2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA) | 979-8-3503-8220-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/DICTA60407.2023.00052

A Bottom-Up Capsule Network for Hierarchical Image Classification

Khondaker Tasrif Noor*, Antonio Robles-Kelly*†, Leo Yu Zhang*‡ and Mohamed Reda Bouadjenek*

*School of Information Technology, Deakin University, Waurn Ponds, VIC, Australia
†Defence Science and Technology Group, Edinburgh, SA 5111, Australia
‡School of Information and Communication Technology, Griffith University, Brisbane, Australia

Email: k.noor@research.deakin.edu.au, antonio.robles-kelly@deakin.edu.au,
leo.zhang@griffith.edu.au, reda.bouadjenek@deakin.edu.au

***Abstract*—Hierarchical image classification is an arduous task in deep learning and computer vision. It requires classifying multiple image classes following a taxonomy or data hierarchy. This paper introduces a bottom-up hierarchical capsule network (BUH-CapsNet) designed to address hierarchical multi-label classification. The hierarchical structure of BUH-CapsNet allows it to build a tree-like structure for classification problems, making use of the data hierarchy. This structure enables the network to learn more complex relationships in the taxonomy by balancing the hierarchical levels and following the fine-to-coarse paradigm, leading to more accurate classification results. Furthermore, the bottom-up architecture of the BUH-CapsNet enforces hierarchical consistency, using the hierarchical structure of the datasets. We trained our BUH-CapsNet considering the hierarchical level weights that keep a balance between the levels. Experiments on six widely available datasets show that BUH-CapsNet achieves better results than existing multi-label classification methods and performs better when handling hierarchical labels.**

***Index Terms*—Image Classification, Hierarchical Multi-label Classification, Capsule Network**

I. Introduction

Image classification is an important task in computer vision and image processing, which involves assigning a set of classes to an image based on its visual content. Image classification has a wide range of applications, such as facial recognition [1], object detection [2], and scene recognition [3]. These classification methods often employ a flat classification approach by assigning each image to a single category based on its visual content, while ignoring the similarities present among the similar categories [4]. Hierarchical multi-label classification (HMC) has gained significant attention in recent times due to its ability to classify instances into multiple classes organized in a hierarchical structure [5]. This approach leverages machine learning methods to tackle complex classification problems across various application fields. Notably, HMC has proven its effectiveness in domains such as medical imaging, where it has been used for tasks like disease diagnosis and identification [6]. Additionally, HMC has been successfully applied in CCTV inspection systems for detecting and classifying objects of interest in surveillance videos [7]. Furthermore, in the context of E-commerce systems, HMC enables personalized product recommendations by classifying items into a hierarchical taxonomy [8]. Even in remote sensing applications, HMC has exhibited its utility by accurately categorizing land cover types and analyzing satellite imagery data [9]. To construct hierarchical classifiers, two common approaches are employed: utilizing hierarchical trees or direct acyclic graphs (DAGs) as the structure [10]. In both cases, convolutional neural networks (CNNs) have emerged as the backbone architecture for achieving state-of-the-art performance. The hierarchical structure provides a flexible framework for capturing the inherent relationships among classes, enabling the model to exploit both global and local dependencies within the data. By leveraging the hierarchical organization of labels, HMC facilitates more interpretable classification outputs, allowing for fine-grained predictions that reflect the complex relationships between classes.

Typically, the architecture of models designed for HMC problems has primarily followed a top-down approach, as highlighted by Naik et al. [11]. This top-down strategy aligns with the coarse-to-fine paradigm, where predictions are generated from the highest-level nodes of the hierarchical tree down to the lowest-level nodes. While this approach has demonstrated success in capturing high-level semantic information, it often results in deep neural connections and can suffer from challenges such as inconsistent predictions and difficulties in modeling fine-grained details. In contrast, the bottom-up approach offers an alternative perspective in addressing HMC tasks by focusing on the fine classes located at the last level of the hierarchical tree. It leverages the information from these fine classes to make predictions for the coarser classes in higher levels [12]. In HMC, this allows for a more granular understanding of the hierarchy, as it starts with the basic building blocks and works its way up. This approach enables the model to identify classes detailedly, building up an understanding of the patterns from the bottom to the top level. Additionally, the bottom-up approach eliminates the need for extensive feature engineering, which is often required in the top-down paradigm. By leveraging the inherent hierarchical structure, the model can learn and extract discriminative features directly from the data. This not only simplifies the modeling process but also allows for a more efficient and effective approach to

979-8-3503-8220-4/23/$31.00 ©2023 IEEE
DOI 10.1109/DICTA60407.2023.00052

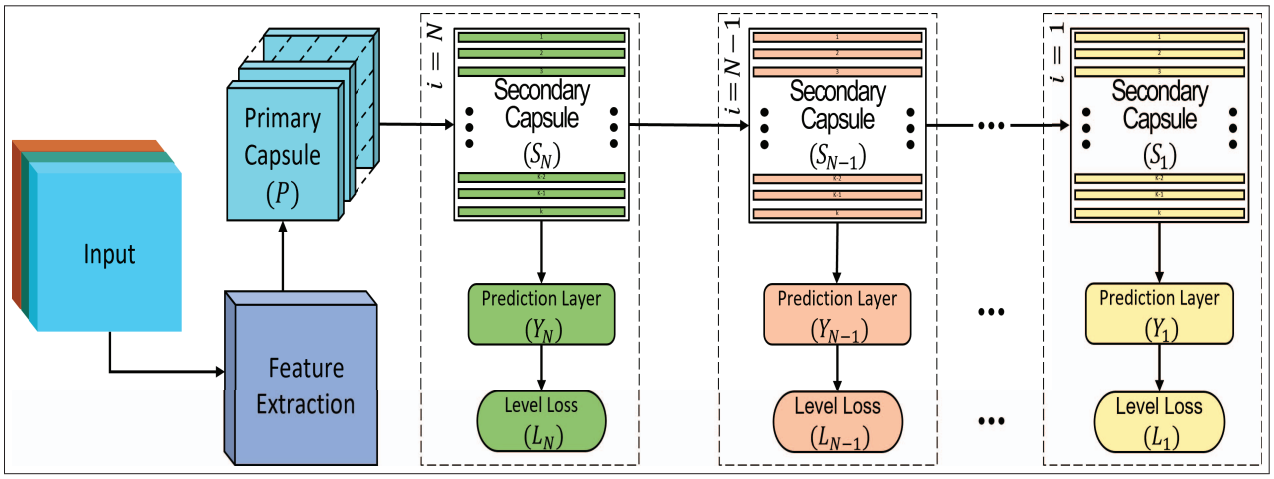

Fig. 1: Architecture of bottom-up hierarchical capsule network (BUH-CapsNet). Each secondary capsule layer in the architecture represents a hierarchical level, and each capsule in the same layer denotes a class in the hierarchical label tree.

hierarchical classification.

In this paper, we introduce a novel approach called Bottom-Up Hierarchical Capsule Network (BUH-CapsNet) specifically designed for addressing hierarchical multi-label classification (HMC) problems. Our proposed method builds upon the concepts of the capsule network (CapsNet) [13], which offers several advantages over traditional convolutional neural networks (CNNs) by utilizing a set of neurons to capture both features and their transformations. One notable strength of CapsNet is its ability to model part-whole relationships between different capsules in different layers, allowing them to learn hierarchical relationships between features [14]. This intrinsic property makes CapsNet well-suited for capturing the hierarchical structure of labels in HMC tasks.Recent studies have demonstrated that CapsNet outperforms CNNs in terms of accuracy and robustness [15], highlighting its potential for advancing hierarchical classification tasks. However, it has been observed that simply stacking multiple capsule layers on top of each other does not necessarily lead to improved model performance at the same level [16]. To overcome this limitation, a hierarchical organization of capsule layers has been proposed, showing promising results in terms of level-wise improvement and overall performance enhancement [17]. This hierarchical arrangement enables capsules to effectively capture the complex relationships between features and classes in a hierarchical manner, facilitating more accurate and meaningful predictions. Moreover, selective feature extraction has been explored in conjunction with capsule networks, resulting in a reduction in trainable parameters while simultaneously improving performance [18]. This approach allows the network to focus on the most discriminative features at each level of the hierarchy, enhancing both efficiency and effectiveness.

In a nutshell, the proposed BUH-CapsNet inherits the advantage of CapsNet in capturing part-whole relationships as well as taking the structure of HMC into consideration. Emphasizing image classification, we have trained and evaluated our BUH-CapsNet on six different widely available datasets. The remaining of this paper is organized as follows. In Sec. II, we provide a detailed explanation of the BUH-CapsNet architecture, highlighting its key components and working methodology. In this section, we outline how our model leverages the hierarchical capsule network approach to tackle HMC tasks. In Section III, we present the experimental setup conducted to evaluate the performance of BUH-CapsNet. We provide comprehensive details regarding the datasets used, training procedures, and evaluation metrics. Additionally, we report and analyze the test results obtained from the experiments, comparing BUH-CapsNet against alternative approaches to demonstrate its superior performance. Finally, we discuss the contribution and conclude this paper in Sec. IV.

II. Bottom-Up Hierarchical Capsule Network

Generally, in hierarchical image classification problems, the class labels are typically organized in a tree taxonomy, where each instance can be assigned to multiple classes at different levels of the hierarchy. To address this challenge, our proposed BUH-CapsNet leverages capsule layers to classify the class labels based on the hierarchical taxonomy for each instance. The overall architecture of our BUH-CapsNet network is presented in Fig. 1. It is designed to capture the hierarchical relationships between features and classes, enabling accurate and interpretable predictions. The architecture consists of several key components, each serving a specific purpose in the hierarchical classification process.

In the initial stage of BUH-CapsNet, the raw image data is fed as input to the network. The network begins with a feature extraction block, which consists of convolutional layers, batch normalization, and pooling layers. This feature extraction process aims to capture local features from the input image that are relevant for classification. These local features

play a crucial role in subsequent hierarchical classification steps. The extracted local features are then reshaped and grouped to form primary capsule layers (P), as shown in Fig. 1. Primary capsules encapsulate the local features and encode them into vectors representing the presence of specific visual patterns or attributes. These vectors capture important information about the primary features detected in the input image.

The output vectors from the primary capsule layers are then passed on to the secondary capsule layer (S_N) at the bottom level of the hierarchy, where N represents the total number of hierarchies present in the label tree. Each secondary capsule in this layer corresponds to a specific class or label within the hierarchy. The primary capsules' output vectors are used as inputs to the secondary capsule layer for the N_{th} level, allowing the network to learn the relationships between primary features and the classes at the bottom level of the hierarchy. To form the complete hierarchical architecture, output from each capsule in the secondary capsule layer goes to the next capsule layer (i.e., output from S_i goes to S_{i-1} for $i \in \{2, 3, \cdots, N\}$), which consists of capsules for classifying the parent classes in taxonomy[1]. It is also worth mentioning that, each secondary capsule layer (S_i) contains C_i number of capsules, where C_i is the number of classes present in the hierarchical level i and each capsule in the corresponding hierarchy is responsible for a class label.

In the BUH-CapsNet model, a bottom-up strategy is employed to implement dynamic routing [13] between the capsule layers. This means that each parent capsule in the hierarchy updates its information from the child capsules using an iterative routing method. This bottom-up architecture allows BUH-CapsNet to take full advantage of CapsNet's ability to capture part-whole relationships, thereby enhancing hierarchical consistency. By considering the outcomes from the child classes, the parent class outputs predictions, leading to improved performance in capturing the hierarchical relationships between classes.

As shown in Fig. 1, each secondary capsule layer (S_i) also has its own prediction layer (Y_i). The primary purpose of these prediction layers is to compute the class predictions for the classes present in the corresponding level of the hierarchy. The prediction layers utilize the vector outputs $\mathbf{v}_k^i$ obtained from the secondary capsule layer (S_i) to make these predictions. Where, k represent a class in the i^{th} hierarchical level. More specifically, for each secondary capsule in layer (S_i), the output vector $\mathbf{v}_k^i$ represents the instantiation parameters of the capsule corresponding to class k. These parameters encode various attributes and properties related to the presence and characteristics of the class within the given input data. The prediction layer (Y_i) processes these vector outputs to generate the classification predictions. By having separate prediction layers (Y_i) for each secondary capsule layer (S_i), our model outputs the class predictions at each level of the hierarchy. This hierarchical prediction scheme allows the model to capture the intricate relationships between the classes in a hierarchical manner, providing more nuanced and accurate predictions.

The overall loss function L_T for our proposed model is a weighted summation of the hierarchical level-wise loss $L_{i,k}$, which is formulated as

$$L_T = \sum_{i=1}^{N} \sum_{k=1}^{C_i} \lambda_i \, L_{i,k}, \tag{1}$$

where λ_i is the hierarchical level weight for level i. In more detail, the hierarchical level-wise loss $L_{i,k}$ is a modified hinge loss [13] with the form

$$\begin{aligned} L_{i,k} = {} & T_{i,k} \max\left(0, m^+ - \left\|\mathbf{v}_k^i\right\|\right)^2 \\ & + \omega \left(1 - T_{i,k}\right) \max\left(0, \left\|\mathbf{v}_k^i\right\| - m^-\right)^2, \end{aligned} \tag{2}$$

where $T_{i,k}$ is 1 if the class label is present; otherwise it will be 0, and m^+, m^- and ω are hyperparameters. By utilizing this loss formulation, our model learns to enforce a margin between relevant and non-relevant classes at each hierarchical level, facilitating the accurate classification of instances within the hierarchical structure.

III. Results and Experiments

In our experiments, we evaluate our BUH-CapsNet model by making use of six different image datasets: EMNIST [19], Fashion-MNIST [20], CIFAR-10 [21], CIFAR-100 [21], Caltech Birds-200-2011 [22], and Stanford Cars [23]. These datasets offer a diverse range of image classification challenges, allowing us to evaluate the robustness and generalizability of BUH-CapsNet across different domains. To provide a comprehensive comparison, we benchmark BUH-CapsNet against two well-established baseline models: the flat classifier CapsNet proposed by Sabour et al. [13] and the hierarchical classifier B-CNN proposed by Zhu et al. [24], which is based on the CNN architecture. It is worth noting that the baseline CapsNet in [13] focuses solely on the classes present in the bottom level of the hierarchy, overlooking the taxonomical relation in the data hierarchy. However, it shares the same capsule layers and routing algorithm with our proposed hierarchical model, allowing for a meaningful performance comparison in terms of accuracy, robustness, and efficiency. On the other hand, the B-CNN model in [24] is a multi-label hierarchical classifier that addresses all the class labels within the datasets. This approach considers the hierarchical structure and explicitly captures the relationships between classes at different levels. By comparing our BUH-CapsNet with the B-CNN model, we aim to assess the effectiveness of our proposed method in improving hierarchical classification performance compared to a CNN-based hierarchical approach.

A. Experimental Setup Details

For all the experiments, we implement our BUH-CapsNet model on TensorFlow using the Adam optimizer with TensorFlow's default settings. In order to train our model, we use an exponential decay learning rate function with an initial value

[1]Hereinafter, we abuse the notion $i \in \{1, 2, 3, \cdots, N\}$ to represent all the N levels since it will not cause any confusion.

of 0.001, and a decay rate of 0.95 is carried out after 10 training epochs. This decay schedule helps in fine-tuning the model over time, allowing it to converge to a better solution.

As mentioned earlier, our BUH-CapsNet model has a feature extraction block and a primary capsule layer (P) for extracting local features and constructing inputs for the secondary capsule layers. For all the implementations of BUH-CapsNet, the feature extraction block consists of four sub-blocks comprised of two convolution layers where each followed by a batch normalization layer, and finally one max pooling layer. All the convolutional layers in the feature extraction block use 3×3 filters with zero padding and rectified linear unit $(ReLu)$ activations function. We gradually increase the number of filters in the convolutional layers from 32 filters for the two convolutional layers in the first sub-block to 512 in the following sub-blocks, i.e. 64, 128, 256 and 512. Likewise, for all the max pooling layers in the feature extraction block, we use a 2×2 pooling window with a stride of 2, resulting in downsampling by a factor of 2 in each pooling step. For modesty, we use 8-dimensional primary capsules (P) and 16-dimensional secondary capsules (S_i) in the BUH-CapsNet architecture for all the scenarios. Further, for training our BUH-CapsNet model we set the hyperparameter $m+$, $m-$ and ω to 0.9, 0.1 and 0.5 in Eq. (2) for all the experiments. To achieve a balanced learning process across different levels of the hierarchical label tree, we follow the approach presented in [25] and set the hierarchical level weights λ_i in Equation (1). These weights are adjusted after each training epoch, considering the level-wise hierarchical accuracy. This adjustment ensures that the model maintains a suitable balance between the different levels, promoting effective learning and classification performance.

B. Datasets

In order to train and test our BUH-CapsNet model for hierarchical multi-label classification problems, we have manually assigned additional coarse and medium classes to the aforementioned datasets to make a tree-like hierarchical structure. In this tree structure, coarse classes are a superclass of all the corresponding medium-level classes, and both of these labels are a superclass for the fine labels. Therefore, each image will have multiple class labels that follow a hierarchical tree, where the number of class labels gradually increases from coarse to fine levels. Table I provides an overview of the datasets used in our experiments, along with the details of their hierarchical levels, the number of training samples, and the number of testing samples.

The EMNIST [19] dataset contains 28×28 grey-scale handwritten images of letters and numbers, which belongs to 47 fine classes. We assign two additional coarse labels, i.e. digit and letter classes, as mentioned in [25] to make a hierarchical structure. This hierarchical structure enables us to capture the hierarchy between digits and letters within the dataset. The Fashion-MNIST [20] dataset also contains 28×28 grey-scale fashion images for 10 classes. Following the steps in [26], we create a three-level hierarchy for the dataset. We assign coarse, medium, and fine labels to create a hierarchical structure that reflects the taxonomy of fashion items.

Dataset	Hierarchical levels	Training samples	Testing samples
EMNIST	2 (Coarse and Fine)	112,800	18,800
FMNIST	3 (Coarse, Medium and Fine)	60,000	10,000
CIFAR-10	3 (Coarse, Medium and Fine)	50,000	10,000
CIFAR-100	3 (Coarse, Medium and Fine)	50,000	10,000
CU_Bird	3 (Coarse, Medium and Fine)	5,944	2,897
Stanford Cars	3 (Coarse, Medium and Fine)	8,144	4,021

TABLE I: Description of datasets used in the experiments. Each dataset is divided into hierarchical levels. Each sample in the dataset is annotated with a label at each level.

For the CIFAR-10 and CIFAR-100 datasets [21], as well as the Caltech Birds-200-2011 [22] and Stanford Cars [23] datasets, we follow the hierarchical label structures proposed in prior works [24], [27]. These datasets are transformed into three-level hierarchical structures, allowing us to model the relationships between classes at different levels within the hierarchical label tree. This hierarchical structure allows us to model the relationships between classes at different levels in the hierarchical label tree. By incorporating these hierarchical label structures into the datasets, we aim to evaluate the effectiveness of BUH-CapsNet in capturing and leveraging the hierarchical relationships between classes for improved multi-label image classification performance.

C. Results

Now we draw our attention to the results yielded by our BUH-CapsNet, the CapsNet as initially proposed in [13] and the B-CCN approach presented in [24] when implementing the aforementioned datasets. In Figure 2, we present the accuracy plots for the fine level classification of all the datasets, showcasing the performance of different classifiers as a function of training epoch. We focus on the fine level since it represents the most challenging classification task within the hierarchy, requiring the identification of more complex features and encompassing the highest number of class labels. It is evident from the plots that our proposed BUH-CapsNet consistently outperforms the alternative classifiers in terms of accuracy, demonstrating its superior capability in handling hierarchical multi-label classification tasks. Moreover, the BUH-CapsNet model exhibits faster convergence, reaching higher accuracy levels within fewer training epochs compared to the baseline models.

To provide a comprehensive assessment of the classifiers' performance, we utilize additional metrics beyond the traditional precision, recall, and F1-score, which may overlook the hierarchical relationships between the classes. Instead, we adopt hierarchical metrics to evaluate the models, considering

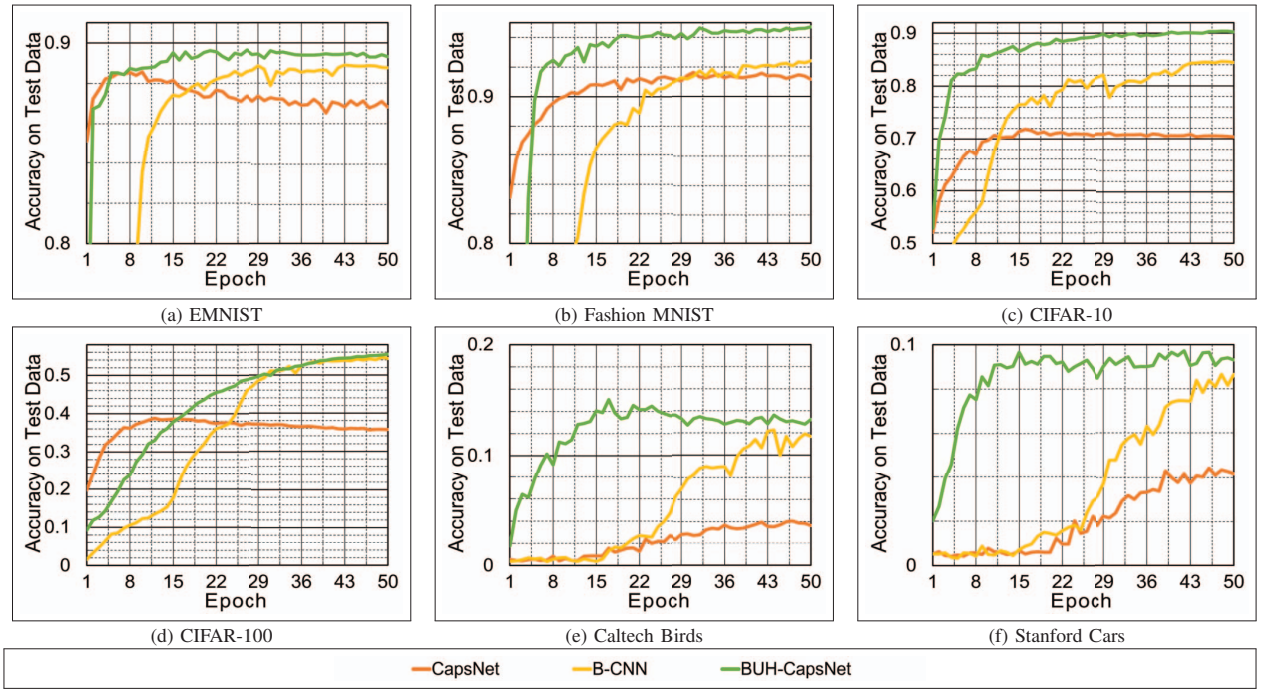

Fig. 2: Fine-level accuracy on the test dataset as a function of training epoch for all the models considered. Each plot represents a different dataset.

the hierarchical precision, hierarchical recall, hierarchical F1-score [28], consistency [29], and exact match score [29]. Fig. 3 presents the overall performance of the classifiers on all the datasets, providing insights into their effectiveness in capturing and utilizing the hierarchical relationships within the data. By leveraging the hierarchical metrics, we obtain a more comprehensive evaluation of the models, accounting for the hierarchical nature of the classification task. The results depicted in Fig. 3 further reinforce the superiority of our BUH-CapsNet model, showcasing its consistently superior performance across all the hierarchical metrics. This reinforces the notion that our proposed model not only surpasses the alternative classifiers in terms of accuracy but also demonstrates its effectiveness in capturing the hierarchical relationships between classes, resulting in improved hierarchical classification performance. This improvement is particularly evident in datasets with more complex features, such as CIFAR-100, Caltech Birds-200-2011, and Stanford Cars. Furthermore, our network consistently achieves higher hierarchical consistency, which is reflected in the hierarchical exact match metric. This outcome is expected since BUH-CapsNet first classifies the fine classes and then utilizes the obtained information to classify the parent classes (coarse and medium classes), thereby enforcing hierarchical consistency. This characteristic is evident in the results obtained across all the datasets, further reinforcing the efficacy of BUH-CapsNet in capturing and leveraging the hierarchical relationships within the data.

It is also worth mentioning that, compared to baseline CapsNet [13], our proposed BUH-CapsNet achieved a much higher classification accuracy, as presented in Fig. 2 and Fig. 3. The superior performance of BUH-CapsNet validates the effectiveness of our hierarchical approach in capturing and leveraging the hierarchical relationships between classes. In Figure 3, we also present additional performance metrics for the baseline CapsNet [13], focusing solely on the fine level of the dataset since it is a flat classifier and does not consider the hierarchical structure. However, our proposed BUH-CapsNet takes full advantage of the hierarchical relationships, resulting in improved performance across all levels of the hierarchy.

Furthermore, we provide insights into the model complexity and efficiency by comparing the number of trainable parameters for different models in Table II. Our BUH-CapsNet exhibits a significant reduction in the number of trainable parameters compared to the baseline CapsNet [13]. This reduction is primarily attributed to the differences in the architecture and design choices between the two models. The baseline CapsNet utilizes convolutional layers to form the primary capsules, which leads to a larger number of trainable parameters. In contrast, our BUH-CapsNet leverages a dedicated feature extractor to form the primary capsules, resulting in a reduced parameter count. Furthermore, the baseline CapsNet employs a decoder network to reconstruct the input image, which adds additional trainable parameters to the model. In contrast, our BUH-CapsNet achieves performance improvements without the need for a decoder network, eliminating the associated trainable parameters. The reduction in the number of trainable

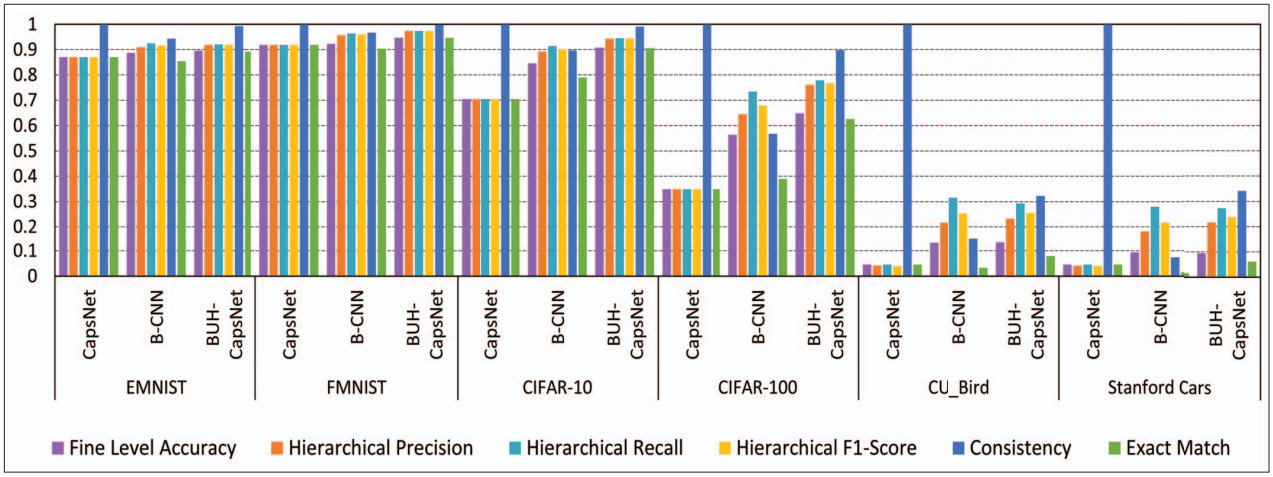

Fig. 3: Hierarchical performance of all the classifiers on the test datasets. Each section in the figure represents a dataset.

parameters offers several advantages. Also, it contributes to faster execution during training and inference, as the model has fewer parameters to update and compute. This can result in improved efficiency and reduced computational costs.

Furthermore, we provide insights into the model complexity and efficiency by comparing the number of trainable parameters for different models in Table II. Our BUH-CapsNet exhibits a significant reduction in the number of trainable parameters compared to the baseline CapsNet [13]. This reduction is primarily attributed to the differences in the architecture and design choices between the two models. The baseline CapsNet utilizes convolutional layers to form the primary capsules, which leads to a larger number of trainable parameters. In contrast, our BUH-CapsNet leverages a dedicated feature extractor to form the primary capsules, resulting in a reduced parameter count. Furthermore, the baseline CapsNet employs a decoder network to reconstruct the input image, which adds additional trainable parameters to the model. In contrast, our BUH-CapsNet achieves performance improvements without the need for a decoder network, eliminating the associated trainable parameters. The reduction in the number of trainable parameters offers several advantages. Also, it contributes to faster execution during training and inference, as the model has fewer parameters to update and compute. This can result in improved efficiency and reduced computational costs.

In our comparative analysis, we observed that BUH-CapsNet, in comparison to the B-CNN approach [24], exhibits slightly higher trainable parameters when implemented on the EMNIST, Caltech Birds-200-2011 and Stanford Cars datasets. This difference in parameter count is primarily attributed to the design of the feature extractor in BUH-CapsNet. However, despite having a slightly higher parameter count, BUH-CapsNet consistently outperforms the CNN approach, as depicted in Fig. 2 and Fig. 3. BUH-CapsNet demonstrates superior performance compared to the alternative models in multiple aspects. Particularly, when applied to datasets with a limited number

Dataset	Trainable Parameters (in Millions)		
	CapsNet [13]	B-CNN [24]	BUH-CapsNet
EMNIST	13.97	0.87	5.10
FMNIST	8.22	9.41	4.79
CIFAR-10	17.39	12.38	5.04
CIFAR-100	44.66	12.48	8.52
CU_Bird	105.72	31.52	38.43
Stanford Cars	104.08	31.50	36.43

TABLE II: Total number of trainable parameters for the classification models on all the datasets.

of training samples, such as Caltech Birds-200-2011 and Stanford Cars, BUH-CapsNet showcases significantly higher model performance. This improvement in performance can be attributed to the hierarchical nature of BUH-CapsNet, which leverages the hierarchical relationships between classes to enhance classification accuracy.

IV. Conclusion

In this work, we proposed a hierarchical capsule network employing a bottom-up approach for image classification. The network uses multiple secondary capsule layers to predict hierarchical class labels following a hierarchical label tree. One key aspect of our approach is the utilization of the bottom-up strategy, which facilitates the propagation of information between the secondary capsule layers. This bottom-up information flow is aligned with the inherent hierarchy present in the data, enabling the network to capture and leverage the hierarchical relationships between classes. By leveraging the data hierarchy, our model achieves improved consistency and maintains a balanced learning process across different levels of the hierarchical label tree. Through extensive experiments conducted on six widely available and diverse datasets, we demonstrated the superior performance of our BUH-CapsNet model compared to alternative approaches. In particular, when

compared to the baseline capsule network, our proposed model exhibited significantly better classification performance while achieving faster convergence rates. These findings highlight the effectiveness and efficiency of our BUH-CapsNet in hierarchical image classification tasks. By combining the power of capsule networks, the utilization of hierarchical label structures, and the bottom-up strategy, our proposed approach contributes to the advancement of hierarchical multi-label classification methods. The promising results obtained from our experiments validate the potential of BUH-CapsNet as a robust and efficient model for addressing complex hierarchical classification problems in various domains.

REFERENCES

[1] E.-J. Cheng, K.-P. Chou, S. Rajora, B.-H. Jin, M. Tanveer, C.-T. Lin, K.-Y. Young, W.-C. Lin, and M. Prasad, "Deep Sparse Representation Classifier for facial recognition and detection system," *Pattern Recognition Letters*, vol. 125, pp. 71–77, Jul. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865519300868

[2] J. Brownlee, *Deep learning for computer vision: image classification, object detection, and face recognition in python*. Machine Learning Mastery, 2019.

[3] L. Xie, F. Lee, L. Liu, K. Kotani, and Q. Chen, "Scene recognition: A comprehensive survey," *Pattern Recognition*, vol. 102, p. 107205, Jun. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003132032030011X

[4] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and 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.

[5] P. Panda and K. Roy, "Semantic driven hierarchical learning for energy-efficient image classification," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, Mar. 2017, pp. 1582–1587, iSSN: 1558-1101.

[6] R. M. Pereira, D. Bertolini, L. O. Teixeira, C. N. Silla, and Y. M. G. Costa, "COVID-19 identification in chest X-ray images on flat and hierarchical classification scenarios," *Computer Methods and Programs in Biomedicine*, vol. 194, p. 105532, Oct. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260720309664

[7] D. Li, A. Cong, and S. Guo, "Sewer damage detection from imbalanced CCTV inspection data using deep convolutional neural networks with hierarchical classification," *Automation in Construction*, vol. 101, pp. 199–208, May 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580518306174

[8] D. Gao, W. Yang, H. Zhou, Y. Wei, Y. Hu, and H. Wang, "Deep Hierarchical Classification for Category Prediction in E-commerce System," May 2020, 7 citations (Semantic Scholar/arXiv) [2022-11-02] arXiv:2005.06692 [cs]. [Online]. Available: http://arxiv.org/abs/2005.06692

[9] C. Shi, Z. Lv, X. Yang, P. Xu, and I. Bibi, "Hierarchical Multi-View Semi-Supervised Learning for Very High-Resolution Remote Sensing Image Classification," *Remote Sensing*, vol. 12, no. 6, p. 1012, Jan. 2020, number: 6 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2072-4292/12/6/1012

[10] C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1, pp. 31–72, Jan. 2011. [Online]. Available: https://doi.org/10.1007/s10618-010-0175-9

[11] A. Naik and H. Rangwala, "Inconsistent Node Flattening for Improving Top-Down Hierarchical Classification," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct. 2016, pp. 379–388.

[12] L. Zhang, S. K. Shah, and I. A. Kakadiaris, "Hierarchical Multi-label Classification using Fully Associative Ensemble Learning," *Pattern Recognition*, vol. 70, pp. 89–103, Oct. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317301899

[13] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic Routing Between Capsules," *Advances in neural information processing systems*, vol. 30, Oct. 2017. [Online]. Available: https://arxiv.org/abs/1710.09829v2

[14] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," in *International conference on learning representations*, 2018. [Online]. Available: https://openreview.net/pdf?id=HJWLfGWRb

[15] M. Kwabena Patrick, A. Felix Adekoya, A. Abra Mighty, and B. Y. Edward, "Capsule Networks – A survey," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 1, pp. 1295–1310, Jan. 2022, publisher: Elsevier. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157819309322

[16] E. Xi, S. Bing, and Y. Jin, "Capsule Network Performance on Complex Data," Dec. 2017, arXiv:1712.03480 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1712.03480

[17] K. T. Noor, A. Robles-Kelly, and B. Kusy, "A Capsule Network for Hierarchical Multi-label Image Classification," in *Structural, Syntactic, and Statistical Pattern Recognition*, ser. Lecture Notes in Computer Science, A. Krzyzak, C. Y. Suen, A. Torsello, and N. Nobile, Eds. Cham: Springer International Publishing, 2022, pp. 163–172.

[18] J. Rajasegaran, V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, and R. Rodrigo, "DeepCaps: Going Deeper With Capsule Networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, Jun. 2019, pp. 10 717–10 725. [Online]. Available: https://ieeexplore.ieee.org/document/8953957/

[19] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 2921–2926, iSSN: 2161-4407.

[20] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," Sep. 2017, arXiv:1708.07747 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1708.07747

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

[22] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Jul. 2011, issue: 2010-001 Num Pages: 8 Number: 2010-001 Place: Pasadena, CA Publisher: California Institute of Technology. [Online]. Available: https://resolver.caltech.edu/CaltechAUTHORS:20111026-120541847

[23] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D Object Representations for Fine-Grained Categorization," in *2013 IEEE International Conference on Computer Vision Workshops*. Sydney, Australia: IEEE, Dec. 2013, pp. 554–561. [Online]. Available: http://ieeexplore.ieee.org/document/6755945/

[24] X. Zhu and M. Bain, "B-CNN: branch convolutional neural network for hierarchical classification," *arXiv preprint arXiv:1709.09890*, 2017.

[25] K. T. Noor and A. Robles-Kelly, "H-Capsnet: A Capsule Network for Hierarchical Image Classification," Rochester, NY, Nov. 2022. [Online]. Available: https://papers.ssrn.com/abstract=4271318

[26] Y. Seo and K.-s. Shin, "Hierarchical convolutional neural networks for fashion image classification," *Expert Systems with Applications*, vol. 116, pp. 328–339, Feb. 2019, publisher: Elsevier. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417418305992

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

[28] A. Kosmopoulos, I. Partalas, E. Gaussier, G. Paliouras, and I. Androutsopoulos, "Evaluation measures for hierarchical classification: a unified view and novel approaches," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 820–865, May 2015. [Online]. Available: http://link.springer.com/10.1007/s10618-014-0382-x

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