapsNet [22]	78.73	70.15	60.18	68.85	89.81	69.50	75.65	71.89	68.92	50.29
	BUH-CapsNet [23]	86.03	77.83	64.87	75.21	92.40	76.04	77.87	76.75	89.81	62.53
	H-CapsNet [13]	80.31	75.68	65.74	73.39	90.08	76.93	78.65	77.12	65.25	53.92
	HD-CapsNet [19]	86.93	79.31	66.38	76.58	91.00	77.43	79.20	78.12	89.80	64.41
	HT-CapsNet	87.17	80.22	67.58	77.45	93.41	78.55	80.33	79.43	91.25	66.65
	HT-CapsNet†	80.73	72.44	58.44	69.28	87.81	73.83	75.51	74.66	85.20	59.59
	HT-CapsNet‡	77.35	69.27	55.37	66.05	85.00	71.48	73.10	72.28	82.25	56.64

† Denotes the HT-CapsNet without the taxonomy guided routing (taxonomy-based masking) in the routing process.

‡ Denotes the HT-CapsNet without the hierarchcial agreement between the capsules in different levels of the taxonomy.

otal insights. First, HT-CapsNet exhibits clearer separation between transport and animal categories at Level-1, with more compact and well-defined clusters. This suggests better high-level feature discrimination. Second, at Level-2, HT-CapsNet maintains clear boundaries between sub-categories

Table 3: Performance evaluation on Caltech-UCSD Birds-200-2011 (CUB-200-2011) [49] and Stanford Cars [50] datasets, comparing HT-CapsNet against baseline methods. The results present accuracy at different hierarchical levels and include hierarchical metrics. The level-wise accuracy demonstrates a progressive improvement as the classification progresses from coarse to fine-grained levels. Meanwhile, the hierarchical metrics evaluate the model using hierarchical information throughout the classification process. The best and second-best results are highlighted in ■ and ■ colors, respectively.

Dataset	Models	Level Wise Accuracy (%)			Hierarchical Metrices (%)						
		Level 1	Level 2	Level 3	$\widehat{Acc}$ @ 1	$\widehat{Acc}$ @ 5	hP	hR	hF1	Cons	EM
CUB-200-2011	VGG16 [51]	26.74	15.61	10.03	15.61	19.83	17.79	17.79	17.79	–	10.03
	VGG19 [51]	23.07	14.52	8.52	13.06	20.03	17.03	17.03	17.03	–	8.52
	ResNet-50 [52]	25.40	12.20	7.62	11.87	16.16	15.07	15.07	15.07	–	7.62
	EfficientNetB7 [53]	15.85	5.58	2.89	5.10	9.30	8.11	8.11	8.11	–	2.89
	CapsNet [20]	17.67	8.04	4.59	7.52	11.87	4.19	4.59	4.00	–	4.59
	B-CNN [16]	34.00	17.60	13.15	18.49	43.64	21.65	31.49	25.27	14.74	3.24
	H–CNN [25]	32.43	16.02	6.27	11.87	32.81	17.11	24.94	19.98	12.92	2.21
	Condition-CNN [54]	38.97	20.88	13.37	20.22	54.17	23.35	28.04	25.97	23.47	7.58
	ML-CapsNet [22]	35.01	20.30	13.75	19.92	37.79	23.05	29.14	25.35	25.26	8.55
	BUH-CapsNet [23]	37.76	20.95	13.36	20.13	42.44	23.26	29.21	25.52	26.21	7.90
	H-CapsNet [13]	31.76	21.59	14.13	20.19	47.03	23.13	30.12	25.94	13.63	5.80
	HD-CapsNet [19]	40.42	21.61	14.39	21.35	40.18	23.47	30.33	26.01	27.34	8.63
	HT-CapsNet	58.06	42.49	30.67	40.89	67.75	43.13	48.00	45.03	59.13	24.09
	HT-CapsNet†	48.45	32.42	20.44	29.88	62.33	39.68	44.16	41.43	49.13	16.08
	HT-CapsNet‡	43.05	27.74	15.13	23.93	58.95	37.53	41.76	39.18	44.13	11.08
Stanford Cars	VGG16 [51]	21.67	4.94	3.33	5.46	9.24	9.98	9.98	9.98	–	3.33
	VGG19 [51]	23.53	5.84	3.84	6.33	5.02	10.74	10.74	10.74	–	3.84
	ResNet-50 [52]	23.49	6.38	4.37	7.01	10.85	11.41	11.41	11.41	–	4.37
	EfficientNetB7 [53]	23.83	4.79	2.83	4.97	8.75	10.48	10.48	10.48	–	2.83
	CapsNet [20]	23.75	6.44	4.58	7.21	11.27	4.05	4.58	4.08	–	4.58
	B-CNN [16]	34.94	9.05	9.38	12.21	32.11	18.17	27.96	21.78	7.44	1.62
	H–CNN [25]	33.49	10.55	6.83	11.07	28.91	16.78	25.55	20.02	9.14	1.56
	Condition-CNN [54]	43.07	16.14	14.00	19.16	45.05	24.91	35.48	28.87	15.24	4.49
	ML-CapsNet [22]	41.31	14.75	10.50	16.02	33.65	21.27	28.40	23.97	22.86	5.26
	BUH-CapsNet [23]	43.70	14.97	9.52	15.41	34.21	21.61	27.27	23.78	28.12	6.12
	H-CapsNet [13]	33.85	13.73	11.96	16.13	35.15	20.60	31.60	24.62	7.66	2.54
	HD-CapsNet [19]	53.34	19.52	14.05	21.26	41.86	26.73	35.69	29.73	29.15	8.13
	HT-CapsNet	67.30	41.24	32.52	42.95	72.04	46.75	49.92	48.02	75.15	28.08
	HT-CapsNet†	57.34	31.42	22.75	32.18	65.99	42.82	45.72	43.99	65.14	20.07
	HT-CapsNet‡	52.42	26.21	17.42	26.17	62.02	40.25	42.98	41.35	60.14	15.07

† Denotes the HT-CapsNet without the taxonomy guided routing (taxonomy-based masking) in the routing process.

‡ Denotes the HT-CapsNet without the hierarchcial agreement between the capsules in different levels of the taxonomy.

while preserving the overall hierarchical structure. Notably, related categories (e.g., sky, water, and road under transport) show appropriate proximity while maintaining distinct clusters. Third, at

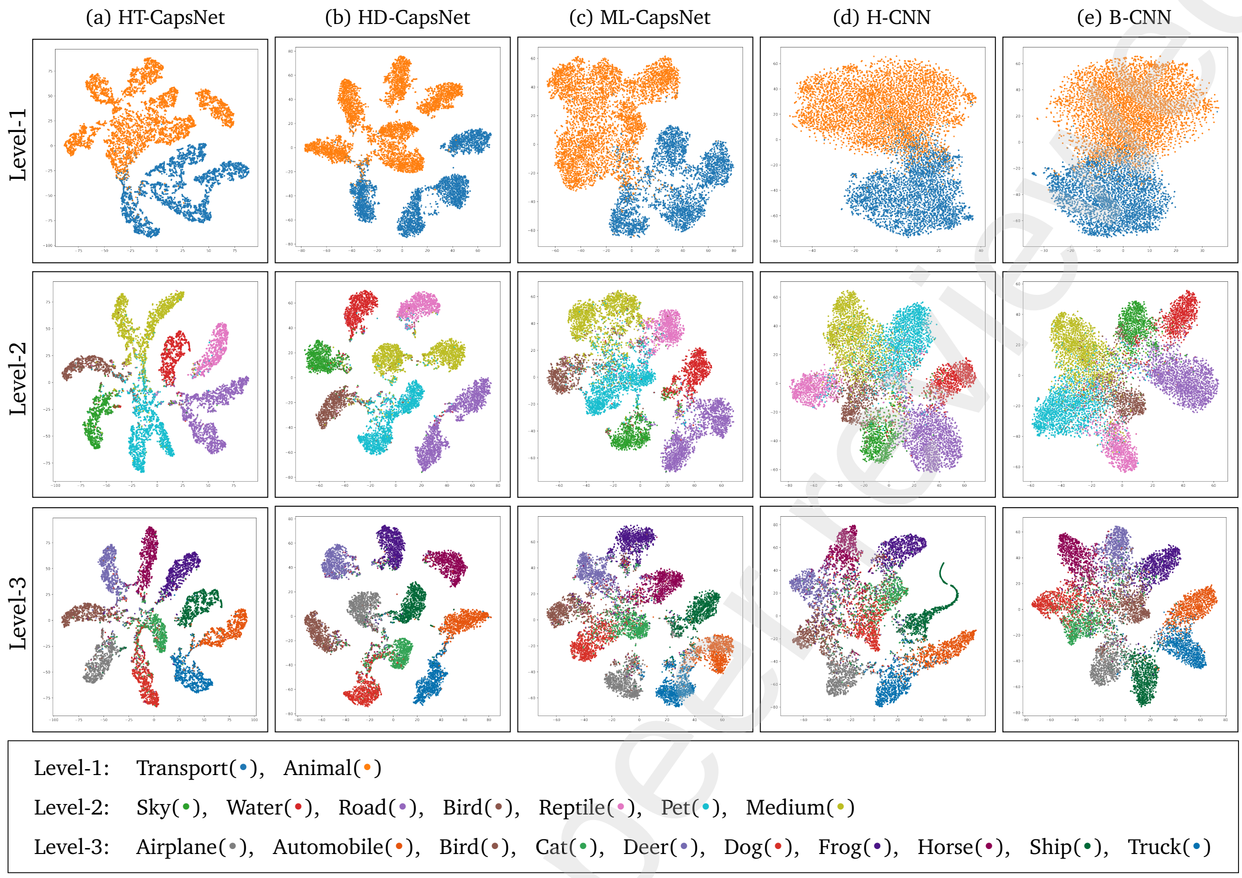

Figure 3: t-SNE visualization of learned feature representations by HT-CapsNet and baseline methods across hierarchical levels. Each point represents a sample, colored according to its ground truth label at the corresponding level. Level-1 shows the coarse binary separation between transport and animal categories. Level-2 demonstrates mid-level categorization into seven subcategories. Level-3 displays fine-grained separation into ten specific classes. HT-CapsNet achieves clearer class separation and more coherent cluster formation compared to baseline methods, particularly at finer levels, while maintaining hierarchical relationships between levels.

the finest level (Level-3), HT-CapsNet demonstrates superior preservation of hierarchical relationships while maintaining fine-grained discrimination. The visualization shows clear sub-clusters that respect parent-child relationships, with smoother transitions between related categories compared to baseline methods.

Furthermore, across all levels, HT-CapsNet produces more compact and well-separated clusters compared to baseline models, where clusters often show significant overlap or diffuse boundaries.

This visual evidence aligns with the quantitative improvements in classification metrics. The progressive refinement from Level-1 to Level-3 in HT-CapsNet's visualizations shows clear hierarchical structure preservation, with child categories properly nested within their parent category spaces. This visual coherence is less evident in baseline models, particularly in H-CNN and B-CNN, where hierarchical relationships become increasingly ambiguous at deeper levels. Notably, all capsule-based models (HT-CapsNet, HD-CapsNet, and ML-CapsNet) demonstrate superior cluster separation and hierarchical preservation compared to convolution-based approaches (H-CNN and B-CNN), which aligns with their better quantitative performance across all datasets. These visualization patterns support the quantitative results and provide intuitive evidence of HT-CapsNet's improved capability in learning hierarchically-aware representations while maintaining discriminative power at all levels of granularity.

4.4. Ablation Study

To validate the effectiveness of each key component in HT-CapsNet, we conducted extensive ablation studies by removing or modifying critical elements of our methods and design choices. The studies focus on three main aspects: the impact of taxonomy-guided routing, the effect of hierarchical agreement mechanisms, and the influence of hierarchical depth on model performance. All ablation experiments were performed across all datasets, with detailed results reported in Tables 1, 2, and 3.

We first examined the effect of removing the taxonomy-guided routing mechanism (HT-CapsNet†), which eliminates the taxonomic mask $m^l_{i,k}$ from the routing process while maintaining other components. This modification results in standard routing coefficients that don't explicitly consider class hierarchy relationships. The performance degradation is notable across all datasets, with the impact becoming more pronounced in complex hierarchical scenarios. On fine-grained datasets like CUB-200-2011 and Stanford Cars, the absence of taxonomy guidance leads to substantial drops in hierarchical metrics, particularly in consistency scores. This degradation pattern suggests that taxonomic information plays a crucial role in guiding the routing process toward hierarchically meaningful representations.

Similarly, we conducted an ablation study to evaluate the impact of the hierarchical agreement mechanism in HT-CapsNet. The modified model (HT-CapsNet‡) removes the hierarchical agreement component while all the other components remain intact. This modification removes the agreement computation between consecutive levels ($h^l_{i,k}$) that is defined in Algorithm 1, which normally ensures that routing decisions at each level are influenced by the predictions from previous levels. The ablation of this mechanism leads to significant performance degradation across all datasets, with the most pronounced effects seen in hierarchical consistency scores and exact match rates. The impact is particularly evident in complex datasets like CUB-200-2011 and Stanford Cars, where the model's ability to maintain coherent predictions across different levels is notably diminished. This degradation pattern suggests that the hierarchical agreement mechanism plays a crucial role in ensuring that the learned representations at each level are properly influenced by and consistent with the predictions from previous levels.

To understand how the number of hierarchical levels affects model performance, we conducted experiments varying the hierarchy depth from 2 to 5 levels on the Marine-tree dataset as a representative example. The results in Table 4 demonstrate the impact of hierarchical depth on classification accuracy at different levels. The results reveal that increasing the number of hierarchical levels consis-

Table 4: Analysis of hierarchical depth impact on model performance using the Marine-tree dataset. Results show how classification accuracy at each level (l=1 to l=5) changes as more hierarchical levels are incorporated into the model. The progressive improvement in accuracy across all levels demonstrates the benefits of deeper hierarchical structures in capturing multi-level semantic relationships. The absolute best results, achieved with all five levels, are marked in bold, highlighting the advantage of utilizing complete hierarchical information.

# Hierarchical Levels	Accuracy per level (%)				
	l=1	l=2	l=3	l=4	l=5
2	89.89	78.59	–	–	–
3	90.76	81.19	61.12	–	–
4	90.97	81.60	61.70	56.75	–
5	**91.21**	**81.90**	**62.02**	**57.12**	**55.05**

tently improves performance across all existing levels, with optimal results achieved using all five levels. This pattern suggests that deeper hierarchical structures provide valuable contextual information that benefits the entire classification process. The improvements are more pronounced at intermediate levels compared to the top level, indicating that additional hierarchical context helps refine mid-level representations without compromising high-level classification performance. Moreover, even as deeper levels are added, the model maintains robust performance on higher levels, demonstrating that increased architectural complexity does not compromise performance on coarser classifications.

These ablation studies validate our architectural choices and demonstrate that both taxonomy-guided routing and hierarchical agreement mechanisms are essential for effective hierarchical learning. The results also support our decision to utilise full hierarchical structures when available, as deeper hierarchies provide valuable contextual information that benefits the entire classification process. Moreover, the studies highlight the complementary nature of our key components, showing that their combination produces synergistic effects that enable more effective hierarchical representation learning.

4.5. Computational Performance Analysis

To assess the computational overhead introduced by our taxonomy-aware routing mechanism, we conducted extensive performance benchmarking by comparing HT-CapsNet with standard dynamic routing [20]. Table 5 presents a comprehensive analysis across different datasets and routing iterations, measuring floating point operations (FLOP), training time metrics, and inference performance. The analysis reveals that the introduction of taxonomy-aware routing introduces a variable computational overhead depending on the dataset complexity. For simpler datasets like Fashion-MNIST, the increase in FLOPs is minimal, at approximately 0.12%. However, for complex fine-grained datasets such as CUB-200-2011, the increase reaches 38.32%. This scaling pattern directly correlates with the complexity of taxonomic relationships present in these datasets, reflecting the additional computational work required to maintain hierarchical consistency during routing.

Training efficiency analysis shows that the average epoch time experiences moderate increases compared to standard routing, ranging from 3% to 21% depending on the dataset size and complexity. The larger datasets, particularly those with complex hierarchical structures, show higher computational overhead during training. How-

Table 5: Computational performance comparing proposed taxonomy-aware routing with standard dynamic routing [20] across different datasets and routing iterations. Metrics include Floating Point Operations (FLOPs), training time, inference latency, and throughput. Arrows (↑/↓) indicate performance changes (increase/decrease) relative to standard routing.

Dataset	Routing Iterations	FLOPs	Avg Epoch Time (s)	Avg Sample Time (mS)	Avg Latency (mS)	Throughput (samples/s)
Fashion-MNIST	2	241.96 M ↑ 0.12%	9.53 ↑ 4.26%	4.83 ↑ 5.53%	2.79 ↑ 2.00%	358.20 ↓ 1.96%
	3	242.1 M ↑ 0.12%	9.52 ↑ 2.36%	4.82 ↑ 2.95%	2.83 ↓ 0.84%	353.65 ↑ 0.85%
	4	242.24 M ↑ 0.12%	9.58 ↑ 4.63%	4.87 ↑ 5.31%	2.81 ↑ 1.86%	355.42 ↓ 1.82%
	5	242.39 M ↑ 0.12%	9.66 ↑ 5.89%	4.89 ↑ 7.53%	2.78 ↑ 4.77%	359.74 ↓ 4.55%
Marine-tree	2	922.81 M ↑ 6.97%	37.07 ↑ 5.88%	6.91 ↑ 17.59%	3.50 ↑ 12.66%	285.93 ↓ 11.23%
	3	925.81 M ↑ 6.98%	37.07 ↑ 10.53%	6.91 ↑ 29.73%	3.50 ↑ 21.31%	285.93 ↓ 17.57%
	4	928.8 M ↑ 6.99%	38.75 ↑ 6.87%	8.40 ↑ 15.22%	4.15 ↑ 6.37%	241.07 ↓ 5.99%
	5	931.79 M ↑ 7.00%	39.91 ↑ 6.94%	9.14 ↑ 13.90%	4.35 ↑ 7.84%	229.96 ↓ 7.27%
CIFAR-10	2	242.15 M ↑ 0.14%	12.34 ↑ 3.40%	4.81 ↑ 3.26%	2.46 ↑ 3.76%	380.12 ↓ 4.07%
	3	242.3 M ↑ 0.14%	12.62 ↑ 0.61%	4.79 ↑ 5.45%	2.66 ↑ 4.45%	375.59 ↓ 4.26%
	4	242.44 M ↑ 0.14%	12.47 ↑ 2.78%	4.83 ↑ 5.47%	2.78 ↑ 3.87%	359.15 ↓ 3.73%
	5	242.59 M ↑ 0.14%	12.49 ↑ 3.97%	4.88 ↑ 5.88%	3.12 ↑ 2.92%	320.15 ↓ 2.21%
CIFAR-100	2	257.53 M ↑ 3.10%	12.58 ↑ 4.14%	4.93 ↑ 5.81%	2.96 ↑ 5.46%	349.95 ↓ 4.11%
	3	258.35 M ↑ 3.11%	12.58 ↑ 5.02%	4.92 ↑ 8.11%	3.09 ↑ 2.78%	337.48 ↓ 4.99%
	4	259.18 M ↑ 3.11%	12.72 ↑ 5.08%	5.00 ↑ 8.12%	3.16 ↑ 1.94%	323.89 ↓ 1.19%
	5	260 M ↑ 3.11%	13.09 ↑ 2.45%	5.07 ↑ 7.52%	3.19 ↑ 2.29%	286.20 ↓ 7.03%
CUB-200-2011	2	1.15 G ↑ 38.32%	31.38 ↑ 21.20%	9.90 ↑ 34.84%	5.07 ↑ 163.50%	197.30 ↓ 15.68%
	3	1.16 G ↑ 37.95%	34.13 ↑ 15.47%	11.30 ↑ 29.72%	5.43 ↑ 21.66%	184.30 ↓ 17.80%
	4	1.17 G ↑ 37.59%	36.06 ↑ 17.10%	12.64 ↑ 26.57%	5.90 ↑ 19.97%	169.60 ↓ 16.64%
	5	1.18 G ↑ 37.24%	38.45 ↑ 31.86%	14.02 ↑ 24.23%	6.48 ↑ 18.21%	154.29 ↓ 15.40%
Stanford Cars	2	1.08 G ↑ 32.23%	55.25 ↑ 10.11%	8.79 ↑ 34.65%	4.39 ↑ 18.31%	227.56 ↓ 15.48%
	3	1.09 G ↑ 32.05%	59.11 ↑ 7.28%	9.70 ↑ 35.05%	4.56 ↑ 24.56%	219.29 ↓ 19.72%
	4	1.09 G ↑ 31.87%	57.77 ↑ 12.83%	10.56 ↑ 28.46%	4.92 ↑ 21.77%	203.28 ↓ 17.88%
	5	1.1 G ↑ 31.70%	61.79 ↑ 9.91%	11.42 ↑ 26.73%	5.25 ↑ 21.03%	190.31 ↓ 17.38%

- All computational measurements were performed on a single NVIDIA A100 GPU with 40GB memory.
- Training metrics (average epoch time and sample time) were calculated using 50 batches per epoch with batch size of 32. Inference metrics (latency and throughput) were measured using 2,000 randomly sampled test examples.

ever, this additional computational cost is justified by the significant improvements in classification performance, especially in scenarios involving complex hierarchical relationships. The training time scaling remains predictable and manageable across different dataset sizes. Examining inference performance metrics reveals interesting patterns in model deployment characteristics. While HT-CapsNet shows slightly increased latency across all configurations, the impact on throughput remains within acceptable bounds. For example, with 5 routing iterations on CUB-200-2011, the most complex dataset in our experiments, the throughput reduction is only 15.40% compared to standard routing. This relatively modest decrease in inference speed suggests that our method maintains practical utility in real-world applications despite its increased sophistication.

The relationship between computational requirements and routing iterations demonstrates efficient algorithmic scaling. Our measurements indicate that the computational overhead scales approximately linearly with the number of routing iterations, suggesting good algorithmic efficiency. More importantly, the relative performance impact remains stable across different iteration counts, indicating robust scaling behavior that maintains predictable performance characteristics as the routing complexity increases. Datasets with complex hierarchical structures, particularly CUB-200-2011 and Stanford Cars, show more pronounced computational requirements, with FLOPs increasing by $31-38\%$. This additional computation directly contributes to the model's superior hierarchical learning capabilities, as evidenced by the performance improvements shown in Tables 1, 2, and 3. The relationship between computational cost and performance improvement appears to be particularly favorable for these complex tasks, where the benefits of improved hierarchical learning outweigh the increased computational demands.

The computational analysis demonstrates that while HT-CapsNet introduces additional computational overhead compared to standard routing approaches, this cost scales predictably with problem complexity and remains reasonable relative to the achieved performance improvements. These findings indicate that the trade-off between computational cost and classification performance is particularly favorable for complex hierarchical tasks, where the benefits of improved hierarchical learning justify the modest increase in computational requirements.

5. Discussion and Limitations

While HT-CapsNet demonstrates significant improvements in hierarchical multi-label classification, several important considerations and limitations warrant discussion. Our analysis reveals both the strengths of our approach and areas that merit further investigation. The superior performance of HT-CapsNet, particularly on fine-grained datasets, validates our core hypothesis that explicitly incorporating taxonomic information into the routing mechanism enhances hierarchical representation learning. The consistent improvements across both coarse and fine-grained levels suggest that our approach successfully balances high-level category discrimination with fine-grained feature detection. This is particularly evident in the t-SNE visualizations, where HT-CapsNet maintains clear cluster separation while preserving hierarchical relationships.

Nonetheless, it is important to recognize several challenges associated with our taxonomy-aware routing mechanism. To begin with, the computational complexity escalates as the hierarchy's depth and breadth increase. Although this added complexity is warranted due to the performance enhancements, it might pose difficulties for hierarchies that are excessively deep or for applications requiring real-time processing. Future research could investigate optimization methods or pruning approaches to alleviate this computational load while preserving performance. Our existing implementation necessitates a predetermined, static taxonomy framework. Although this works well for numerous practical applications with clearly established class hierarchies, it might restrict adaptability in situations where taxonomic connections are ambiguous or changing. Expanding the model to accommodate dynamic or probabilistic taxonomies could enhance its range of use. Additionally, HT-CapsNet demonstrates strong performance across a variety of datasets, its advantages are most pronounced in complex, fine-grained classification tasks. For simpler hierarchical structures, the additional complexity of our approach may not always justify the marginal improvements over simpler methods. This suggests the need for adaptive mechanisms that can adjust the routing complexity based on the task requirements.

The current model also assumes clean, well-defined hierarchical relationships. In practice, some classes might have ambiguous relationships or belong to multiple parent categories. Future work could explore modifications to handle such overlapping hierarchies or direct acyclic graph based taxonomic relationships. Additionally, investigating ways to automatically learn or refine taxonomic structures from data could make the approach more adaptable to scenarios where expert-defined hierarchies may be suboptimal. Furthermore, a significant constraint lies in the requirement for carefully tuned hyperparameters, particularly in the routing mechanism. Although our empirical studies provide guidance for parameter selection, developing more robust, self-adaptive parameter tuning strategies could improve the model's usability across different domains.

Despite these constraints, our findings indicate that HT-CapsNet marks a considerable advancement in hierarchical multi-label classification. The model's ability to maintain hierarchical consistency while achieving top-tier performance suggests promising directions for future research in hierarchical deep learning architectures. Looking ahead, several promising research directions emerge. Investigating the integration of self-supervised learning techniques could reduce the dependence on large labeled datasets. These considerations highlight both the significant potential and the remaining challenges in hierarchical deep learning, pointing toward exciting opportunities for future research and development in this field.

6. Conclusion

In this paper, we introduced HT-CapsNet, a novel hierarchical taxonomy-aware capsule network architecture that effectively addresses the challenges of hierarchical multi-label classification. Our approach uniquely integrates taxonomic relationships into the capsule routing mechanism through a taxonomy-guided routing algorithm, enabling more effective learning of hierarchical features while maintaining consistency across classification levels. Comprehensive experiments across diverse datasets demonstrate that HT-CapsNet consistently outperforms existing approaches, with particularly significant improvements in complex, fine-grained classification tasks. The empirical results validate that both taxonomy-guided routing and hierarchical agreement mechanisms contribute significantly to the model's performance, while visualization analysis reveals that HT-CapsNet learns more discriminative and hierarchically consistent representations compared to existing approaches. Beyond the immediate technical contributions, this work opens several promising directions for future research in hierarchical deep learning, suggesting potential applications in domains where hierarchical relationships play a crucial role.

References

[1] C. N. Silla, A. A. Freitas, A survey of hierarchical classification across different application domains, Data Min Knowl Disc 22 (1) (2011) 31–72. doi:10.1007/s10618-010-0175-9.

[2] Z. Yuan, H. Liu, H. Zhou, D. Zhang, X. Zhang, H. Wang, H. Xiong, Self-Paced Unified Representation Learning for Hierarchical Multi-Label Classification, Proceedings of the AAAI Conference on Artificial Intelligence 38 (15) (2024) 16623–16632. doi:10.1609/aaai.v38i15.29601.

[3] M. Han, H. Wu, Z. Chen, M. Li, X. Zhang, A survey of multi-label classification based on supervised and semi-supervised learning, Int. J. Mach. Learn. & Cyber. 14 (3) (2023) 697–724. doi:10.1007/s13042-022-01658-9.

[4] J. Kim, B. J. Choi, FedTH : Tree-based Hierarchical Image Classification in Federated Learning, in: Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022), 2022, pp. 1–7.

[5] J. Zhou, C. Ma, D. Long, G. Xu, N. Ding, H. Zhang, P. Xie, G. Liu, Hierarchy-Aware Global Model for Hierarchical Text Classification, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 1106–1117. doi:10.18653/v1/2020.acl-main.104.

[6] R. E. Armah-Sekum, S. Szedmak, J. Rousu, Protein function prediction through multi-view multi-label latent tensor reconstruction, BMC Bioinformatics 25 (1) (2024) 174. doi:10.1186/s12859-024-05789-4.

[7] C. Feng, I. Patras, MaskCon: Masked Contrastive Learning for Coarse-Labelled Dataset, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Vancouver, BC, Canada, 2023, pp. 19913–19922. doi:10.1109/CVPR52729.2023.01907.

[8] X. Guo, X. Liu, Z. Ren, S. Grosz, I. Masi, X. Liu, Hierarchical Fine-Grained Image Forgery Detection and Localization, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Vancouver, BC, Canada, 2023, pp. 3155–3165. doi:10.1109/CVPR52729.2023.00308.

[9] Z. Xu, X. Yue, Y. Lv, W. Liu, Z. Li, Trusted Fine-Grained Image Classification through Hierarchical Evidence Fusion, Proceedings of the AAAI Conference on Artificial Intelligence 37 (9) (2023) 10657–10665. doi:10.1609/aaai.v37i9.26265.

[10] Z. Lin, J. Jia, F. Huang, W. Gao, A coarse-to-fine capsule network for fine-grained image categorization, Neurocomputing 456 (2021) 200–219. doi:10.1016/j.neucom.2021.05.032.

[11] X. Huo, G. Sun, S. Tian, Y. Wang, L. Yu, J. Long, W. Zhang, A. Li, HiFuse: Hierarchical multi-scale feature fusion network for medical image classification, Biomedical Signal Processing and Control 87 (2024) 105534. doi:10.1016/j.bspc.2023.105534.

[12] R. Wang, C. Zou, W. Zhang, Z. Zhu, L. Jing, Consistency-aware Feature Learning for Hierarchical Fine-grained Visual Classification, in: Proceedings of the 31st ACM International Conference on Multimedia, MM '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 2326–2334. doi:10.1145/3581783.3612234.

[13] K. T. Noor, A. Robles-Kelly, H-CapsNet: A capsule network for hierarchical image classification, Pattern Recognition 147 (2024) 110135. doi:10.1016/j.patcog.2023.110135.

[14] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, Y. Yu, HD-CNN: Hierarchical deep convolutional neural networks for large scale visual recognition, in: Proceedings of the IEEE International Conference on Com-

puter Vision, 2015, pp. 2740–2748.

[15] D. Roy, P. Panda, K. Roy, Tree-CNN: A hierarchical Deep Convolutional Neural Network for incremental learning, Neural Networks 121 (2020) 148–160. doi:10.1016/j.neunet.2019.09.010.

[16] X. Zhu, M. Bain, B-CNN: Branch convolutional neural network for hierarchical classification, arXiv preprint arXiv:1709.09890 (2017). arXiv:1709.09890.

[17] F. M. Miranda, N. Köhnecke, B. Y. Renard, HiClass: A Python library for local hierarchical classification compatible with scikit-learn, Journal of Machine Learning Research 24 (29) (2022) 1–17. arXiv:2112.06560, doi:10.48550/arXiv.2112.06560.

[18] Y. Huo, Y. Lu, Y. Niu, Z. Lu, J.-R. Wen, Coarse-to-Fine Grained Classification, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1033–1036. doi:10.1145/3331184.3331336.

[19] K. T. Noor, A. Robles-Kelly, L. Y. Zhang, M. R. Bouadjenek, W. Luo, A consistency-aware deep capsule network for hierarchical multi-label image classification, Neurocomputing 604 (2024) 128376. doi:10.1016/j.neucom.2024.128376.

[20] S. Sabour, N. Frosst, G. E. Hinton, Dynamic Routing Between Capsules, in: Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017, pp. 1–11. doi:10.48550/arXiv.1710.09829.

[21] A. Pajankar, A. Joshi, Convolutional Neural Networks, in: A. Pajankar, A. Joshi (Eds.), Hands-on Machine Learning with Python: Implement Neural Network Solutions with Scikit-learn and PyTorch, Apress, Berkeley, CA, 2022, pp. 261–284. doi:10.1007/978-1-4842-7921-2_14.

[22] K. T. Noor, A. Robles-Kelly, B. Kusy, A Capsule Network for Hierarchical Multi-label Image Classification, in: A. Krzyzak, C. Y. Suen, A. Torsello, N. Nobile (Eds.), Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2022, pp. 163–172. doi:10.1007/978-3-031-23028-8_17.

[23] K. T. Noor, A. Robles-Kelly, L. Y. Zhang, M. R. Bouadjenek, A Bottom-Up Capsule Network for Hierarchical Image Classification, in: 2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2023, pp. 325–331. doi:10.1109/DICTA60407.2023.00052.

[24] S. Zheng, S. Chen, Q. Jin, Few-Shot Action Recognition with Hierarchical Matching and Contrastive Learning, in: S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, T. Hassner (Eds.), Computer Vision – ECCV 2022, Lecture Notes in Computer Science, Springer Nature Switzerland, Cham, 2022, pp. 297–313. doi:10.1007/978-3-031-19772-7_18.

[25] Y. Seo, K.-s. Shin, Hierarchical convolutional neural networks for fashion image classification, Expert Systems with Applications 116 (2019) 328–339. doi:10.1016/j.eswa.2018.09.022.

[26] W. Qi, C. Chelmis, Hybrid Loss for Hierarchical Multi–label Classification Network, in: 2023 IEEE International Conference on Big Data (BigData), 2023, pp. 819–828. doi:10.1109/BigData59044.2023.10386341.

[27] T. Boone-Sifuentes, M. R. Bouadjenek, I. Razzak, H. Hacid, A. Nazari, A Mask-based Output Layer for Multi-level Hierarchical Classification, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 3833–3837. doi:10.1145/3511808.3557534.

[28] Y. Liu, L. Zhou, P. Zhang, X. Bai, L. Gu, X. Yu, J. Zhou, E. R. Hancock, Where to Focus: Investigating Hierarchical Attention Relationship for Fine-Grained Visual Classification, in: S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, T. Hassner (Eds.), Computer Vision – ECCV 2022, Lecture Notes in Computer Science, Springer Nature Switzerland, Cham, 2022, pp. 57–73. doi:10.1007/978-3-031-20053-3_4.

[29] Y. Xie, C. Yao, M. Gong, C. Chen, A. K. Qin, Graph convolutional networks with multi-level coarsening for graph classification, Knowledge-Based Systems 194 (2020) 105578. doi:10.1016/j.knosys.2020.105578.

[30] D. Fu, H. Zhong, X. Zhang, Q. Zhou, C. Wan, B. Wu, Y. Hu, Graph relationship-driven label coded mapping and compensation for multi-label textile fiber recognition, Engineering Applications of Artificial Intelligence 133 (2024) 108484. doi:10.1016/j.engappai.2024.108484.

[31] J. Lanchantin, T. Wang, V. Ordonez, Y. Qi, General Multi-label Image Classification with Transformers, in: 2021

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Nashville, TN, USA, 2021, pp. 16473–16483. doi:10.1109/CVPR46437.2021.01621.

[32] J. Wu, H. Yang, T. Gan, N. Ding, F. Jiang, L. Nie, CHMATCH: Contrastive Hierarchical Matching and Robust Adaptive Threshold Boosted Semi-Supervised Learning, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Vancouver, BC, Canada, 2023, pp. 15762–15772. doi:10.1109/CVPR52729.2023.01513.

[33] J. Chen, P. Wang, J. Liu, Y. Qian, Label Relation Graphs Enhanced Hierarchical Residual Network for Hierarchical Multi-Granularity Classification, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, New Orleans, LA, USA, 2022, pp. 4848–4857. doi:10.1109/CVPR52688.2022.00481.

[34] G. E. Hinton, S. Sabour, N. Frosst, Matrix capsules with EM routing, in: International Conference on Learning Representations, OpenReview.net, 2018, pp. 1–15.

[35] M. Kwabena Patrick, A. Felix Adekoya, A. Abra Mighty, B. Y. Edward, Capsule Networks – A survey, Journal of King Saud University - Computer and Information Sciences 34 (1) (2022) 1295–1310. doi:10.1016/j.jksuci.2019.09.014.

[36] T. Hahn, M. Pyeon, G. Kim, Self-routing capsule networks, Advances in neural information processing systems 32 (2019).

[37] J. Gugglberger, D. Peer, A. Rodríguez-Sánchez, Training Deep Capsule Networks with Residual Connections, in: I. Farkaš, P. Masulli, S. Otte, S. Wermter (Eds.), Artificial Neural Networks and Machine Learning – ICANN 2021, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2021, pp. 541–552. doi:10.1007/978-3-030-86362-3_44.

[38] J. Choi, H. Seo, S. Im, M. Kang, Attention Routing Between Capsules, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), IEEE, Seoul, Korea (South), 2019, pp. 1981–1989. doi:10.1109/ICCVW.2019.00247.

[39] G. Sun, S. Ding, T. Sun, C. Zhang, SA-CapsGAN: Using Capsule Networks with embedded self-attention for Generative Adversarial Network, Neurocomputing 423 (2021) 399–406. doi:10.1016/j.neucom.2020.10.092.

[40] J. Rajasegaran, V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, R. Rodrigo, DeepCaps: Going Deeper With Capsule Networks, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA, 2019, pp. 10717–10725. doi:10.1109/CVPR.2019.01098.

[41] A. Byerly, T. Kalganova, I. Dear, No routing needed between capsules, Neurocomputing 463 (2021) 545–553. doi:10.1016/j.neucom.2021.08.064.

[42] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, A. Mohammadi, COVID-CAPS: A capsule network-based framework for identification of COVID-19 cases from X-ray images, Pattern Recognition Letters 138 (2020) 638–643. doi:10.1016/j.patrec.2020.09.010.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is All you Need, in: Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017, pp. 1–11.

[44] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer Normalization (Jul. 2016). arXiv:1607.06450, doi:10.48550/arXiv.1607.06450.

[45] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, TensorFlow: A System for Large-Scale Machine Learning, in: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016, pp. 265–283.

[46] H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms (Sep. 2017). arXiv:1708.07747, doi:10.48550/arXiv.1708.07747.

[47] T. Boone-Sifuentes, A. Nazari, I. Razzak, M. R. Bouadjenek, A. Robles-Kelly, D. Ierodiaconou, E. S. Oh, Marinetree: A Large-scale Marine Organisms Dataset for Hierarchical Image Classification, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 3838–3842. doi:10.1145/3511808.3557634.

[48] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Tech. rep., Toronto, ON, Canada (2009).

[49] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200-2011 Dataset, https://resolver.caltech.edu/CaltechAUTHORS:20111026-120541847 (Jul. 2011).

[50] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D Object Representations for Fine-Grained Categorization, in: 2013 IEEE International Conference on Computer Vision Workshops, IEEE, Sydney, Australia, 2013, pp. 554–561. doi:10.1109/ICCVW.2013.77.

[51] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition (Apr. 2015). arXiv:1409.1556, doi:10.48550/arXiv.1409.1556.

[52] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[53] M. Tan, Q. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, in: Proceedings of the 36th International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.

[54] B. Kolisnik, I. Hogan, F. Zulkernine, Condition-CNN: A hierarchical multi-label fashion image classification model, Expert Systems with Applications 182 (2021) 115195. doi:10.1016/j.eswa.2021.115195.

[55] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, Mixup: Beyond Empirical Risk Minimization, in: 6th International Conference on Learning Representations, 2018, pp. 1–13.