Hierarchical Multi-Label Open-Set Classification

Chun-Hua Lin
cl4335@columbia.edu

***Abstract*—This study addresses the challenging problem of hierarchical multi-label classification, a specific case in machine learning where each sample has more than one label, and all these labels are hierarchically structured. We leveraged pre-trained models and novel neural network architectures such as ViT, and HMCN, combined and designed to optimize both local and global loss functions. The architectures aim to capture local hierarchical class relationships and global information across the entire class hierarchy. The proposed method is evaluated on the pre-defined dataset in the context of image classification tasks. Notably, our evaluation includes previously unseen novel classes during training, presenting zero-shot learning scenarios. The proposed method outperforms the baselines. In addition, we illuminate the challenges and obstacles encountered during the experimentation process.**

I. Introduction

Hierarchical Multi-Label Classification (HMC) poses a complex challenge, as labels are not mutually exclusive, and objects are labeled to both a superclass and its corresponding subclasses—either all subclasses or a subset, depending on the task. The hierarchical structure, representing interconnected classification objectives, can be likened to either a tree or a Directed Acyclic Graph [1]. HMC has found widespread application in text classification and image annotation; where images are categorized into hierarchical classes. Furthermore, bioinformatics tasks, particularly protein function prediction, present intricate classification problems addressed by HMC methodologies.

Open-set (OS) recognition refers to a scenario where the model encounters samples during inference that do not belong to any of the classes it was exposed to during training. In other words, the model is faced with samples from classes that were not part of its training data. It acknowledges the presence of ”unknown unknowns” or novel classes that were unforeseen during the model's training phase, in contrast to a closed-set condition, where the model assumes that all test samples fall into one of the predefined classes [2]. An open-set condition recognizes the real-world complexity where new, previously unseen classes may emerge. This situation is particularly relevant in dynamic environments where the data distribution can change over time, leading to the introduction of novel classes that the model has not been trained on.

This study focuses on a dataset characterized by three overarching super-classes: bird, dog, and reptile, each hosting its own set of sub-classes. However, the challenge arises from the distributional disparity between the training and test datasets, compounded by the appearance of novel super-classes and sub-classes in the test dataset that were absent during the training phase. Consequently, this sets the stage for a Hierarchical Multi-Label Open-Set Classification task. The primary goal is to accurately predict both the super-class label and the corresponding sub-class label within each category for every image, including entities that have not been encountered previously.

Recognizing the pivotal role of transfer learning in enhancing model efficiency and accuracy, we leverage pre-trained models as part of the architecture. This enables us to capitalize on the knowledge acquired from a different but related task, providing a robust foundation for extracting image embeddings for the intricacies of our multi-label image classification challenge. Furthermore, we acknowledge the importance of data augmentation in refining the generalization capabilities of our model. Augmentation techniques play a critical role in expanding the diversity of our training dataset, ensuring that our model is well-equipped to handle the inherent variations in somewhat real-world scenarios in this study.

II. Related Work

A. ResNet

ResNet, short for Residual Network, is a pivotal architectural innovation in deep learning and convolutional neural networks (CNNs) [3]. ResNet was designed to address challenges associated with training very deep neural networks. The fundamental idea behind ResNet is the introduction of residual connections, also known as skip connections or shortcut connections. Traditional deep networks faced difficulties in training as their depth increased, often leading to issues like vanishing gradients, making it difficult to learn meaningful representations in earlier layers. ResNet tackles this problem by introducing skip connections, which allow the network to transmit information directly from one layer to deeper layers. By learning the residual mapping, which represents the difference between the input and output of a particular layer, the challenges associated with training deep architectures are mitigated.

The residual blocks enable ResNet to effectively learn and represent complex features in a hierarchical manner. As a result, ResNet architectures can extend to hundreds or even thousands of layers while maintaining efficient training and avoiding degradation issues. This has made ResNet a cornerstone in various computer vision tasks.

B. ViT

Vision Transformer (ViT) is a groundbreaking architecture in the field of computer vision that has significantly advanced image processing adopting a transformer-based structure [4]. ViT revolutionizes traditional convolutional neural network

(CNN) approaches by employing transformer architecture, originally popularized in natural language processing tasks.

ViT divides an image into fixed-size non-overlapping patches, treating each patch as a ”token” similar to words in natural language processing. These patches are then embedded in the feature space and positionally encoded, allowing ViT to capture global relationships and dependencies within the image. The final embeddings are then passed into the self-attention blocks. The attention mechanism in transformers enables ViT to consider interactions between all patches, enhancing its ability to recognize complex patterns and long-range dependencies.

C. CLIP

CLIP, short for Contrastive Language-Image Pre-Training [5], is a multi-modal vision and language model designed to be used for image-text similarity and zero-shot image classification. The model comprises two primary components: the text encoder and the image encoder. For image encoders, the CLIP model leverages both ResNet and ViT architectures. The text encoder, on the other hand, is implemented using a Transformer.

CLIP is able to effectively map images and text to the same feature space. CLIP utilizes the image encoder to generate image embeddings and the text encoder to obtain text embeddings. The embeddings are then projected to a latent space with identical dimensions. The dot product between the projected image and text features is then used as a similar score. The baseline used for this study is CLIP/B-32. B-32 means the ”Base (B)” size ViT model with an image patch size of 32. The larger the patch size.

D. MultiNet

MultiNet [6] introduces an efficient and effective feed-forward architecture. This architecture is designed to concurrently reason about semantic segmentation, image classification, and object detection. MultiNet features a shared encoder across all tasks and incorporates three branches, each implementing a dedicated decoder for a specific task. The joint training implementation includes independent forward passes for examples related to each of the three tasks, and added gradients during the back-propagation steps. This approach allows distinct training parameters for each decoder.

E. HMCN

HMCN [1] (Hierarchical Multi-Label Classification Networks) is designed for hierarchical multi-label classification. Two versions of HMCN were presented: HMCN-F, a robust feedforward variant with increased parameters, and HMCN-R, a more efficient recurrent version utilizing shared weights and an LSTM-like structure for hierarchical information encoding.

Algorithms designed for HMC assign labels to objects within the class hierarchy, enabling association with one or multiple paths. To achieve this, these algorithms must optimize a loss function either locally or globally. Local learning algorithms take an approach by discovering features that dictate class relationships in specific layers of the class hierarchy. Then, these algorithms combine the locally derived predictions to generate the final classification, utilizing a top-down strategy to form a hierarchy of classifiers. Each classifier within this hierarchy is responsible for predicting particular nodes or specific hierarchical levels. Conversely, global approaches employ a single classifier capable of associating objects with their corresponding classes across the entire hierarchy. The decision between global and local approaches presents a trade-off. Global methods are generally more cost-effective and avoid the well-known error-propagation problem, although there’s a risk of potential underfitting due to the oversight of localized information within the hierarchy. On the other hand, local approaches, while more computationally intensive with their cascade of classifiers, excel in extracting nuanced information from specific regions of the class hierarchy, potentially leading to overfitting.

HMCN is the first hierarchical multi-label classification approach to seamlessly integrate both local and global information, simultaneously penalizing hierarchical violations.

F. Dynamic Intra-Class Splitting

Dynamic Intra-class Splitting (DICS) is a novel deep learning approach designed to address open-set recognition challenges proposed in [7]. In open set recognition, the training dataset consists solely of samples from a limited number of known classes. During inference, the open set recognizer not only needs to accurately classify samples from known classes but also to appropriately reject samples from unknown classes. Conventional deep learning models, assuming a closed set environment, are unsuitable for this task due to their assumptions. To tackle open set recognition, various specialized approaches have been explored, including adaptations of support vector machines and state-of-the-art methods utilizing generated samples to model unknown classes such as autoencoders [8]. In contrast, DICS characterizes unknown classes through atypical subsets of training samples obtained via intra-class splitting (ICS), see Fig.1. ICS is a two-stage algorithm, including the extraction of the atypical subset of the training samples and the training step. On the other hand, DICS is a one-stage method where ICS is performed dynamically epoch by epoch. Therefore, only one neural network must be trained.

III. Method

A. Dataset

The project utilized a training dataset comprising 6321 images, each with a resolution of 32×32 pixels, categorized into classes such as bird, dog, or reptile. Each corresponding class had its own discrete set of subclasses, further categorizing the images into more distinctive labels. The test set was composed of 12376 images of the same resolution but included images that did not fit into the predetermined classes or subclasses. Thus, these images could be labeled with one of the classes from the training data, but may not fit with any of the subclasses, or they may not belong to any of the overarching superclasses at all. In any of these cases, the

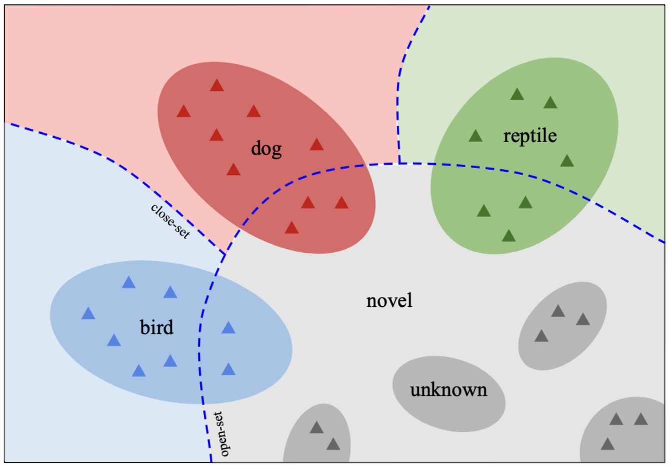

Fig. 1. Basic visualization of ICS on this study's dataset: Split given data from K known classes into $K + 1$ classes where the $K + 1$-class classifier is expected to reject samples from unknown (novel) classes. Here for our superclasses, $K = 3$ including bird, dog, and reptile; subclasses are represented by the triangles within the superclasses. The additional class combines all atypical subsets from each superclass. Compared to close-set boundaries, the $K + 1$-class classifier learns the open-set boundary to distinguish novel classes.

model was directed to classify the unfitting image into a novel superclass and/or subclass.

B. Data Augmentation

To address the distributional disparity between the training and test datasets and the appearance of novel super-classes and sub-classes in the test dataset, we utilize data augmentation techniques to transform the existing training images. We aim to through augmentation, help improve the model's generalization and robustness. The following augmentation was used for the experiment:

- Random Horizontal Flip: This augmentation randomly flips images horizontally. It is effective in scenarios where the orientation of an object in an image does not impact its classification.
- Random Vertical Flip: Similar to horizontal flip, random vertical flip randomly flips images vertically. This is useful when the vertical orientation of objects in the image does not carry specific semantic meaning for the task.
- Random Rotate: Random rotation involves rotating the image by a random angle. This augmentation is beneficial for tasks where the orientation of objects is significant, helping the model recognize objects from various viewpoints.
- Color Jittering: Color jittering involves applying random changes to the brightness, contrast, saturation, and hue of the image. This augmentation helps the model become less sensitive to variations in lighting conditions and color distributions, making it more robust across different environments.
- Random Erase: Random erase is a technique where a random part of the image is deliberately erased or replaced with random noise. This helps the model become more robust to occlusions and encourages it to focus on the remaining informative parts of the image.

C. ResNet-Only Model

The purpose of this simplistic model was to test the ResNet metric on the dataset for the purpose of employing it in later model combinations. The version of ResNet utilized for this project was ResNet101, primarily selected for its robust feature extraction capabilities. In addition, the utilization of Resnet101 theoretically allows the model to obtain more accurate predictions while simultaneously lowering the training time of the model, by leveraging transfer learning to train the weights of the ResNet101 network on the given dataset.

The ResNet-only model was designed by adapting a pre-existing ResNet101. The model was implemented by removing the head of the original model and replacing it with two different heads, one for predicting the sub-classes and one for predicting the super-classes. During training, all weights were frozen except those belonging to the two new heads.

D. ViT-Only Model

For this project, we also fine-tune and test the ViT model on the dataset and explore its potential integration into future model combinations. The transformer-based nature of ViT, coupled with its self-attention mechanisms, offers a unique approach to feature extraction. The chosen ViT variant, such as ViT-B/16 (Base model with a patch size of 16) and ViT-B/32, was employed due to its well-balanced trade-off between model size and computational efficiency. Similar to ResNet101, we leverage transfer learning, initializing its weights with pre-trained knowledge from a large-scale dataset. This approach not only facilitates effective feature learning but also contributes to faster convergence during training.

The VIT-only model was designed by adapting a VIT_B_16. The model was implemented by removing the head of the original model and replacing it with two different heads, one for predicting the sub-classes and one for predicting the super-classes. During training, all weights were frozen except those belonging to the two new heads.

E. Hierarchical Multi-Label Open-Set Classifier

Our proposed method, Hierarchical Multi-Label Open-Set Classifier (HMLOSC), draws inspiration from various works [1], [3], [6], [7]. In addressing the open-set multi-label hierarchical classification problem, we designed HMLOSC with a hierarchical structure akin to the HMCN-F algorithm [1]. HMLOSC integrates one global output and two local outputs, corresponding to each level of the hierarchy (i.e., superclasses and subclasses). This hierarchical architecture enhances the model's ability to learn inherent patterns in labeled data.

The information flows in two directions: he global flow initiates at the encoder, where features are extracted, and traverses through fully connected (FC) layer blocks until

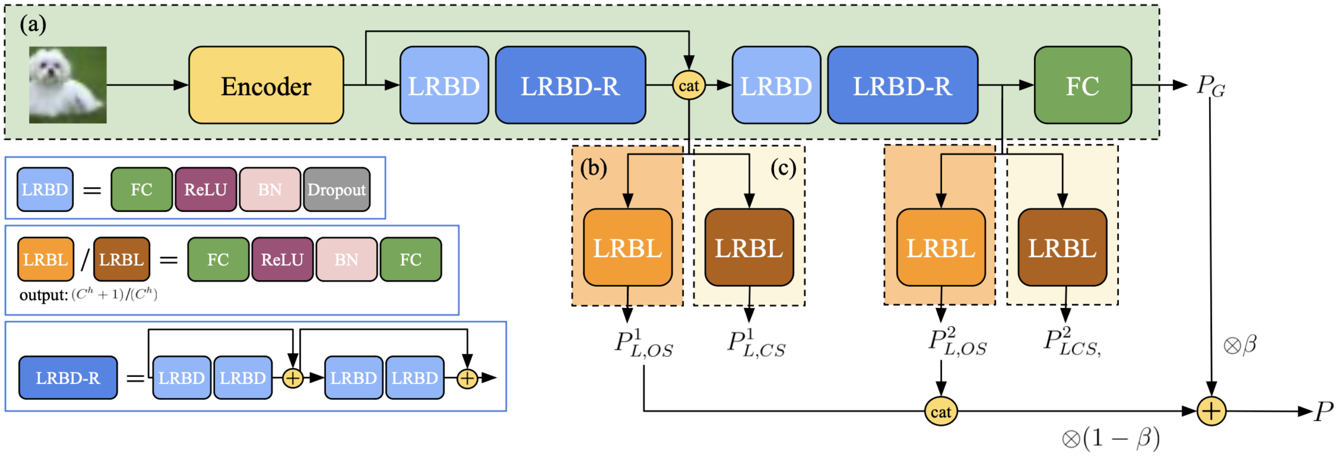

Fig. 2. The architecture of the HMLOSC network: (a) The global flow initiates with feature extraction using an encoder of choice (ViT), followed by traversal through global FC blocks (LRBD, LRBD-R), and the final fully connected layer. (b) The local flow diverges from the main flow and proceeds through a local FC block (LRBL). (c) A distinct branch of the local layer is responsible for predictions without considering the novel class for close-set predictions.

reaching the global output. Simultaneously, the local flows starts at the encoder, pass through the global FC layer blocks, and then diverge into their dedicated local FC layer blocks, concluding at the respective local outputs. The main flow benefits from the reuse of original input features in each layer, improving the learning of local information. This allows each layer to specialized in finding relationships between the original features and the information required for classifying classes in a given hierarchical level.

To generate the final prediction, all local outputs are concatenated and pooled with the global output for a consensual prediction. In an effort to expedite training and ensure network robustness, the global FC layer blocks incorporate shortcuts inspired by [3].

The global flow plays a pivotal role in transmitting information from the current hierarchical level (ith level) to the subsequent level (i + 1) in the hierarchy. This global flow is dynamically shaped by the local outputs, reinforcing level-wise relationships within the overarching information flow. The reinforcement is achieved through the backpropagation of gradients specific to the set of classes associated with each hierarchical level. For a visual representation, refer to Fig.2, which provides a graphical depiction of the HMLOSC structure.

In an extension of our fundamental architecture, an additional local FC CS (Close-Set) layer block is introduced at each local level, distinct from the original local FC block, specifically tailored for close-set predictions. This additional local FC CS layer is only used during training and is responsible for predicting the correct classes for each training sample. Subsequently, the sample with the least confident prediction, indicated by the lowest probability, is reassigned to a new ”novel” class for training the primary architecture.

The loss function is designed to minimize the sum of the global, local, and close-set loss functions. The loss function allows prioritizing either open-set loss (global loss + local loss) or close-set loss, offering adaptability based on specific optimization requirements.

Formally, let $x \in \mathbb{R}^{|D|\times 1}$ be the extracted input feature vector, $|D|$ be the number of features, C^h be the set of classes of the h^{th} hierarchical level, $|C|$ the total number of classes, and $|H|$ the total number of hierarchical levels.

Let A_G^1 denote the activations in the first level of the global flow (first level of the class hierarchy) and is given by:

$$A_G^1 = \phi(W_G^1 x + b_G^1) \tag{1}$$

where the weight $W_G^1 \in \mathbb{R}^{|A_G^1|\times|D|}$ and the bias $b1G \in \mathbb{R}^{|A_G^1|\times 1}$ are the parameters for learning global information directly from the input, and ϕ is a non-linear activation function (e.g., ReLU). The subsequent global activations are given by:

$$A_G^h = \phi(W_G^h(A_G^{h-1} \odot x) + b_G^h) \tag{2}$$

where $\odot$ denotes vector concatenation. Finally, The HMC prediction based on global information $\hat{y}_G \in \mathbb{R}^{|C|\times 1}$ is then calculated by:

$$P_G = \sigma(W_G^{|H|+1} A_G^{|H|} + b_G^{|H|+1}) \tag{3}$$

where $WG \in \mathbb{R}^{|C|\times|Ah|}$ is the final layer weight matrix that has $|C|$ neurons, and σ is the sigmoid activation.

With respect to the local flows, let A_L^h denote the activations in the h^{th} layer of the local flow, which is calculated by:

$$A_L^h = \phi(W_T^h A_G^h) + b_T^h \tag{4}$$

where $W_T^h \in \mathbb{R}^{|A_L^h|\times|A_G^h|}$ is a transition weight matrix that maps a global hidden layer to a local hidden layer, and $b_T^h \in \mathbb{R}^{|v|\times 1}$ is the transition bias vector. Hence, the local predictions for level h, $\hat{y}_L^h \in \mathbb{R}^{|C^h|}$, are given by:

TABLE I
HMC MODELS PERFORMANCE BY ACCURACY(%)

	HMCN(384)	HMCN(768)	HMCN(1536/B64)	HMCN(1536/B32)	Deep-HMCN(1536/B32)	Deep-HMLOSC	HMLOSC
Superclass	69.14	69.26	69.34	68.93	69.18	**72.81**	71.01
Subclass	13.04	14.33	14.86	15.67	15.12	50.54	**59.62**

TABLE II
BASELINE AND ENCODER-ONLY PERFORMANCE BY ACCURACY(%)

	CLIP	CNN	ResNet101	ViT-B/16
Superclass	35.33	43.72	33.31	**69.35**
Subclass	7.89	2.05	0.43	**16.13**

TABLE III
DATA AUGMENTATION PERFORMANCE BY ACCURACY(%)

Deep-HMLOSC	All	No Erase	None
Superclass	70.05	**75.88**	74.16
Subclass	31.62	37.35	**38.73**

$$P^h_{L,OS} = \sigma(W^h_L A^h_L + b^h_L) \tag{5}$$

In addition, the close-set predictions for level h, $\hat{y}^h_{L,CS} \in \mathbb{R}^{|C^h|}$, are given by:

$$P^h_{L,CS} = \sigma(W^h_{L,CS} A^h_L + b^h_{L,CS}) \tag{6}$$

1) Dynamic Intra-class Splitting: Consider each sample as $x_i \in \chi = \{x_1, x_2, ..., x_N\}$ has an individual class label $y_{CS,i} \in \{0, 1, ..., C^h\}$. After the e^th training epoch, the score $s_i^{(e)}$ of an input sample x_i is denoted as follows:

$$s_i^{(e)} = \begin{cases} P^h_{L,CS}(y_i|x_i) & \text{if } \hat{y}^h_{L,CS,i} = y_{CS,i} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Where $\hat{y}^h_{L,CS,i}$ is the close-set predicted class for x_i at the h^th local level. For each training epoch (e), scores for all training samples are collected into a score set $S^{(e)} = \{s_1^{(e)}, s_2^{(e)}, \ldots, s_N^{(e)}\}$. Let a hyperparameter $0 < \rho < 1$ to represent the intra-class splitting ratio and $S_\rho^{(e)}$ as the ρ^{th} fraction of $S^{(e)}$ containing the lowest scores, $\max S_\rho^{(e)}$ serves as a threshold distinguishing between atypical and typical samples. Hence the novel label is assigned to atypical samples as follows:

$$y_{OS,i} = \begin{cases} y_{CS,i} & \text{if } P^h_{L,CS}(y_i|x_i) > \max S_\rho^{(e)} \\ y_{novel} & \text{otherwise} \end{cases} \tag{8}$$

2) Objective Function: Now let the global flow from Eq. 3 be denoted as $f_G(\cdot)$, the local flow from Eq. 5 be denoted as $f_{L,OS}(\cdot)$ and the close-set classifier from Eq. 8 be denoted as $f_{L,CS}(\cdot)$. Then, the objective of the proposed method is as follows:

$$\min_{f_g, f_{L,OS}, f_{L,CS}} \begin{pmatrix} \mathbb{E}_{(x,y_G)} \mathcal{L}_G(f_G(x), y_G) \\ +\mathbb{E}_{(x,y_{L,OS})} \mathcal{L}_{L,OS}(f_{L,OS}(x), y_{L,OS}) \\ +\mathbb{E}_{(x,y_{L,CS})} \mathcal{L}_{L,CS}(f_{L,CS}(x), y_{L,CS}) \end{pmatrix} \tag{9}$$

where $\mathcal{L}_{L,OS}$ and $\mathcal{L}_{L,CS}$ are the learning objectives for $C^h + 1$ and C^h local classification problems, respectively.And $\mathcal{L}_G$ is the learning objective for $|C|$ global classification problem. In addition, the hyperparameter λ controls the trade-off between open-set loss (global loss + local loss) and close-set loss. In this work, the categorical entropy loss is used for all terms in the objective function.

3) Inference: Utilizing both local and global information, the final predictions during inference $P \in \mathbb{R}^{|C| \times 1}$ are calculated as follows:

$$P = \beta(P^1_{L,OS} \odot P^2_{L,OS} \odot ...P^{|H|}_{L,OS}) + (1 - \beta)P_G \tag{10}$$

where the hyperparameter $\beta \in [0, 1]$ regulates the trade-off between local and global information. By default, we set $\beta = 0.5$ to assign equal importance to both local and global information within the class hierarchy.

IV. RESULT

First, we compared the baseline models, CLIP and CNN, where CLIP outperformed CNN in the subclass task but lagged behind in the superclass task (refer to Table II). Interestingly, the ResNet-Only Model displayed the weakest performance, while the VIT-Only Model exhibited substantial improvement. Consequently, we opted for a VIT as the encoder for both the HMCN and HMLOSC models.

Although the VIT encoder paired with the HMCN head yielded only marginal improvements, sometimes even underperforming the VIT-Only Model, thorough experimentation with various global weight dimensions and batch sizes uncovered optimal settings (global weight dimension of 1536, batch size 32, and deeper global FC blocks). While this configuration enhanced the extraction of high-dimensional features, the resultant increase in classification accuracy was modest, remaining below 1% for both subclass and superclass predictions (refer to Table I).

However, an oversight in training set composition, lacking novel class samples for both super and subclasses, led to ineffective training of weights associated with the novel neuron. To address this issue, we introduced Dynamic Intra-class Splitting, culminating in the HMLOSC model. Notably, the super-class prediction accuracy on the test set improved to 0.71019, while the sub-class prediction demonstrated an even more substantial enhancement, concluding with a final accuracy of 0.59621 (see Table I).

In the domain of data augmentation, individual or combined methods, such as Random Horizontal Flip, Random Vertical Flip, Random Rotate, Color Jittering, and Random Erase, generally boosted the superclass performance. Surprisingly, the combination of all augmentations, except for Random Erase, performed exceptionally well in the superclass task, indicating enhanced generalizability for superclass predictions, albeit not as effective for subclasses. Notably, all augmentations have caused the performance of the subclass to plummet. We argue that the low resolution of the dataset might not be suitable for augmentations as it diminishes the available information of the image.

V. Conclusion

In conclusion, our comprehensive exploration of various model architectures and data augmentation strategies illuminated key insights into their impact on both subclass and superclass predictions. The superiority of CLIP over CNN in the subclass task and the nuanced performance variations with different model configurations underscore the importance of tailoring architectures to specific objectives. The introduction of Dynamic Intra-class Splitting in the HMLOSC model proved instrumental in addressing training set limitations, substantially improving accuracy in both superclass and subclass predictions. Future work could delve into refining the interplay between global and local information in our hierarchical models, potentially leveraging more advanced attention mechanisms or exploring novel data augmentation techniques. Additionally, exploring strategies to enhance the model's interpretability, scalability, and efficiency could contribute to its applicability across diverse domains. Furthermore, investigating the model's robustness to noisy or limited labeled data scenarios and extending its capabilities to handle dynamic class distributions in open-set scenarios represent promising avenues for future research.

References

[1] Wehrmann, J., Cerri, R. & Barros, R. Hierarchical Multi-Label Classification Networks. *Proceedings Of The 35th International Conference On Machine Learning*. **80** pp. 5075-5084 (2018,7,10), https://proceedings.mlr.press/v80/wehrmann18a.html

[2] Chuanxing Geng, Sheng-jun Huang and Songcan Chen. Recent Advances in Open Set Recognition: A Survey, 2018; arXiv:1811.08581. DOI: 10.1109/TPAMI.2020.2981604.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. Deep Residual Learning for Image Recognition, 2015; arXiv:1512.03385.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2020; arXiv:2010.11929.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision, 2021; arXiv:2103.00020.

[6] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla and Raquel Urtasun. MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving, 2016; arXiv:1612.07695.

[7] P. Schlachter, Y. Liao, and B. Yang, "Deep Open Set Recognition Using Dynamic Intra-class Splitting," *SN COMPUT. SCI.*, vol. 1, p. 77, 2020.

[8] Dor Bank, Noam Koenigstein and Raja Giryes. Autoencoders, 2020; arXiv:2003.05991.