



Problem 3: Hierarchical Group Robustness

Introduction:

Training deep learning models using Empirical Risk Minimization (ERM) on a dataset, poses the risk of relying on *spurious correlation*. These are correlations between certain patterns in the training dataset and the target, despite lacking any causal relationship. Learning such correlations as shortcuts can negatively impact models' accuracy on groups that are not presented enough and do not contain the spurious patterns associated with the target. This leads to the importance of the study of datasets that have unbalanced groups. We use a hierarchical dataset with multiple underrepresented groups to study the robustness of methods on different groups.

Dataset:

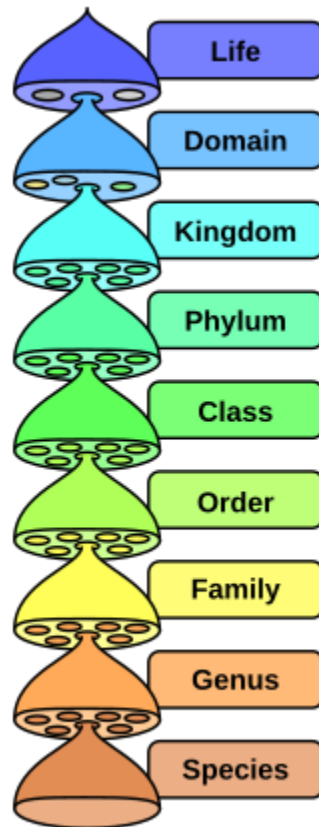
We use a modified version of the [iNat2021](#) as our target dataset. The iNat2021 dataset organizes all macroscopic living organisms under a [Biological Taxamoni](#) with 7 levels of **categorizations**: Kingdom, Phylum, Class, Order, Family, Genus, and Species. Our training set includes 100,000 images representing various species. The modifications made to the iNat dataset involve **secret** image-level transformations applied to both the training and test images and there is no distribution shift between train and test set. These transformations ensure a fair comparison of methods among participants.





RAYAN AI Contest

— First Phase Questions —



Living beings are categorized under a scientific taxonomy [1].





RAYAN AI Contest

— First Phase Questions —



A sample from [modified-inat](#).

Definitions:

Hierarchical Dataset: When your data is organized in a hierarchical structure (a tree), each data point has multiple labels corresponding to its predecessors in the hierarchy. For example, a whale is labeled as both a mammal and an animal, while a tulip is labeled as both a plant and a monocot. This means you can assign a label to an entity at every level of categorization in the hierarchy tree.





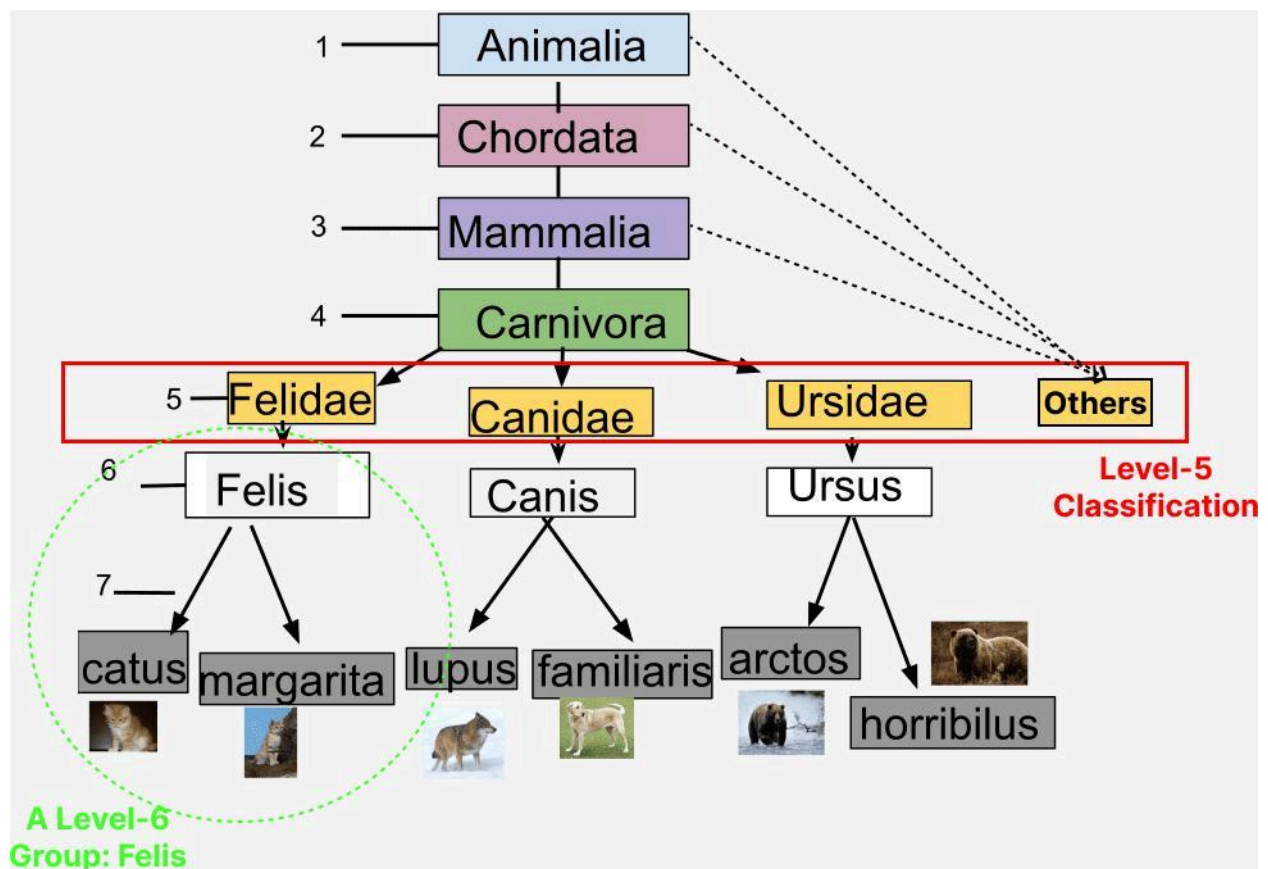
RAYAN AI Contest

— First Phase Questions —

A **Group** refers to all the data classified under a particular category. For example, all animals fall under the Animalia group, all mammals under the Mammalia group, and all dogs and wolves under the Dog (Canis) group. A group categorized at the n 'th layer of the hierarchy is called a **level- n group**.

The classification under the category of the predecessor in the n 'th layer is a **level- n classification** (where the 0'th layer represents all entities and the root of the tree).

Take a look at the image below for more clarification on the definitions.



Dataset Access

The training dataset is available through [RayanAi Hugging Face](#).





Problem Statement:

We have a hierarchical dataset (modified-iNat) that is unbalanced across its level-7 groups (the species groups in the iNat dataset). Your task is to train the **best level-2 classifier** and aim to achieve the best accuracy across all the species indifferently. You may only use **Resnet50** as your feature extractor architecture (The exact information about the architecture and the submission method is described thoroughly in the Model Architecture and Submission Format subsections). You can only train and submit the model weights (not the architecture). The exact evaluation method is described below.

Evaluation Metric:

The test set used to evaluate your method is balanced across 10,000 species, which correspond to level-7 groups. Your objective is to maximize the classification accuracy at level-2 for the test set, across all level-X groups. It's important to note that the classification level (level-2) is **different** from the group level (level-X) on which the accuracies are calculated.

To measure performance, we first identify the K groups with the lowest accuracies at level-X, then average these accuracies to produce the final metric. Therefore, achieving a high overall average accuracy is not the main focus.

For this problem we set $X=7$ and K is secret value between 10% and 50% of number of groups.

A simple ERM classifier, without any regard for problem characteristics, is provided in the notebook as a baseline.

Note: The class order is determined by concatenating the label of each node from the root to the node, and then sorting these concatenated labels alphabetically.

Model Architecture:

We use `torchvision.models.resnet50` as our base, but we modify the final linear classification layer. Instead of the default, we use `Linear(in_features=2048, out_features=13, bias=True)` for our specific task. Participants are provided with model weights that have been pretrained on the training set. The weights are available through





RAYAN AI Contest

— First Phase Questions —

[RayanAi Hugging Face](#).

Submission Format:

We use the specified architecture as the feature extractor. Specifically, you can only submit your model weights. Submitting your own model architecture is not permitted.

```
import zipfile
torch.save(model.state_dict(), 'resnet.pth')
with zipfile.ZipFile('submission.zip', 'w') as zipf:
    zipf.write('resnet.pth')
# Submit
```

References

[1] “Taxonomy (biology).” *Wikipedia*, [https://en.wikipedia.org/wiki/Taxonomy_\(biology\)](https://en.wikipedia.org/wiki/Taxonomy_(biology)). Accessed 19 September 2024.

Version: 1.4

