Neurocomputing 604 (2024) 128376

Contents lists available at ScienceDirect

Neurocomputing

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

A consistency-aware deep capsule network for hierarchical multi-label image classification

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

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

ARTICLE INFO

Communicated by B. Li

Keywords:
Image classification
Hierarchical multi-label classification
Capsule network

ABSTRACT

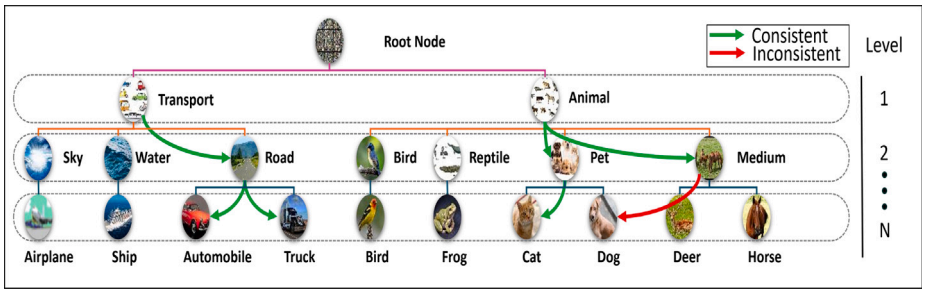
Hierarchical classification is a significant challenge in computer vision due to the logical order and interconnectedness of multiple labels. This paper presents HD-CapsNet, a novel neural network architecture based on deep capsule networks, specifically designed for hierarchical multi-label classification(HMC). By incorporating a tree-like hierarchical structure, HD-CapsNet is designed to leverage the inherent ontological order within the hierarchical label tree, thereby ensuring classification consistency across different levels. Additionally, we introduce a specialized loss function that promotes accurate hierarchical relationships while penalizing inconsistencies. This not only enhances classification performance but also strengthens the network's robustness. We rigorously evaluate HD-CapsNet's efficacy by benchmarking it against existing HMC methods across six diverse datasets: Fashion-MNIST, Marine-Tree, CIFAR-10, CIFAR-100, Caltech-UCSD Birds-200-2011, and Stanford Cars. Our results conclusively demonstrate that HD-CapsNet excels in learning hierarchical relationships and significantly outperforms the competition in various image classification tasks. Our implementation is available at https://github.com/tasrif-khondaker/HD-CapsNet.

1. Introduction

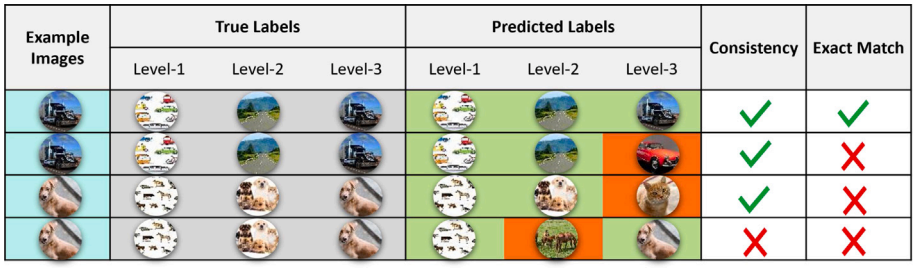
Image classification is notably a challenging task in computer vision and machine learning due to the high variability and complexity of natural images. In most cases, image classifiers are designed to classify images to address image categories based on the features of the objects in the images. However, in many real-world applications, images can contain multiple objects, and each object can belong to multiple classes. For example, an image of a dog can be classified as a "dog", "pet", "animal", etc. based on its features. To address these complexities, there is increasing interest in hierarchical multi-label image classification. The goal is to model the hierarchical relationships between visual concepts, so that an image classifier can output multiple labels according to a taxonomy tree. For hierarchical classification methods, a multi-label classification approach is commonly used to model the taxonomy. The main advantage over single-label classification is the ability to classify images at multiple granularities and model relationships between labels at different levels. This comes at the cost of more complex models and inference techniques. But hierarchical multi-label classification (HMC) unlocks capabilities like filtering image search results by high-level categories or retrieving images according to fine-grained attributes. Hierarchical image classification has been adopted in many applications such as object recognition [1], medical image analysis [2], fine-grained image classification [3], and scene understanding [4]. Generally, in the hierarchical classification approach, objects are classified following a coarse-to-fine paradigm [5], i.e. images are first classified into coarse categories (such as animals, transport, etc.), and then further classified into finer categories within each coarse category (such as dog, cat, etc. within animals). Which requires a multi-label classification approach that can handle multiple labels following a predefined taxonomy for a single instance. To demonstrate and evaluate the advantage of hierarchical classification, we refer to Fig. 1, which shows (a) an example of a hierarchical label tree for the CIFAR-10 dataset, and (b) some prediction labels for the images following the label tree. Note that, the examples in Fig. 1 indicate that hierarchical relationships between the classes in the label tree follow an ontological order. This information can be used to guide classification models to learn the hierarchical relationships between the classes in the label tree, which can improve classification performance. Hierarchical image classification has several

* Corresponding author.
E-mail addresses: k.noor@research.deakin.edu.au (K.T. Noor), antonio.robles-kelly@deakin.edu.au (A. Robles-Kelly), leo.zhang@deakin.edu.au, leo.zhang@griffith.edu.au (L.Y. Zhang), reda.bouadjenek@deakin.edu.au (M.R. Bouadjenek), wei.luo@deakin.edu.au (W. Luo).

https://doi.org/10.1016/j.neucom.2024.128376
Received 3 November 2023; Received in revised form 1 June 2024; Accepted 10 August 2024
Available online 14 August 2024



(a)



(b)

Fig. 1. (a) Example of an ontology-guided hierarchical label tree for the CIFAR-10 dataset. The dataset has 3 hierarchical levels with 2, 7, and 10 classes at the coarse, medium, and fine levels, respectively. (b) Classification outcomes of the HMC approach based on the hierarchical relationships of different levels of sample images from the CIFAR-10 dataset. Note that hierarchical consistency requires predictions to follow the hierarchical branch, whereas exact match requires the predictions to align with the ground truth.

advantages over flat (non-hierarchical) classification such as it can handle large and complex datasets with a large number of classes by breaking them down into subcategories [6], improve the accuracy of classification by adjusting the decision boundaries and reducing the computational complexity by making use of the taxonomic structure in the label tree [7], and can provide a better understanding of the relationships between different image features.

In recent years, deep learning-based methods, especially convolutional neural networks (CNNs), have shown remarkable performance in solving hierarchical image classification problems. The convolution algorithm in the CNN extracts lower-level features in the shallow layers and higher-level features in the deeper layers [8]. Which allows the CNN to learn hierarchical representations of visual features from raw pixel data in different layers. However, CNNs have some limitations in capturing the hierarchical structure of objects in images, which can lead to misclassification and poor generalization performance. The attention regions of CNN classifiers can be derived as an attention heatmap in the middle layers of the network. However, Guo et al. in [9] observed that, their consistency is not maintained under various transformations. Despite this, CNNs struggle with certain limitations, such as only relying on object features to classify [10] and lack the ability to understand the relationship between objects [11].

Capsule networks (CapsNet), introduced by Sabour et al. in [10], aim to resolve some of these shortcomings by representing entities as vectors, allowing for more expressive and interpretable models [10]. Each layer in a CapsNet consists of a set of capsules, which are groups of neurons that represent different features in the image. Capsules in lower layers output a vector representing different properties of features, and capsules in higher layers use this information to compute hierarchical relationships between object parts [12] and compute a prediction of the presence of the whole object. This allows CapsNet to learn hierarchical representations of objects more efficiently, which can improve the performance of hierarchical image classification by making use of a hierarchical label tree [13]. Notwithstanding the encouraging outcomes of CapsNet, the model can still be refined to improve their ability to learn hierarchical representations of objects in images making use of the relationship between different layers in the network. However, current capsule network implementations still have limitations in effectively learning hierarchical representations of object features for classification problems. Simply stacking more capsule layers results in routing challenges [14] and diminished spatial encoding. This even leads capsule networks to lose their natural advantages to connect the hierarchical relationships between the feature properties. Modified routing algorithms and network architectures have been introduced to overcome this limitation by better modelling the relationships [15] and reducing the computational complexity [16]. The models in [13,17] employ capsule networks to learn the hierarchical structure of the dataset by connecting multiple capsule layers in a hierarchical manner. Further, the model in [18] uses a capsule network to predict multiple labels along with a generative adversarial network (GAN) to learn the hierarchical structure. However, these approaches do not consider the hierarchical consistency between the capsule layers and linearly increase the computational complexity with the number of capsules in the higher levels.

In this paper, we propose the HD-CapsNet architecture, a hierarchical deep capsule network that can model the hierarchical relationships more effectively following a data hierarchy. Our approach involves using multiple capsule layers in the architecture to represent the hierarchical structure of the dataset. This allows the model to tackle all the levels in the hierarchy at once, resembling a global hierarchical classification approach in [6] which follows a coarse-to-fine paradigm. Furthermore, capsule layers in HD-CapsNet are connected to each other in a hierarchical manner, which allows the model to keep a better balance between the hierarchies. We also propose a modified loss function that enforces hierarchical consistency between the outputs of the network. The loss function penalises inconsistencies in the predicted class probabilities across different levels of the hierarchy, and encourages the network to learn representations that are both discriminative and hierarchical. We evaluate the proposed HD-CapsNet architecture on several widely available benchmark datasets, including Fashion-MNIST, Marine-tree, CIFAR-10, CIFAR-100, CUB-200-2011, and Stanford Cars, and compare it against other deep learning methods. Our experiments show that the proposed network achieves a margin of improvement in model performance on these datasets. In summary, the contributions of this paper are as follows:

- We propose HD-CapsNet, a hierarchical deep capsule network that uses multiple capsule layers to learn the hierarchical relationship between different levels in the label tree and use this information to improve the classification performance.
- We introduce a novel loss function that enforces hierarchical consistency in the capsule network.
- We demonstrate the effectiveness of our approach on several benchmark datasets and show that it outperforms the alternatives.

2. Related work

In recent years, the field of deep learning has seen significant progress in the development of capsule networks, which are a promising alternative to traditional convolutional neural networks (CNN). CNNs are widely used and effective in various image classification tasks, they have performed quite well in Multi-label image classification problems [19]. However, CNNs encounter limitations in hierarchical multi-label classification (HMC) tasks, where understanding the relationships and hierarchical structure among labels is crucial. Capsule networks aim to address some limitations of CNNs in encoding higher-level compositionality of objects and scenes. By segmenting images into groups of capsules that represent visual elements at increasing levels of abstraction through a parse tree, capsule networks can parse images into meaningful hierarchies in a more human-interpretable way. As alternatives to CNNs, capsule networks aim to overcome limitations related to hierarchical reasoning by modelling part-whole and spatial relationships more explicitly. Over the past several years, substantial progress has been made in developing and demonstrating the capabilities of capsule networks across a range of computer vision tasks. While still an emerging approach, capsule networks have shown promising results across tasks like image classification [10], object recognition [20], and image segmentation [21]. As architectural innovations and training techniques continue to improve capsule networks, they may gain adoption as more robust and interpretable alternatives to CNNs for computer vision tasks requiring a structured hierarchical understanding of visual scenes.

The concept of capsules was first introduced by Hinton et al. in [22], and it was further developed by Sabour et al. in [10] with the introduction of the Capsule Network architecture. CapsNet uses capsules to model relationships between the features of an image and to output pose parameters, which can be used to represent the position, orientation, and scale of objects in an image. Dynamic routing between Capsules proposed a method for routing information between capsule layers using agreement, which allows the network to learn the most relevant features for a given input. This is in contrast to traditional CNNs, which output class probabilities and do not explicitly model the relationships between features. Since the introduction of CapsNet, a number of research papers have been published that extend and improve upon the original architecture. Some of the most notable papers include Matrix Capsules with EM Routing [12] which allows the network to learn the relative geometry between different features in an image, introduction of MS-CapsNet in [23] that proposes a multi-scale capsule network to fully encode hierarchical features of images, and DeepCaps proposed in [15] proposes a deep architecture for capsule networks using 3D convolution-based dynamic routing algorithm and a pooling strategy to enhance the existing ones. Moreover, the study in [24] highlights the potential of capsule networks in multi-label image classification tasks. They demonstrated that by replacing the last layer of a pre-trained CNN with a capsule layer, the performance of the network could be substantially improved. This finding suggests that capsule networks, with their unique approach to processing and understanding image data, can significantly augment the performance of existing CNN architectures, leveraging their robust feature detection capabilities.

However, current implementations show that adding more capsule layers in capsule networks presents some challenges needing resolution. Increasing the number of capsule layers results in greater complexity and computational expense for the routing algorithms connecting lower-level capsules to higher-level ones. The dynamic routing process [10] must now route signals across additional capsule layers, expanding the dimensionality of the overall routing issue. Furthermore, propagating signals over many capsule layers can diminish information as it passes through the model, rendering the network more susceptible to overfitting on limited training data. To mitigate these matters, modifications to routing algorithms, network architecture, novel regularization methods, and larger datasets may be imperative to effectively train deep capsule networks. To address these challenges, the work in [25] proposed a new capsule network architecture that uses a modified routing algorithm by introducing a routing weight initialization technique. Further, the work presented in [14] provides experimental evidence that implementing residual connections in deep capsule networks can help reduce the loss of spatial information. Another recent development is the COVID-CAPS architecture proposed in [26], which incorporates an additional capsule layer to increase network depth. COVID-CAPS was shown to outperform alternative approaches on the task of classifying COVID-19 cases. These results demonstrate that capsule networks represent a promising technique for capturing the hierarchical structure of a dataset and modelling the complex relationships between the various features within an image. The increased representation capacity provided by the capsule and routing mechanisms enables these networks to learn part-whole relationships and perform detailed feature extraction. By modelling entities as capsules and actively routing information between them, capsule networks are able to preserve spatial relationships and encode higher-level semantic meaning more effectively than standard convolutional neural networks. Overall, these recent advancements indicate that further exploration of deep capsule networks, leveraging techniques such as residual connections, is a worthwhile direction that may lead to performance improvements on tasks requiring a detailed understanding of spatial, part-whole, and hierarchical relationships within image data.

In the realm of standard multi-label image classification, labels are typically treated as distinct and unrelated entities. Each image can be labelled with multiple descriptors, but these usually do not imply any hierarchical or interdependent relationships. To address this, various methodologies have been developed. The work in [27], explored the combination of multiple classification models to classify multi-label aerial images, while the method in [28] introduced a multi-branch strategy for classifying images into numerous labels. However, these methods often overlook the hierarchical structure inherent in the data. Contrastingly, hierarchical image classification takes a layered approach, dividing the task into several levels of complexity. In this approach, the lower levels of the hierarchy represent low-level features [8], while the higher levels represent more abstract features. For example, in a hierarchical image classification system for animals, the first level of the hierarchy might classify images into broad categories such as “mammals”, “birds”, “reptiles”, and“fish” based on the low-level features, while subsequent levels of the hierarchy might classify images into more specific categories such as “dogs”, “cats”, “parrots”, “turtles”, “snakes”, and “sharks” using more level specific features. Empirical results have shown the efficacy of this approach in handling variations in object appearance and improving the interpretability [6] of the network. Some notable works in this area include CNN based hierarchical image classification in [29], which uses a branching strategy to classify images into multiple labels, and Tree-CNN in [30] that organizes the incrementally available data into feature-driven superclasses and improves upon existing hierarchical CNN models. Later, Kolisnik et al. proposed a hierarchical neural network for image classification in [31] that first learns low-level features common to all levels in the class hierarchy through a common CNN base model and then uses a hierarchical classification approach to classify images into multiple labels employing conditional weight matrix. In comparison with B-CNN, which uses a branch structure to classify images into multiple

labels in [8], the Condition-CNN in [31] uses a more effective hierarchical classification approach to improve the classification performance. Further, the authors in [32] proposed a mask-based output layer for hierarchical image classification that allows the model to learn the relationships between different levels of classes and to make predictions at multiple levels simultaneously.

In recent times, capsule networks have shown great promise in HMC tasks. This has led to an increase in research exploring different architectures and techniques to improve the performance of capsule networks in hierarchical classification tasks. In particular, ML-CapsNet in [13] uses multiple capsule layers with a common primary capsule layer to model the hierarchical structure of the dataset, and BUH-CapsNet in [33] employs a bottom up approach to learn the hierarchical structure of the dataset by connecting multiple capsule layers in a hierarchical manner. Further, the H-CapsNet in [17] employs dedicated capsule networks for each level of the hierarchy to further improve feature vectors with level specific features. Further, the HGT&BC model in [18] is introduced for hierarchical image classification, which uses a hierarchical GAN-tree to extract features and bi-directional capsules for prediction. The iterative modular capsule development approach adopted by these architectures highlights the suitability of capsule networks for learning hierarchical links within image data. Specifically, the gradual growth and integration of tailored capsules enables the extraction of hierarchical patterns from images. These recent advances demonstrate capsule networks' significant potential to drive progress in hierarchical image classification through capsule design and structured hierarchical modelling. Although these approaches have shown promising results in capturing the hierarchical structure of the dataset, they do not consider the hierarchical consistency between the levels. Further, the models in [13,17] linearly increase the number of capsules in each level mimicking the label tree, which can lead to a large number of capsules in the higher levels. This can increase the computational complexity of the model and can lead to overfitting. Overall, capsule networks have shown great promise in improving the performance of deep neural networks for image classification and other tasks. While significant strides have been taken, there are still gaps that need to be filled to fully understand the capabilities and limitations of capsule networks, recent advances in the field suggest that these approaches may become increasingly pivotal to the future development of deep learning.

3. HD-CapsNet

Our proposed model, called the hierarchical deep capsule network (HD-CapsNet), can learn the hierarchical structure of an image based on a hierarchical label tree. The model takes full advantage of the capsule network to learn the hierarchical representation of the dataset by employing the coarse-to-fine paradigm present in the hierarchical label tree. For training and evaluating the proposed classifier considers the entire class hierarchy at once. Hence, it falls under the global hierarchical classifier category [6], and further allows the model to employ hierarchical consistency in the learning process. The overall architecture of the proposed method is shown in Fig. 2(a). Note that the proposed architecture shown in Fig. 2(a) is composed of a feature extraction block, a common primary capsules (P) and N number of secondary capsules (S_i) where i is the hierarchical index and N is the total number hierarchical levels.

The feature extraction block present in our proposed model in Fig. 2(a) is composed of four sub-blocks, where each sub-block contains two convolutional layers followed by a batch normalization layer and lastly a maxpooling layer. The first sub-block is used to extract the features from the input image, and the remaining three sub-blocks are used to extract the features from the output of the previous sub-block. The purpose of the feature extraction block is to extract the features from the input image and to reduce the dimensionality of the extracted features. The convolutional layers in the feature extraction block are used to extract the features from the raw input image, and the max-pooling layers are used to reduce the dimensionality of the extracted features. The batch normalization layers are used to normalize the extracted features in order to improve the stability [34] of the model. The output of the last sub-block, **H** is then reshaped to form the primary capsule layer as shown in Fig. 2(b).

In the primary capsule layer (P), as described in [10], a crucial process occurs where the layer applies a non-linear transformation to the initially generated local feature maps. These feature maps are essentially representations of the extracted features from the input image. The non-linear transformation is a key step that converts these feature maps into a collection of vectors. As shown in Fig. 2(b), this transformation is achieved by first reshaping, and then flattening and squashing the output **H** of the feature extraction block. The squashing function is defined as:

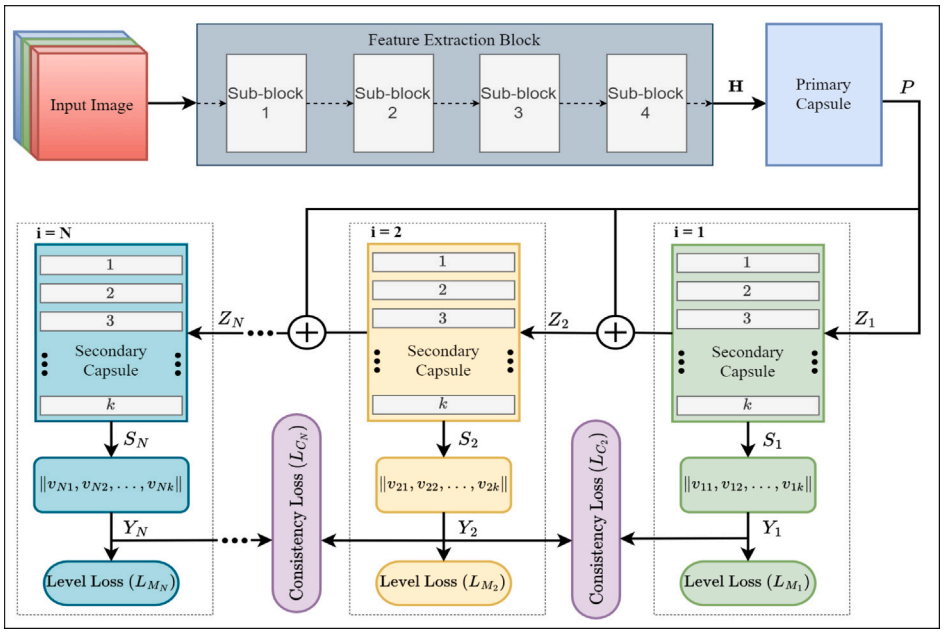
$$v_o = \frac{\|v_{in}\|^2}{1+\|v_{in}\|^2}\frac{v_{in}}{\|v_{in}\|} \tag{1}$$

where v_{in} is the total input vector and v_o is the output vector. In the context of capsule networks, each vector represents a capsule, and the squash function is designed to normalize the length of these vectors while preserving their direction. This process ensures that the length of each vector is between 0 and 1, which is crucial for the proper functioning of capsule networks. This transformation is vital because it allows the primary capsule layer to encapsulate not just the presence of features in the image, but also their instantiation parameters like pose, orientation, and spatial hierarchy [10]. The vectors formed as a result of this transformation are multi-dimensional, each representing a specific feature or pattern detected in the input image. In our implementation, the dimensionality of these vectors is determined by the complexity of the dataset and the intricacy of the label tree in the hierarchical classification task. The number of vectors in primary capsule layer, denoted by r, and their dimensionality, denoted by n_P, are thus tailored to suit the specific requirements of the dataset and the classification problem at hand. The transformed set of vectors, which encapsulate rich feature information, are then forwarded to the secondary capsule layers for further processing and refinement.

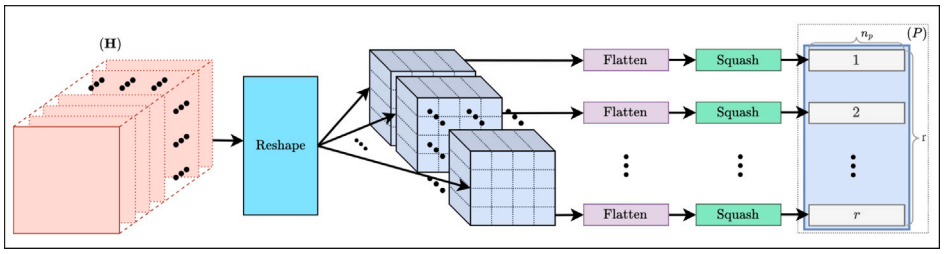
Secondary capsule layers in Fig. 2(a) are responsible for representing: (1) the hierarchical structure of the dataset, and (2) the probability of the presence of specific features or patterns in the input image. To effectively capture the hierarchical structure of the dataset, each secondary capsule layer (S_i) in the architecture represents a hierarchical level, and each capsule (k_i) in the same layer denotes a class in the hierarchical label tree. Here k is the class index in the hierarchical label tree at the ith hierarchical level. Thus, the number of secondary capsule layers in the architecture will be as many as the number of hierarchical levels, and the number of capsules in each layer will be as numerous as the number of classes in the corresponding level. As mentioned in Fig. 2(a), secondary capsule layer ($S_{i=1}$) representing the coarse level take the output of the primary capsule layer (P), whereas secondary capsule layers ($S_{i>1}$) representing the finer levels takes the output of both the previous secondary capsule layer (S_{i-1}) and the primary capsule layer (P). Therefore, the input of the secondary capsule layer follows the following conditional equation:

$$Z_i = \begin{cases} P, & \text{if } i = 1 \\ S_i(Z_{i-1} \| P), & \text{Otherwise} \end{cases} \tag{2}$$

Here, Z_i is the input for the secondary capsule layer for ith level. The output of the primary capsule layer is used to learn the presence of specific features or patterns in the input image, and the output of the previous secondary capsule layer is used to learn the hierarchical relationships between the features according to the data taxonomy using an iterative dynamic routing by agreement algorithm. This allows the model to learn the hierarchical structure of the dataset and to keep consistency between the levels. Each capsule in the secondary



(a) Architecture of hierarchical deep capsule network (HD-CapsNet)



(b) Primary Capsules

Fig. 2. (a) The HD-CapsNet extracts features from the input image using a feature extraction block, and then transforms the features into primary capsules P. Each secondary capsule layer S_i in the architecture represents a hierarchical level, with each capsule within the same layer denoting a class in the hierarchical label tree. Each S_i takes Z_i as input, as defined by Eq. (2). Further, to enforce hierarchical consistency, Eq. (6) is used to penalize inconsistencies in the predicted class probabilities across different levels. (b) The primary capsule layer is formed by first reshaping the extracted features, and then flattening and squashing them using the non-linear squashing function defined in Eq. (1).

capsule layer outputs a vector of length n_S and represents the probability of a class k in the hierarchical label tree at the ith hierarchical level. As mentioned in Eq. (2), the input of the secondary capsule layer for the first hierarchical level only contains the output of the primary capsule layer, as a result, the routing is done only between these two layers following the iterative dynamic routing by agreement algorithm [10]. However, for the remaining hierarchical levels, the input of the secondary capsule layer is a concatenation of the output vectors of the previous secondary capsule layer and the primary capsule layer, thus the input vectors of the secondary capsule layer contain both the feature properties and the hierarchical structure of the dataset. Therefore, the routing by agreement algorithm considers the previous level's output and the corresponding features to output much more accurate vectors for the current level. In this way, our HD-CapsNet is capable of learning the hierarchical structure of the dataset and utilizes the hierarchical structure to keep consistency between the levels.

In this network architecture, the output from the secondary capsules is processed through a designated prediction layer, denoted as Y_i, at each level of the hierarchical structure. This layer is responsible for predicting the class label of the image in accordance with the hierarchy defined in the label tree. Specifically, the prediction layer outputs a vector of probabilities, $Y_i = [\hat{y}_{i,1}, \hat{y}_{i,2}, \ldots, \hat{y}_{i,k}]$, where $\hat{y}_{i,k}$ is the probability for each class k in the ith hierarchy. This computation is based on normalizing the lengths of the capsule vectors as follows:

$$\hat{y}_{i,k} = \frac{\|v_{i,k}\|}{\sum_{j=1}^{k_i} \|v_{i,j}\|} \tag{3}$$

Here, $\|v_{i,k}\|$ is the length of the output vector of the kth capsule in the S_i secondary capsule layer. Thus, $\hat{y}_{i,k}$ evaluates the probabilities of each class by considering the vector lengths of the corresponding capsules in the secondary capsule layer. Therefore, the proposed HD-CapsNet model outputs a vector of probabilities for each class at each level of the hierarchy.

3.1. Loss function

As stated in CapsNet [10], capsule networks are trained using a margin loss function in order to foster the output vector corresponding to the actual label to have a higher magnitude than the other vectors. Hence, we made use of the margin loss function (L_M) for each secondary capsule layer (S_i) with a hierarchical level weight (Y_i) to keep a balance between the levels by prioritizing from coarse-to-fine levels. Further, we introduced a consistency loss function (L_C) to enforce the consistency between the levels. As a result, the total loss function (L_{Total}) is a weighted sum of the margin loss function (L_M) and the consistency loss function (L_C) as follows:

$$L_{\text{Total}} = Y_1 L_{M_1} + \sum_{i=2}^{N} Y_i[(1-\Lambda)L_{M_i} + \Lambda L_{C_i}] \tag{4}$$

Here Λ is the weight parameter for the consistency loss function. Λ defines the importance of the level consistency along with the margin

loss function which determines the level wise classification loss. The margin loss function (L_M) is defined as follows:

$$L_{M_i} = \sum_{k=1}^{i} T_k \max(0, m_+ - \|v_k\|)^2 + \lambda(1 - T_k)\max(0, \|v_k\| - m_-)^2 \quad (5)$$

Here, T_k is the target class label, v_k is the output vector of the kth capsule in the ith secondary capsule layer, λ is the down-weighting parameter for absent classes, m_+ and m_- are the margin parameters. The consistency loss function (L_C) in Eq. (4) is calculated based on the parent–child relationship between the classes in the hierarchical label tree. The consistency loss function (L_C) is defined as follows:

$$L_{C_i} = \begin{cases} 0, & \text{if } i \leq 1 \\ \sum_{j=1}^{K_i} \left(1 - P(k_{i,j} \mid k_{i-1})\right) \cdot (1 - c(k_{i,j})), & \text{if } i > 1 \end{cases} \quad (6)$$

Where, K_i is the total number of classes at the current level i, $k_{i,j}$ represents the jth label at level i, $c(k_{i,j})$ is the consistency score of the class label $k_{i,j}$, which is equal to 1 if it is consistent following the label tree and 0 otherwise. k_{i-1} represents the parent class label at the previous level $i-1$, $P(k_{i,j} \mid k_{i-1})$ is the probability of the class label $k_{i,j}$ belonging to the class label k_{i-1} at the previous level $i-1$. The probability $P(k_{i,j} \mid k_{i-1})$ is calculated using the function as follows:

$$P(k_{i,j} \mid k_{i-1}) = \frac{\exp(v_{i,j})}{\sum_{m=1}^{K_i} \exp(v_{i,m})} \quad (7)$$

Where $v_{i,j}$ is the logit output for class $k_{i,j}$ at level i given its parent class k_{i-1} at level $i-1$. The summation term represents the sum of the probabilities of all the selective classes at the current level that does not follow the label tree and is not consistent. As a result, the consistency loss function in Eq. (6) enforces the consistency between the levels by penalizing the probabilities of the classes that do not follow the label tree and are not consistent.

4. Experiments

We have undertaken numerous experiments to analyse the performance of our proposed HD-CapsNet model. To evaluate our proposed HD-CapsNet and other classifier mentioned in the literature, we have used six different image datasets: Fashion-MNIST [35], Marine-Tree [36], CIFAR-10 [37], CIFAR-100 [37], Caltech-UCSD Birds-200-2011 (CUB-200-2011) [38] and Stanford Cars [39]. Further, we have compared the performance of our proposed HD-CapsNet with the CapsNet in [10], the CNN based branch hierarchical classifier (B-CNN) in [8], hierarchical convolutional neural network (H-CNN) in [29], Condition-CNN method in [31], ML-CapsNet in [13], BUH-CapsNet in [33] and H-CapsNet approach in [17]. Note that, the baseline CapsNet in [10] is not a hierarchical classifier and targets only the fine level classes overlooking the hierarchical relationships. But it shares the same capsule architecture and the routing algorithm with our proposed model. The B-CNN in [8], H-CNN in [29], Condition-CNN in [31], ML-CapsNet in [13], BUH-CapsNet in [33] and H-CapsNet in [17] on the other hand is a global hierarchical classifier that targets all the classes in the hierarchy. Hence, we have used these classifiers to compare the performance of our proposed HD-CapsNet.

4.1. Datasets

As indicated earlier, we have used six different image datasets with different numbers of classes and hierarchical relationships in our experiments. The details of the datasets are as follows:

The *Fashion-MNIST* dataset is a dataset consist of 70,000 grayscale images of 10 different fashion items. The dataset is divided into 60,000 training images and 10,000 testing images. Each image is of size 28×28 pixels. The dataset is a balanced dataset with 6,000 images per class. The original dataset does not have any hierarchical structure. Hence, we have created a hierarchical structure for the dataset by grouping the classes into two additional levels as shown in [29]. The coarse level consists of two classes and the medium level consists of six classes. In the hierarchical structure, the classes in the coarse level are the parent classes of the classes in the medium level and the classes in the medium level are the parent classes of the classes in the fine level for the corresponding classes grouped under. Thus, the classes in the hierarchical structure form a parent–child relationship.

The *Marine-Tree* dataset is a dataset consist of 160,000 colour images of marine organisms divided into tropical, temperate and combined subsets. The dataset provides a hierarchical structure with five levels. For simplicity, we have used the first 3 levels of the hierarchical structure in our experiments. The coarse level consists of 2 classes, the medium level consists of 10 classes and the fine level consists of 38 classes. Further, we set the image size to 64×64 pixels.

In a related fashion, the *CIFAR-10* and *CIFAR-100* datasets are two different datasets consist of 60,000 colour images of 10 and 100 different child classes grouped under 20 different parent classes respectively. The datasets are divided into 50,000 training images and 10,000 testing images. Each image is of size 32×32 pixels. To achieve a 3 level hierarchical structure we have assigned 2 additional levels for CIFAR-10 and 1 additional level for CIFAR-100 dataset by following the footsteps of [8]. Therefore, in CIFAR-10 dataset, the additional coarse level consists of 2 classes and the medium level consists of 7 classes, and in CIFAR-100 dataset, the additional coarse level consists of 8 classes.

The CUB-200-2011 dataset contains colour images of 200 different bird species and the Stanford Cars dataset contains colour images of 196 different car models. We have followed the hierarchical structure provided in [32] for both the datasets to assign a 3 level hierarchical structure, where the training, validation and testing sets contain 5944, 2897 and 2897 images for CUB-200-2011 dataset and 8144, 4020 and 4021 images for Stanford Cars dataset respectively. The coarse, medium and fine levels consist of 39, 123 and 200 classes for CUB-200-2011 dataset and 13, 113 and 196 classes respectively for the Stanford Cars dataset. In our experiments we have specified the image size as 64×64 pixels for both the datasets.

4.2. Experimental setup

Through our experiments, we have used the same data preprocessing and augmentation techniques for all the datasets. Here we used Mix-Up data augmentation technique [40] to augment the training data. Mix-Up data augmentation technique is a simple yet effective data augmentation technique that generates new training samples by linearly interpolating between two randomly selected training instances. The interpolation is done by taking the weighted average of the two training samples and the corresponding labels. The weight is sampled from a beta distribution with a parameter α. The value of α is set to 0.2 for all the experiments. Here we make use of *Adam* optimizer with an exponential decay learning rate scheduler. The initial learning rate is set to 0.001 and the decay rate is set to 0.95 which is executed after 10 training epochs for all the experiments. 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. The decay rate of 0.95 gradually reduces the learning rate, which helps prevent the model from oscillating or diverging away from the optimal solution in the later stages of training. Further, we have trained the models for 100 epochs within all the datasets for a fair comparison.

As mentioned earlier in Section 3, the feature extraction block is composed of 4 sub-blocks containing 2 convolutional layers followed by a batch normalization layer and 1 max-pooling layer. For all the scenarios, the convolutional layers in the sub-blocks use a kernel size of 3×3 with zero padding and *ReLu* activation function. The number

of filters in the convolutional layers are set to 64, 128, 256 and 512 respectively for the 4 sub-blocks. The batch normalization layers in the sub-blocks are set with TensorFlow's default parameters, and the maxpooling layers use a kernel size of 2 × 2 with a stride of 2. Throughout all the experiments, we have predefined the value of n_P to be 8 for the primary capsule layer, as a result, the r value for the primary capsule layer varied according to the input image size as the number of filters feature extraction block is fixed.

Each dimension in the secondary capsules represents a feature or a pattern in the input image [10]. Therefore, the number of dimensions in the secondary capsules is determined by the complexity of the dataset and the intricacy of the label tree in the hierarchical classification task. In our experiment, the fashion-MNIST, CIFAR-10 and CIFAR-100 datasets are consist of simple images with a small number of classes in the hierarchical levels, whereas the Marine-Tree, CUB-200-2011 and Stanford Cars datasets are consist of more complex images with a large number of classes in the hierarchical levels. Hence, we have set the value of n_S for the secondary capsule layers to be 32, 16 and 8 for the coarse, medium and fine levels for the fashion-MNIST, CIFAR-10 and CIFAR-100 datasets respectively. For the Marine-Tree, CUB-200-2011 and Stanford Cars datasets, we have set the value of n_S to be 64, 32 and 16 for the coarse, medium and fine levels respectively. Here we trained the HD-CapsNet model employing the dynamic routing algorithm [10] with 2 routing iterations for all the datasets. For training the HD-CapsNet model, we have employed a dynamic loss weight distribution strategy as mentioned in [17] for the loss weights (Y_i) in Eq. (4). This loss weight distribution strategy is initially assigned to prioritize the hierarchical levels following the coarse-to-fine paradigm, and later adjust the weights based on the training accuracy. Additionally, the consistency weight (Λ) in Eq. (4) is set to 0.2 for all the experiments. The upper margin m_+ and the lower margin m_- in Eq. (5) are set to ensure that the output vector corresponding to the actual label has a magnitude greater than those corresponding to incorrect labels. In our experiments, we have set the value of m_+ to 0.9 and the value of m_- to 0.1 for all implementations. These values are chosen to encourage the length of the vector for a capsule to be longer than 0.9 if the capsule is associated with the correct label, and shorter than 0.1 if it is associated with an incorrect label. We determined these values through a hyperparameter search on the corresponding datasets.

The baseline CapsNet model is trained with the same hyperparameter as mentioned in [10] where the primary capsules are 8-dimensional and secondary capsules are 16-dimensional and uses dynamic routing with 2 iterations for all the datasets. For the B-CNN model we have used the base-B model as mentioned in [8] which does not include pre-trained weights. All other hyperparameters were kept the same as Zhu and Bain mentioned in [8]. Similarly, we have used the same hyperparameters as mentioned in [29] for H-CNN and in [31] for the Condition-CNN model. For the ML-CapsNet, BUH-CapsNet, and H-CapsNet models, we used the same hyperparameters as mentioned in [13,33], and [17], respectively, and kept the capsule dimensions the same as the HD-CapsNet model for a fair comparison.

4.3. Ablation study

We have also undertaken an ablation study to analyse the performance of the proposed HD-CapsNet model by comparing the performance of the model with different settings. The ablation study is conducted by training the model without the skip connections between the secondary capsule layers, and by training the model without the proposed consistency loss function. The ablation study is conducted on all the datasets mentioned in Section 4.1. We do this to analyse the effect of the skip connections between the secondary capsule layers and the proposed consistency loss function on the model performance. This is because the skip connections between the secondary capsule layers are used to learn the hierarchical structure of the dataset, and the proposed consistency loss function is used to enforce the consistency between the levels. Therefore, we expect the model to perform better with the skip connections between the secondary capsule layers and the proposed consistency loss function.

In order to analyse the effect of the skip connections between the secondary capsule layers on the model performance, we have trained the model without the skip connections between the secondary capsule layers. We have removed the skip connections between the secondary capsule layers in Fig. 2(a) while keeping all other hyperparameters the same. Hence, the input of the secondary capsule layers for the first hierarchical level solely encompasses the output of the primary capsule layer, while the input of the secondary capsule layers for the subsequent hierarchical levels solely assimilates the output of the preceding secondary capsule layer exclusively. Similarly, for analysing the effect of the proposed consistency loss function on the model performance, we have trained the model without the proposed consistency loss function. We have set the consistency weight (Λ) in Eq. (4) to 0.0 while keeping all other hyperparameters the same. As a result, the total loss function (L_{Total}) in Eq. (4) is a weighted sum of the margin loss function (L_M) only.

4.4. Evaluation metrics

Traditional evaluation metrics such as accuracy, precision, recall and F1-score are not sufficient to evaluate the hierarchical classification models [6], as these metrics overlook the hierarchical structure of the dataset. In datasets with complex class structures, where an instance may belong to multiple hierarchical levels, traditional metrics do not provide a comprehensive view of the model's ability to navigate and predict within this complex structure. In hierarchical models, misclassifying a label at a higher level in the hierarchy is usually more severe than a misclassification at a lower level. Traditional metrics treat all misclassifications equally, without accounting for the varying degrees of severity based on the hierarchical structure. As a result, traditional metrics do not account for this partial correctness, which is crucial in hierarchical settings. To comprehensively evaluate the performance of the proposed HD-CapsNet model and other alternatives in our experiments, we have employed both hierarchical and traditional metrics. Traditional classification metrics, such as level-wise accuracy and mean Average Precision (mAP), are used to assess general model performance. On the other hand, hierarchical metrics, which include hierarchical precision, hierarchical recall, hierarchical F1-score, Consistency, and the exact match score, take into account the hierarchical structure of the dataset. These metrics provide a more nuanced evaluation by considering how well the model performs within this hierarchical context.

Hierarchical Precision (hP) expands upon the traditional precision metric, tailored for assessing hierarchical classification systems. It quantifies the ratio of accurately predicted class labels (including ancestors) to the overall predicted labels. Hierarchical Recall (hR), an extension of the standard recall metric in hierarchical classification, gauges the ratio of correctly predicted true class labels (including ancestors) against the complete set of true labels. Hierarchical F1-score (hF1) amalgamates hierarchical precision and recall into a unified metric, akin to the conventional F1-score's combination of precision and recall. It represents the harmonic mean of hierarchical precision and recall. They are defined as follows:

$$hP = \frac{\sum_i |(\hat{Y}_i) \cap (Y_i)|}{\sum_i |(\hat{Y}_i)|} \tag{8}$$

$$hR = \frac{\sum_i |(\hat{Y}_i) \cap (Y_i)|}{\sum_i |(Y_i)|} \tag{9}$$

$$hF = \frac{2 \times hP \times hR}{hP + hR} \tag{10}$$

Here, $\hat{Y}_i$ and Y_i in Eqs. (8) and (9) is a set consisting of the predicted and true labels for the ith example and all its ancestors classes respectively.

Table 1
Performance of all the classification models on the Fashion-MNIST [35], CIFAR-10 [37] and CIFAR-100 [37] datasets. The best results are highlighted in bold. Here, † denotes the HD-CapsNet models without the proposed consistency loss L_C as specified in Eq. (4), and ‡ denotes those without the skip connections between the secondary capsule layers, respectively.

Dataset	Models	Trainable Params (M)	Level Wise Accuracy (%)			mAP	Hierarchical Metrices (%)				
			Coarse	Medium	Fine		hP	hR	hF1	Cons	EM
Fashion-MNIST	CapsNet	8.22	99.62	95.89	91.90	91.79	91.90	91.90	91.90	–	91.90
	B-CNN	9.41	99.63	95.44	92.33	98.23	95.77	96.48	96.07	96.73	90.44
	H-CNN	44.07	99.79	96.76	93.16	98.45	96.55	96.79	96.65	98.88	92.58
	Condition-CNN	46.66	99.78	96.65	93.42	98.53	96.65	96.84	96.73	99.16	92.85
	ML-CapsNet	11.01	99.70	95.89	92.10	97.85	95.85	96.19	95.99	98.35	91.31
	BUH-CapsNet	**4.79**	99.89	97.53	94.75	98.43	97.38	97.41	97.40	99.80	94.68
	H-CapsNet	14.18	99.73	97.06	93.95	98.69	96.86	97.36	97.07	97.60	92.69
	HD-CapsNet	4.82	**99.92**	**97.79**	94.83	**98.95**	97.51	97.54	97.52	**99.84**	**94.70**
	HD-CapsNet †	4.82	99.89	97.78	**94.92**	97.95	**97.53**	**97.59**	**97.55**	99.70	**94.70**
	HD-CapsNet ‡	**4.73**	99.91	97.63	94.66	98.33	97.40	97.42	97.41	99.80	94.60
CIFAR-10	CapsNet	17.39	93.19	76.53	70.42	64.87	70.42	70.42	70.42	–	70.42
	B-CNN	12.38	96.08	87.13	84.54	94.70	89.26	91.48	90.18	89.72	78.99
	H-CNN	52.69	96.01	86.71	81.29	93.11	87.89	89.90	88.72	90.21	76.88
	Condition-CNN	54.92	95.86	83.78	79.74	91.57	86.56	88.36	87.30	91.30	75.30
	ML-CapsNet	22.71	97.95	90.03	86.78	94.89	91.38	92.24	91.74	95.47	85.24
	BUH-CapsNet	**5.04**	98.72	93.81	90.84	94.62	94.41	94.59	94.48	99.06	90.56
	H-CapsNet	18.25	97.61	92.58	91.12	97.12	93.92	94.60	94.74	91.24	86.65
	HD-CapsNet	5.23	**98.79**	**94.28**	**91.22**	**97.32**	**94.74**	**94.89**	**94.80**	**99.18**	**90.95**
	HD-CapsNet †	5.23	98.71	94.01	90.97	97.22	94.53	94.73	94.62	98.99	90.58
	HD-CapsNet ‡	**4.84**	98.76	93.36	90.26	97.22	94.09	94.30	94.18	98.94	89.85
CIFAR-100	CapsNet	44.66	56.53	45.06	34.93	21.38	34.93	34.93	34.93	–	34.93
	B-CNN	12.48	71.08	61.99	56.38	68.05	64.41	73.42	67.93	56.87	38.90
	H-CNN	53.10	74.00	67.27	51.40	66.89	64.23	71.67	67.14	60.27	40.49
	Condition-CNN	55.30	73.38	61.27	47.91	62.30	61.07	67.18	63.45	65.01	39.50
	ML-CapsNet	80.75	78.73	70.15	60.18	71.57	69.50	75.65	71.89	68.92	50.29
	BUH-CapsNet	8.52	86.03	77.83	64.87	79.92	76.04	77.87	76.75	89.81	62.53
	H-CapsNet	23.73	80.31	75.68	65.74	77.59	76.93	78.65	77.12	65.25	53.92
	HD-CapsNet	**7.85**	**86.93**	**79.31**	**66.38**	**80.94**	77.43	79.20	78.12	89.80	**64.41**
	HD-CapsNet †	**7.85**	86.81	78.73	66.23	80.45	**77.84**	**79.56**	**78.52**	89.78	63.80
	HD-CapsNet ‡	**5.55**	86.57	78.33	57.08	63.30	73.86	75.00	74.31	**92.51**	56.10

The Consistency (Cons) score is a metric that tells us how many test examples match the hierarchy structure, regardless of whether they are actually correct or not. It is expressed as a percentage, showing the proportion of aligned test examples. Whereas, Exact Match (EM) score determines the percentage of predictions that completely match the ground truth across all levels of the hierarchy. This score provides an indication of how accurately the predictions align with the actual data. These are formulated as follows:

$$Cons = \frac{1}{N}\sum_{j=1}^{N}\prod_{i=1}^{L-1}\mathbb{1}\left(T_P(y_{i+1,j}) = y_{i,j}\right) \tag{11}$$

$$EM = \frac{1}{N}\sum_{j=1}^{N}\prod_{i=1}^{L}\mathbb{1}(y_{i,j} = \hat{y}_{i,j}) \tag{12}$$

where, L is the total number of hierarchical levels, N is the total number of examples and $\mathbb{1}(\cdot)$ is an indicator function that returns 1 if the condition inside is true, and 0 otherwise. In Eq. (11) hierarchical label tree is denoted by T where $T_P(y)$ returns the parent node for any child node y. In Eq. (11) and (12), $\mathbf{y}_{i,j}$ and $\hat{\mathbf{y}}_{i,j}$ represent the true and predicted labels at hierarchy level i for label j.

5. Results and discussion

Now we focus on the results yielded by our proposed HD-CapsNet model and other alternatives while employing on aforementioned six different image datasets. We commence by comparing the performance of the HD-CapsNet model with the baseline models i.e. CapsNet in [10], B-CNN approach in [8], H-CNN model in [29], Condition-CNN in [31], ML-CapsNet in [13], BUH-CapsNet in [33] and H-CapsNet in [17]. Subsequently, we compare the performance of HD-CapsNet against the ablation versions of it, as discussed in Section 4.3. In Tables 1 and 2, we present the overall performance of all the models under consideration across all datasets. Note that in the tables, our proposed HD-CapsNet yielded better classification accuracy compared to the alternatives. It is also important to mention that, the proposed HD-CapsNet model outperformed the alternative classification models in terms of other metrics as shown in the tables. Compared to the baseline CapsNet, it exceeds the model performance by a considerable margin. This is due to the fact that the HD-CapsNet model utilizes the hierarchical structure of the data to learn the hierarchical features which enhance the model performance. Further, compared to the B-CNN approach in [8], the H-CNN method in [29], and Condition-CNN in [31], the HD-CapsNet model yielded a better performance as the model is able to learn the hierarchical features in a more efficient manner. This efficiency is due to the inherent advantages of capsule networks, such as the ability to learn the spatial relationships between features and the capability to reduce noise in the data, over traditional CNN-based models. These advantages enable the HD-CapsNet model to learn the hierarchical features in a more optimized way. Traditional CNNs, lacking in these specific capabilities, do not optimize the learning of hierarchical features as effectively as capsule networks, which is a key factor in the improved performance of our HD-CapsNet model.

In comparison with the ML-CapsNet approach detailed in [13], the HD-CapsNet model also demonstrated superior performance. This is attributed to the differing methodologies employed to learn hierarchical features. The ML-CapsNet approach utilizes multiple secondary capsule layers with identical structures to capture the hierarchical features. It then implements dynamic routing between a shared primary capsule layer and each of these secondary capsule layers. This approach is less efficient because ML-CapsNet has to learn hierarchical features separately for each secondary capsule layer, essentially repeating the learning process for every layer rather than building upon previous layers' knowledge. Further, the BUH-CapsNet model in [33] also employs a hierarchical classification approach, but it uses a bottom-up learning strategy. This approach is less efficient because the BUH-CapsNet model has to learn the hierarchical features from the fine-level up to

Table 2
Performance of all the classification models on the Marine-Tree [36], Caltech-UCSD Birds-200-2011 (CUB-200-2011) [38] and Stanford Cars [39] datasets. The best results are highlighted in bold. Here, † denotes the HD-CapsNet models without the proposed consistency loss L_C as specified in Eq. (4), and ‡ denotes those without the skip connections between the secondary capsule layers, respectively.

Dataset	Models	Trainable Params (M)	Level Wise Accuracy (%)			mAP	Hierarchical Metrices (%)				
			Coarse	Medium	Fine		hP	hR	hF1	Cons	EM
Marine-tree	CapsNet	39.37	86.36	70.34	46.73	10.94	46.73	46.73	46.73	–	46.73
	B-CNN	31.29	88.28	75.88	54.48	30.02	72.69	77.03	74.42	80.63	47.29
	H-CNN	52.81	88.25	75.14	49.99	27.60	70.66	75.21	72.47	78.13	44.72
	Condition-CNN	88.07	88.75	76.64	53.99	31.33	72.91	76.46	74.34	82.66	49.10
	ML-CapsNet	39.22	86.62	68.21	37.06	12.26	62.91	66.79	64.45	79.92	34.30
	BUH-CapsNet	**9.78**	88.48	76.49	52.33	26.86	72.35	73.17	74.07	91.78	52.53
	H-CapsNet	54.45	88.38	77.49	52.44	26.85	72.93	**80.97**	**76.74**	83.07	54.85
	HD-CapsNet	13.58	**89.88**	**78.60**	**57.15**	**32.72**	**75.02**	76.04	75.44	**94.47**	**55.59**
	HD-CapsNet †	13.58	89.50	77.57	53.75	27.98	73.29	74.76	73.88	92.37	51.85
	HD-CapsNet ‡	**5.97**	86.98	77.82	55.04	26.24	73.35	75.76	74.36	86.95	49.34
CUB-200-2011	CapsNet	105.72	17.67	8.04	4.59	1.50	4.19	4.59	4.00	–	4.59
	B-CNN	**31.52**	34.00	17.60	13.15	12.87	21.65	**31.49**	25.27	14.74	3.24
	H-CNN	97.63	32.43	16.02	6.27	8.49	17.11	24.94	19.98	12.92	2.21
	Condition-CNN	88.89	38.97	20.88	13.37	13.34	23.35	28.04	25.97	23.47	7.58
	ML-CapsNet	55.74	35.01	20.30	13.75	15.26	23.05	29.14	25.35	25.26	8.55
	BUH-CapsNet	**38.43**	37.76	20.95	13.36	14.00	23.26	29.21	25.52	26.21	7.90
	H-CapsNet	127.26	31.76	21.59	14.13	15.64	23.13	30.12	25.94	13.63	5.80
	HD-CapsNet	106.01	**40.42**	**21.61**	**14.39**	**15.69**	**23.47**	30.33	**26.01**	**27.34**	**8.63**
	HD-CapsNet †	106.01	36.59	17.78	10.87	10.68	20.29	26.56	22.62	24.09	6.28
	HD-CapsNet ‡	47.56	35.66	16.98	2.14	2.96	14.97	20.86	17.13	21.44	1.55
Stanford Cars	CapsNet	104.08	23.75	6.44	4.58	1.72	4.05	4.58	4.08	–	4.58
	B-CNN	**31.50**	34.94	9.05	9.38	8.72	18.17	27.96	21.78	7.44	1.62
	H-CNN	97.60	33.49	10.55	6.83	7.59	16.78	25.55	20.02	9.14	1.56
	Condition-CNN	88.85	43.07	16.14	14.00	15.69	24.91	35.48	28.87	15.24	4.49
	ML-CapsNet	54.10	41.31	14.75	10.50	12.27	21.27	28.40	23.97	22.86	5.26
	BUH-CapsNet	36.43	43.70	14.97	9.52	10.95	21.61	27.27	23.78	28.12	6.12
	H-CapsNet	110.15	33.85	13.73	11.96	11.55	20.60	31.60	24.62	7.66	2.54
	HD-CapsNet	81.17	**53.34**	**19.52**	**14.05**	**16.21**	**26.73**	**35.69**	**29.73**	**29.15**	**8.13**
	HD-CapsNet †	81.17	47.50	16.39	11.74	12.65	23.56	31.40	26.50	25.76	6.19
	HD-CapsNet ‡	**25.85**	46.01	12.29	1.57	2.23	17.10	24.04	19.79	13.60	0.87

the coarse-level classes, which limits the model's ability to learn the hierarchical features effectively. Additionally, H-CapsNet in [17] is a hierarchical classification model that utilizes the hierarchical structure of the dataset to learn the hierarchical features. The trade-off here is that, since the H-CapsNet model uses multiple independent capsule layers to learn feature representations at each level of the hierarchy, the trainable parameters in the H-CapsNet model increase significantly. This may lead to overfitting. Moreover, none of the models enforce learning the hierarchical consistency between the levels, which can lead to suboptimal learning of hierarchical features. In contrast, the HD-CapsNet model architecture enables more efficient learning of hierarchical features by utilizing the dataset's inherent structure. The deeper capsule layers in the HD-CapsNet model allow for more optimized learning of hierarchical features, as the routing algorithm captures better agreement between the corresponding capsule layers. Consequently, the HD-CapsNet model outperformed the ML-CapsNet, BUH-CapsNet, and H-CapsNet models in terms of classification accuracy and other metrics, as demonstrated in the tables.

The findings are further corroborated by the data presented in Fig. 3. The figure shows the accuracy as a function of training epoch for the proposed HD-CapsNet model and other models as mentioned in the literature on the Fashion-MNIST, CIFAR-10, CIFAR-100, Marine-Tree, CUB-200-2011, and Stanford Cars datasets. Notably, in the plots, we illustrate the accuracy of the last level in the hierarchy on the test dataset. This is because the last level in the hierarchy represents the fine level classes, and the fine level classes are difficult to classify compared to the coarse and medium level classes. It is evident that in all scenarios, all the hierarchical classification methods yielded better classification accuracy than the flat classification CapsNet approach in [10]. This outcome is anticipated, as the hierarchical classification methods capitalize on the advantages of the data hierarchy. Furthermore, among the hierarchical methods, the proposed HD-CapsNet method not only performed better but also converged at a faster rate. This is expected, as the HD-CapsNet architecture leverages the data hierarchy by maintaining hierarchical consistency among the levels.

It is also worth noting that, overall the HD-CapsNet model performed better with the proposed loss function, which includes additional consistency loss, than without it. For training the model without the proposed loss function with additional consistency loss, we have set the consistency weight (Λ) in Eq. (4) to 0.0, while keeping all other hyperparameters the same. Fig. 4 shows the level wise classification accuracy of the HD-CapsNet model with and without the proposed loss function. Note that, in all the cases the HD-CapsNet model with the proposed loss function not only performed better but also converged faster. As mentioned in the Tables 1 and 2, the HD-CapsNet model with the proposed loss function outperformed the version without it in terms of hierarchical consistency across all datasets. This is because the consistency loss in the proposed loss function is designed to improve the generalization of the model by penalizing the model for making inconsistent predictions. Such a definition of the consistency loss not only improves the hierarchical consistency of the model but also improves the level wise classification accuracy. This is evident from the results shown in Tables 1 and 2, where the HD-CapsNet model with the proposed loss function yielded better accuracy than the HD-CapsNet model without the proposed loss function on all the datasets. Further, in Tables 1 and 2 we show the hierarchical matrices such as hierarchical precision, hierarchical recall, hierarchical F1-score and exact match score for all the datasets. Note that, for the complex datasets such as Marine-Tree, CUB-200-2011 and Stanford Cars, the HD-CapsNet model with the proposed loss function yielded better hierarchical precision, hierarchical recall, hierarchical F1-score and exact match score compared to the HD-CapsNet model without the proposed loss function. Similarly, the proposed HD-CapsNet model performed better with the skip connections than without the connections, as shown in Tables 1 and 2, and in Fig. 4. However, for the CIFAR-100 dataset, the HD-CapsNet model without the skip connections performed better in terms

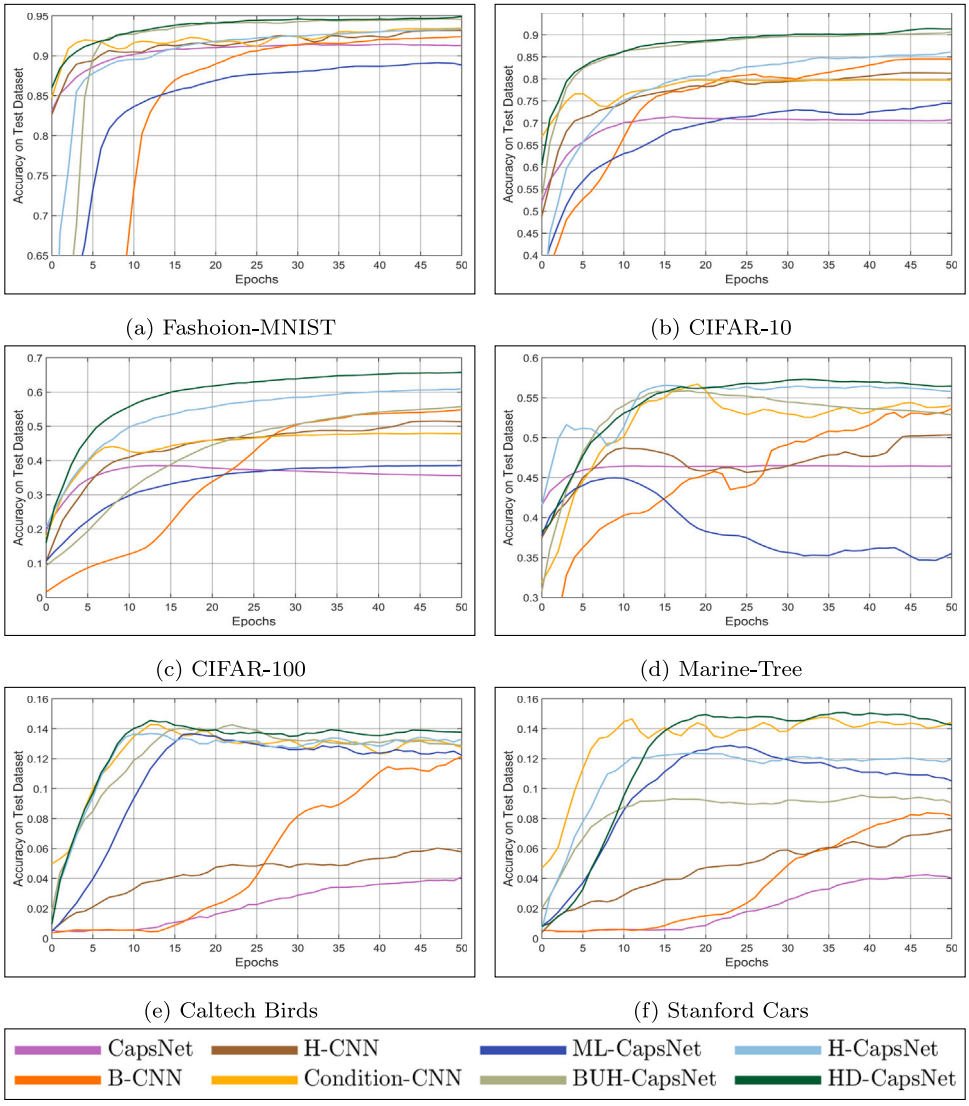


Fig. 3. Accuracy of the finest level as a function of training epochs for the proposed HD-CapsNet and alternative models: CapsNet in [10], B-CNN in [8], Condition-CNN in [31], ML-CapsNet in [13], BUH-CapsNet in [33], and H-CapsNet in [17], across datasets (a) Fashion-MNIST, (b) CIFAR-10, (c) CIFAR-100, (d) Marine-Tree, (e) CUB-200-2011, and (f) Stanford Cars.

of consistency score than the model with the skip connections. This occurs because the hierarchical structure of the CIFAR-100 dataset is balanced, and the classes are evenly distributed at the last level. Consequently, the ablated version of the HD-CapsNet model tends to overfit, leading to a drop in accuracy. This tendency is evident in Fig. 4(c), where the HD-CapsNet model without skip connections quickly overfits, resulting in decreased accuracy. As a result, the HD-CapsNet model without the skip connections yields a significantly lower EM score. It is also worth mentioning that, the HD-CapsNet model with the proposed loss function performed better in this complex datasets with fewer trained examples. Compared to the other models, this difference in model performance is even more apparent.

The proposed HD-CapsNet model has shown promising results in addressing the challenges of hierarchical image classification. Compared to the baseline CapsNet model, the HD-CapsNet model yielded better classification accuracy while using fewer trainable parameters on most datasets, as mentioned in Tables 1 and 2. This is not surprising as the HD-CapsNet model utilizes the feature extraction block to reduce the dimensionality of the extracted features while keeping the spatial relationships between the features intact. On the other hand, the baseline CapsNet model relies on the convolutional layers to extract the features in order to form the primary capsules, which increases the number of trainable parameters. Further, the inputs to the secondary capsules corresponding to the finer levels appear to improve the level wise accuracy and hierarchical consistency while not affecting the routing process. This is because the routing process is based on the agreement between the secondary capsules and the concatenation of the primary capsules, as well as the outputs of the previous secondary capsules. Therefore, the routing process not only allows the model to learn the hierarchical features in a more optimized way, but also allows the model to learn the spatial relationships between the complex features accountable for hierarchical consistency.

One interesting observation from our experiments is that, for the fashion-MNIST and CIFAR-100 dataset the HD-CapsNet model with the proposed loss function yielded better hierarchical consistency and EM score but not better hierarchical precision, hierarchical recall, hierarchical F1-score and exact match score compared to the HD-CapsNet model without the proposed loss function. Although this difference in model performance is not significant, it is worth mentioning that, the HD-CapsNet model with the proposed loss function is designed to enforce the hierarchical consistency and is performing as expected. The exact match score is a metric that measures the correctly classified hierarchical labels following the label tree, whereas hierarchical consistency measures the percentage of prediction follows the label tree.

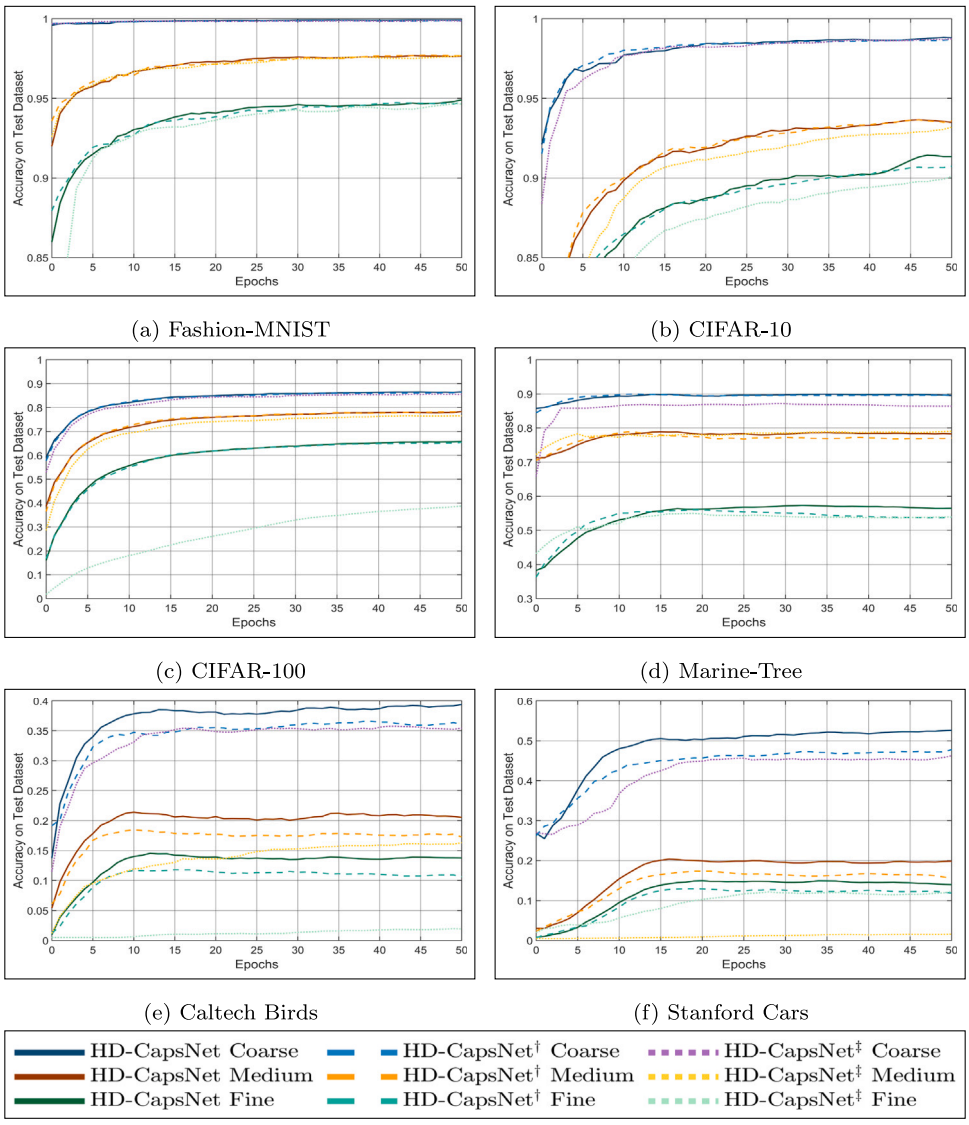


Fig. 4. Accuracy as a function of training epoch for the proposed HD-CapsNet model and its ablations on (a) Fashion-MNIST, (b) CIFAR-10, (c) CIFAR-100, (d) Marine-Tree, (e) CUB-200-2011 and (f) Stanford Cars datasets. Here, † denotes the HD-CapsNet models without the proposed consistency loss L_C as specified in Eq. (4), and ‡ denotes those without the skip connections between the secondary capsule layers, respectively.

Hierarchical precision, recall and F1-score are metrics that are more sensitive to class imbalance and hierarchical structure. In hierarchical label trees, some categories are related to others, misclassifying an instance can affect multiple levels of the hierarchy. Therefore, hierarchically consistent misclassified instances can affect the hierarchical precision, recall and F1-score.

6. Conclusion

In this paper, we have proposed a novel hierarchical deep capsule network (HD-CapsNet) model for hierarchical multi-label classification methods. Our proposed architecture introduces a new approach to the traditional capsule network by including a multi-level hierarchical structure following the data hierarchy to improve the accuracy of image classification. Additionally, we have introduced a loss function that enforces consistency between the hierarchical levels, improving the model's ability to learn and generalize. The results of our experiments on different datasets support the claim that the proposed HD-CapsNet outperforms the alternative deep learning models in terms of classification accuracy and hierarchical metrics. Our proposed model achieves significant improvement in accuracy while ensuring low computational complexity. Overall, the proposed HD-CapsNet appears to be a favourable strategy for hierarchical image classification tasks. We believe that our work has opened up new possibilities for exploring hierarchical deep learning architectures with improved performance and generalization capabilities. Future research can extend our findings by exploring ways to incorporate more complex hierarchical structures, different loss functions and routing by agreement algorithm to further improve the performance of hierarchical deep learning models.

CRediT authorship contribution statement

Khondaker Tasrif Noor: Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **Antonio Robles-Kelly:** Supervision. **Leo Yu Zhang:** Supervision. **Mohamed Reda Bouadjenek:** Supervision. **Wei Luo:** Supervision.

Declaration of competing interest

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

Data availability

Data will be made available on request.

References

[1] Y. Guo, Y. Liu, E.M. Bakker, Y. Guo, M.S. Lew, CNN-RNN: A large-scale hierarchical image classification framework, Multimedia Tools Appl. 77 (8) (2018) 10251–10271, http://dx.doi.org/10.1007/s11042-017-5443-x.

[2] K. Kowsari, R. Sali, L. Ehsan, W. Adorno, A. Ali, S. Moore, B. Amadi, P. Kelly, S. Syed, D. Brown, HMIC: hierarchical medical image classification, a deep learning approach, Information 11 (6) (2020) 318, http://dx.doi.org/10.3390/info11060318.

[3] Q. Jiao, Z. Liu, G. Li, L. Ye, Y. Wang, Fine-grained image classification with coarse and fine labels on one-shot learning, in: 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2020, pp. 1–6, http://dx.doi.org/10.1109/ICMEW46912.2020.9105959.

[4] Q. Li, A. Liang, H. Liu, Hierarchical semantic segmentation of image scene with object labeling, J. Image Video Proc. 2018 (1) (2018) 15, http://dx.doi.org/10.1186/s13640-018-0254-1.

[5] 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 Computer Vision, 2015, pp. 2740–2748.

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

[7] F.M. Miranda, N. Köhnecke, B.Y. Renard, HiClass: A python library for local hierarchical classification compatible with scikit-learn, J. Mach. Learn. Res. 24 (29) (2022) 1–17, http://dx.doi.org/10.48550/arXiv.2112.06560, arXiv:2112.06560.

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

[9] H. Guo, K. Zheng, X. Fan, H. Yu, S. Wang, Visual attention consistency under image transforms for multi-label image classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 729–739.

[10] S. Sabour, N. Frosst, G.E. Hinton, Dynamic Routing Between Capsules, in: Advances in Neural Information Processing Systems, 30, Curran Associates, Inc., 2017, http://dx.doi.org/10.48550/arXiv.1710.09829.

[11] C. Lu, R. Krishna, M. Bernstein, L. Fei-Fei, Visual relationship detection with language priors, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, in: Lecture Notes in Computer Science, Springer International Publishing, Cham, 2016, pp. 852–869, http://dx.doi.org/10.1007/978-3-319-46448-0_51.

[12] G.E. Hinton, S. Sabour, N. Frosst, Matrix capsules with EM routing, in: International Conference on Learning Representations, 2018.

[13] 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, in: Lecture Notes in Computer Science, Springer International Publishing, Cham, 2022, pp. 163–172, http://dx.doi.org/10.1007/978-3-031-23028-8_17.

[14] 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, in: Lecture Notes in Computer Science, Springer International Publishing, Cham, 2021, pp. 541–552, http://dx.doi.org/10.1007/978-3-030-86362-3_44.

[15] 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, http://dx.doi.org/10.1109/CVPR.2019.01098.

[16] J. Zhang, Q. Xu, L. Guo, L. Ding, S. Ding, A novel capsule network based on deep routing and residual learning, Soft. Comput. 27 (12) (2023) 7895–7906, http://dx.doi.org/10.1007/s00500-023-08018-x.

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

[18] B. Wang, X. Hu, C. Zhang, P. Li, P.S. Yu, Hierarchical GAN-tree and bi-directional capsules for multi-label image classification, Knowl.-Based Syst. 238 (2022) 107882, http://dx.doi.org/10.1016/j.knosys.2021.107882.

[19] J. Xu, H. Tian, Z. Wang, Y. Wang, W. Kang, F. Chen, Joint input and output space learning for multi-label image classification, IEEE Trans. Multimed. 23 (2021) 1696–1707, http://dx.doi.org/10.1109/TMM.2020.3002185.

[20] J. Gu, Interpretable Graph Capsule Networks for Object Recognition, Proceedings of the AAAI Conference on Artificial Intelligence 35 (2) (2021) 1469–1477, http://dx.doi.org/10.1609/aaai.v35i2.16237.

[21] R. LaLonde, U. Bagci, Capsules for Object Segmentation, 2018, http://dx.doi.org/10.48550/arXiv.1804.04241, arXiv:1804.04241.

[22] G.E. Hinton, A. Krizhevsky, S.D. Wang, Transforming auto-encoders, in: International Conference on Artificial Neural Networks, Springer, 2011, pp. 44–51.

[23] C. Xiang, L. Zhang, Y. Tang, W. Zou, C. Xu, MS-CapsNet: A novel multi-scale capsule network, IEEE Signal Process. Lett. 25 (12) (2018) 1850–1854.

[24] D. Pan, Y. Lu, P. Kang, A deep learning model for multi-label classification using capsule networks, in: D.-S. Huang, V. Bevilacqua, P. Premaratne (Eds.), Intelligent Computing Theories and Application, in: Lecture Notes in Computer Science, Springer International Publishing, Cham, 2019, pp. 144–155, http://dx.doi.org/10.1007/978-3-030-26763-6_14.

[25] S. Ramasinghe, C.D. Athuraliya, S.H. Khan, A Context-aware Capsule Network for Multi-label Classification, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018.

[26] 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 Recognit. Lett. 138 (2020) 638–643, http://dx.doi.org/10.1016/j.patrec.2020.09.010.

[27] J. Jayasree, A.V. Madhavi, G. Geetha, Multi-Label Classification On Aerial Images Using Deep Learning Techniques, in: 2023 International Conference on Networking and Communications (ICNWC), 2023, pp. 1–6, http://dx.doi.org/10.1109/ICNWC57852.2023.10127406.

[28] J. Zhang, J. Ren, Q. Zhang, J. Liu, X. Jiang, Spatial Context-Aware Object-Attentional Network for Multi-Label Image Classification, IEEE Trans. Image Process. 32 (2023) 3000–3012, http://dx.doi.org/10.1109/TIP.2023.3266161.

[29] Y. Seo, K.-s. Shin, Hierarchical convolutional neural networks for fashion image classification, Expert Syst. Appl. 116 (2019) 328–339, http://dx.doi.org/10.1016/j.eswa.2018.09.022.

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

[31] B. Kolisnik, I. Hogan, F. Zulkernine, Condition-CNN: A hierarchical multi-label fashion image classification model, Expert Syst. Appl. 182 (2021) 115195, http://dx.doi.org/10.1016/j.eswa.2021.115195.

[32] 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, in: CIKM '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 3833–3837, http://dx.doi.org/10.1145/3511808.3557534.

[33] 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, http://dx.doi.org/10.1109/DICTA60407.2023.00052.

[34] S. Santurkar, D. Tsipras, A. Ilyas, A. Madry, How Does Batch Normalization Help Optimization? in: Advances in Neural Information Processing Systems, 31, Curran Associates, Inc., 2018.

[35] H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms, 2017, http://dx.doi.org/10.48550/arXiv.1708.07747, arXiv:1708.07747.

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

[37] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Tech. Rep., Toronto, ON, Canada, 2009.

[38] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200–2011 Dataset, (2010–001) 2011, Dataset, https://resolver.caltech.edu/CaltechAUTHORS:20111026-120541847.

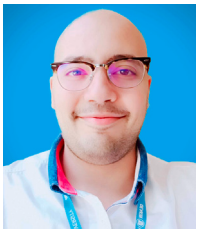
[39] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D Object Representations for Fine-Grained Categorization, in: 2013 IEEE International Conference on Computer Vision Workshops, IEEE, Sydney, Australia, 2013, pp. 554–561, http://dx.doi.org/10.1109/ICCVW.2013.77.

[40] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, Mixup: Beyond empirical risk minimization, 2017, arXiv preprint arXiv:1710.09412.

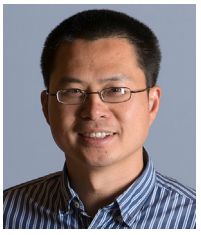
Khondaker Tasrif Noor received a B.Sc. degree in Electrical and Electronics Engineering from BRAC University, Bangladesh. Following this, he completed a Master's degree in Electronics Engineering from Macquarie University, Australia. He is currently a Ph.D. candidate at the School of Information Technology, Deakin University, Australia. His research primarily focuses on the development and optimization of neural architectures for hierarchical classification.

Prof. Antonio Robles-Kelly is a Group Leader with the Defence Science and Technology Group (DSTG). Prior to joining DSTG, he held a Professorial Chair at Deakin, where he also served as the Associate Head of School of IT (Research). He has held Senior Researcher positions at Data61 and NICTA, been an Adjunct A/Prof at the ANU, a Visiting Scientist at CSIRO Astronomy and Space, a Postdoctoral Fellow of the Australian Research Council and the President of the Australian Pattern Recognition Society. He is an associate editor of the Pattern Recognition Journal and IET Computer Vision, a Senior Member of the IEEE and an Honorary Professor at Deakin. He has also been a technical committee member of several mainstream computer vision and pattern recognition conferences. His research interests are in the areas of pattern recognition, machine learning, computer vision and artificial intelligence.

Dr Leo Yu Zhang (M'17) is currently a Senior Lecturer with the School of Information and Communication Technology, Griffith University, QLD, Australia. And he used to be a faculty member at the School of Information Technology, Deakin University, from 2018 to 2023. He received the Ph.D. degree from City University of Hong Kong in 2016. He held various research positions with the City University of Hong Kong, the University of Macau, the University of Ferrara, and the University of Bologna. His current research focuses on trustworthy AI and applied cryptography, and he has published over 100 articles in refereed journals and conferences, such as TIFS, TDSC, Oakland, AsiaCCS, NeurIPS, CVPR, ICCV, IJCAI, AAAI, etc.



Dr Mohamed Reda Bouadjenek is a Senior Lecturer (Assistant Professor) of Applied Artificial Intelligence within the School of Information Technology at Deakin University, Australia. His previous roles include Research Fellow positions at The University of Toronto (2017-2019) and The University of Melbourne (2015-2017). Prior to that, he served as a postdoc researcher at the French Institute for Research in Computer Science and Automation (INRIA) from 2014 to 2015 in France. Reda holds a Ph.D. and an M.Sc. in Computer Science from the University of Paris-Saclay, France, earned in 2013 and 2009, respectively. He also obtained a B.Sc. in Computer Science from the University of Science and Technology Houari Boumediene in 2008. Reda's research expertise encompasses a wide range of topics within the data-driven domains of Machine Learning, Deep Learning, and Information Retrieval. Leveraging analytical and algorithmic tools from these fields, he addresses real-world challenges across various applications, including recommender systems, interactive visual search interfaces, social network analysis, and data quality.



Dr Wei Luo is a Senior Lecturer in the School of IT at Deakin University. He holds a Ph.D. in Computing Science from Simon Fraser University. Dr. Wei Luo's research focuses on improving the reliability of machine learning models to benefit the broader community.