HIERMATCH: Leveraging Label Hierarchies for Improving Semi-Supervised Learning

Ashima Garg Shaurya Bagga Yashvardhan Singh Saket Anand

IIIT-Delhi, India

{ashimag, shaurya17104, yashvardhan17123, anands}@iiitd.ac.in

Abstract

Semi-supervised learning approaches have emerged as an active area of research to combat the challenge of obtaining large amounts of annotated data. Towards the goal of improving the performance of semi-supervised learning methods, we propose a novel framework, HIERMATCH*, a semi-supervised approach that leverages hierarchical information to reduce labeling costs and performs as well as a vanilla semi-supervised learning method. Hierarchical information is often available as prior knowledge in the form of coarse labels (e.g., woodpeckers) for images with fine-grained labels (e.g., downy woodpeckers or golden-fronted woodpeckers). However, the use of supervision using coarse-category labels to improve semi-supervised techniques has not been explored. In the absence of fine-grained labels,* HIERMATCH *exploits the label hierarchy and uses coarse class labels as a weak supervisory signal. Additionally,* HIERMATCH *is a generic-approach to improve any semi-supervised learning framework, we demonstrate this using our results on recent state-of-the-art techniques MixMatch and FixMatch. We evaluate the efficacy of* HIERMATCH *on two benchmark datasets, namely CIFAR-100 and NABirds.* HIERMATCH *can reduce the usage of fine-grained labels by 50% on CIFAR-100 with only a marginal drop of 0.59% in top-1 accuracy as compared to MixMatch.*

1. Introduction

Recent achievements in deep learning can largely be attributed to the use of vast collections of labeled data. Data annotation is often tedious and time-consuming, and many applications such as fine-grained visual classification (FGVC) [11, 33, 15] and medical diagnostics require inputs from field experts for data annotation, thus increasing the cost of annotation. Semi-supervised learning (SSL) methods [24, 2, 3, 19, 25] have emerged as a practical approach to overcome this costly requirement of supervised learning by leveraging vast quantities of unlabeled data along with limited supervision from labeled data. Acquiring large amounts of unlabeled data usually results in very little or no additional cost.

Existing knowledge databases organize data by grouping related categories. WordNet [21], a language dataset, categorizes the most generic object classes semantically in groups, which makes the hierarchical structure of labels easily accessible for most visual datasets. While recent SSL methods [24, 3] have made encouraging progress, the use of label hierarchies with SSL remains unexplored. As obtaining fine-grained labels typically needs expert knowledge, an effective way of reducing annotation cost is to make use of coarser-level supervision obtained from non-experts. For instance, in NABirds [26], a fine-grained dataset, `downy woodpeckers` and `golden-fronted woodpeckers` are different fine-grained classes belonging to the same coarse class `woodpecker`. A layperson can easily assign the label `woodpecker` as compared to annotating it with fine-grained labels that can be done only by subject experts. We argue that by leveraging the readily available label hierarchies, we can trade-off expensive finer-grained labels annotated by experts in favor of cheaper, coarser-grained labels from non-experts, and yet devise semi-supervised techniques that perform nearly as well as baselines that use a larger number of fine-grained labels.

Our claim builds upon the observation that all classes in a dataset do not differ equally from each other [4]. By judiciously using label hierarchies, it is possible to train SSL-based classifiers with stronger priors despite weaker supervision (in the form of coarser labels) and achieve comparable fine-grained accuracy. For example, consider a dataset shown in Figure 1 containing images of different kinds of `Flower` and `Fish` (Level 2 categories). The goal is to train a ConvNet based classifier for the Level 3 categories, i.e., *types of* `Flowers` and `Fish`. By explicitly using label

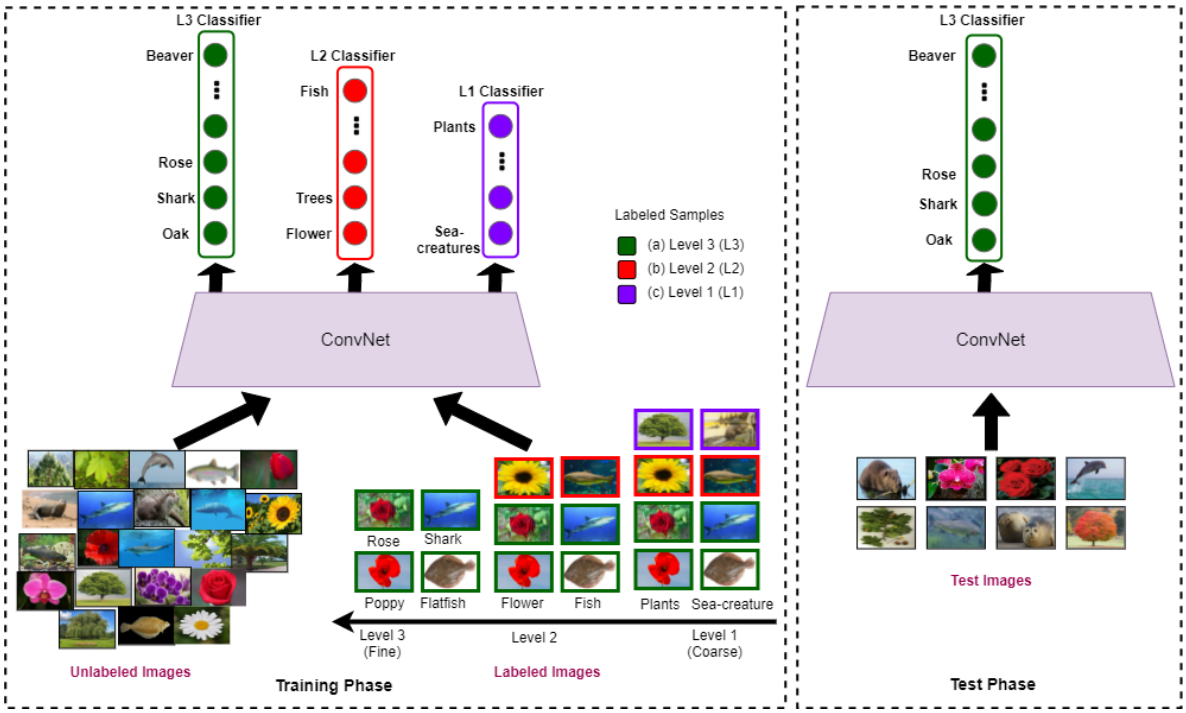

Figure 1. **Overview of HIERMATCH**. Our proposed training strategy makes use of samples labeled at different hierarchical levels. Our network consists of different hierarchical classifiers (inspired by [7]). At test time, we use only the finest-level classifier for classification. We assume that coarse-level labels are available for the samples labelled at a finer-level. HIERMATCH is a general-framework that can be used with any semi-supervised learning approach to incorporate information about label hierarchies.

hierarchies during the SSL training, it should be possible to impose stronger priors and learn feature representations that will reduce confusion between unrelated Level 3 categories, e.g., between `Sunflower` and `Flatfish`. This observation is based on the argument that a human who is confident of an image being that of a `Flower` (Level 2) is very unlikely to identify it as a `Flatfish` (Level 3).

Existing techniques leverage the hierarchical priors on data in tasks such as object recognition [20, 15, 14], text-classification [29], retrieval tasks [1], or by reducing the severity of mistakes [4, 16]. To our knowledge, ours is the first work that leverages hierarchical labels as knowledge priors in the semi-supervised setting leading to reduced labeling costs.

In this paper, we present an approach to incorporate label hierarchy in the SSL framework that trades off more expensive labels at the finest granularity of categories for cheaper labels at coarser granularity *while maintaining classification performance* at the finest level. We refer to this training strategy as HIERMATCH and test it in conjunction with two recently proposed semi-supervised learning techniques, MixMatch [3] and FixMatch [24]. We demonstrate the effectiveness of HIERMATCH on datasets that offer two or more levels of granularity.

Our contributions are as follows:

- We propose a novel framework, HIERMATCH, for improving the performance of a semi-supervised learning algorithm by exploiting the label hierarchies inherent in datasets.
- We experimentally validate that HIERMATCH achieves significant performance improvement on CIFAR-100 [18] and fine-grained NABirds [26] dataset compared to MixMatch [3] by reducing 4000 and nearly 1000 fine-grained labels on CIFAR-100 and NABirds respectively. HIERMATCH can reduce the requirement to 50% samples at the finest level with only 0.59% performance drop on CIFAR-100.
- We demonstrate the robustness of HIERMATCH to the choice of hierarchy with experiments on two different hierarchies for CIFAR-100: one with two levels, and the other with three. HIERMATCH improves upon baselines for both the settings.
- We show that HIERMATCH does not need any additional hyperparameters. Furthermore, it works well with the hyperparameters as those set for the respective baseline SSL methods, therefore eliminating the need for any additional hyperparameter tuning.

The rest of the paper is organized as follows: Related work is discussed in Section 2. In Section 3, we first discuss

the preliminary ideas, followed by our proposed approach. Section 4 presents our experimental evaluations, followed by a discussion of results and their analysis in Section 5 and we finally conclude in Section 6.

2. Related Work

2.1. Semi-supervised learning methods

Semi-supervised techniques utilize large amounts of unlabeled data by imposing a loss on predictions made on this unlabeled data. In the past, loss terms have been proposed to either nudge the model to make predictions with high confidence on unlabeled data (Entropy Minimization [13]), or to predict a similar output distribution on perturbed inputs (consistency regularization [19, 25, 3, 24]).

Laine et al. [19] incorporate consistency regularization loss in their π-*Model*. For every image, labeled and unlabeled, they pass two augmentations of it through their network and force their network to make similar predictions on both (unsupervised loss term). For labeled images, they additionally impose cross-entropy loss (supervised loss term). As the training progresses and the model grows more confident in its predictions, they ramp up the weight of the unsupervised loss term. To reduce the number of forward passes per iteration, thus simplifying training, they also propose *Temporal Ensembling* which alleviates the need for the second forward pass by maintaining an exponential moving average (EMA) of the model's previous predictions. However, *Temporal Ensembling* introduces a large memory constraint that can pose challenges on large datasets. Subsequently, Tarvainen et al. [25] improved upon the π-Model [19] by introducing a student-teacher model, wherein the weights of the teacher model are an EMA of the weights of the student model. They empirically show that using a teacher model improves the quality of targets produced that are used to enforce consistency regularization.

Zhai et al. [31] bridge the gap between self-supervised learning and semi-supervised learning by proposing S4L. The basic S4L technique is simple, a self-supervised loss (eg. predicting rotations [17]) is computed on every image (labeled and unlabeled), and an additional cross-entropy loss is computed only on the labeled image.

Berthelot et al. [3] combine different SSL techniques like consistency regularization, entropy minimization and general regularisation using mixup [32] in their proposed framework, MixMatch. ReMixMatch [2] by Berthelot et al. introduce improvements over MixMatch [3] by using distribution alignment and augmentation anchoring. They change consistency regularization by comparing weak augmentations of an image with its strong augmentation and show improvements. Sohn et al. [24], propose FixMatch, which combines different SSL components such as pseudo labeling and consistency regularization. Unlike ReMixMatch [2] that uses *sharpening* for consistency regularization, FixMatch [24] uses *pseudo-labels* of a weakly-augmented image against the strong augmentation of the same image. Despite its simplicity, FixMatch achieves state-of-the-art performance on most datasets and gives promising results in extremely label-scarce settings. To the best of our knowledge, none of the current works in the domain of SSL leverage hierarchical information present in most datasets to boost their model performance.

2.2. Hierarchical classification methods

The hierarchical structure present in data has been used in the supervised setting for various learning tasks, some of which we discuss in this section.

Devise [10], a label-embedding method, uses pretrained visual models to obtain the image embeddings and a pretrained word2vec model to obtain the corresponding label embeddings. They optimized a ranking loss such that the cosine similarity of the correct label should be higher than the similarity of any other label. Barz and Denzler [1] map samples onto a unit hypersphere with distances based on the least common ancestor and maximize the correlation based on the cosine similarity. Bilal et al. [6] analyze feature maps at different layers of the convnet block and errors in a simple classifier using the confusion matrix and branch out classification layers from the output of different convnet blocks in the decreasing order of coarseness at the last layer.

Wu et al. [28] use a multi-task loss function for each hierarchical level post the last fully connected layer. In the label inference step they smoothen the prediction values to enforce consistency across different semantic levels. Bertinetto et al. [4] point out flaws with Top-k error metrics that, for misclassified samples, all classes other than the ground-truth class are treated as equally wrong. To reduce the severity of the mistakes, they propose two novel loss functions, hierarchical cross-entropy (HXE) loss, and soft-labels loss. Karthik et al. [16] use a conditional risk minimization framework to correct mistakes by introducing hierarchical information at test time. Unlike conventional classifiers that maximize the probability of sample belonging to a class, they propose the use of a minimum risk class, defined based on the cost matrix.

Wang et al. [27] exploit label hierarchies to tackle the task of domain adaptation in FGVC by integrating curriculum learning with adversarial learning to progressively improve domain adaptation in difficult setting of fine-grained categories.

3. Proposed Approach

3.1. Semi-Supervised Baselines [3, 24]

Our technique leverages semi-supervised learning algorithms that incorporate consistency regularisation in their

loss functions. We employ MixMatch [3] and FixMatch [24] as SSL techniques for our experiments.

The semantic structure of an image is invariant to the augmentation applied. Consequently, consistency regularisation encourages a model to make similar predictions on different augmentations of the same image. In the simplest formulation consistency is measured as the mean squared error between predictions on two augmentations of the same image. [19] introduce consistency loss, as an unsupervised loss, and compute this loss on both labeled as well as unlabeled images. The unsupervised loss is ramped up with time as the model grows more confident in its predictions.

MixMatch extends the π model [19] by proposing entropy minimisation via temperature scaling on unlabeled images. This forces the network to make more confident predictions for unlabeled images. MixMatch additionally employs MixUp [32] as a regularizer to encourage convex prediction behaviour between convex combinations of samples. Akin to the π model, MixMatch ramps up the unsupervised loss (consistency regularisation) as training progresses and the model becomes more confident in its predictions. MixMatch achieves much better results than the π model [19], albeit the number of hyperparameters and training time increase.

FixMatch [24] improves on the π model [19] by using pseudo labeling and changing the loss in consistency regularisation. FixMatch produces both a weak and a strong augmentation of every unlabeled image. The prediction of the network on the weakly augmented image is used as a pseudo label for the image if the predicted class probability is higher than some threshold value. This pseudo label is then used as the ground truth label for the prediction of the network on the strongly-augmented image.

FixMatch enforces consistency regularisation only when the model is confident and does away with the need for time based consistency regularisation as is in π model [19] and MixMatch [3]. FixMatch uses less hyperparameters than MixMatch, makes fewer forward passes through the network than MixMatch [3], and has improved results by using pseudo-labels on unlabeled images, confidence thresholding, and introducing strong augmentations in consistency regularisation,

3.2. Hierarchical Classification [7]

Chang et al. [7], by means of a comprehensive human study, conclude that the label for an image is subjective depending on expertise in the domain and that most people prefer multi-granularity labels. They suggest that someone familiar with birds would assign the fine-grained label `Flamingo` to an image containing a flamingo, however, a layperson would assign the coarse label `Bird` to the very same image. An architecture consisting of K independent label classifiers, one for each hierarchical level, forked from a common feature extraction backbone has been used before [27] towards the end of multi-granularity classification. The authors of [7] establish that jointly optimizing for both coarse-grained and fine-grained classification equally on the aforementioned architecture degrades fine-grained classification performance while improving coarse-grained classification performance. Based on this finding, they propose disentangling of features fed into the independent label classifiers. They divide the feature vector f equally into K unique parts denoted as f_k. Each independent label classifier uses its features as well as features of classifiers at levels finer than itself to make predictions. Additionally, gradients from a label classifier are stopped from flowing to finer features during backpropagation and only flow to the features of this classifier. This results in coarse grained classifiers enjoying the benefits of fine grained features while ensuring fine grained features remain unaffected by coarse grained supervision.

3.3. HierMatch

Let H be the number of hierarchical levels in the dataset and let K^i denote the number of classes at the i-th level of the hierarchy, where $i = 1$ is the coarsest and $i = H$ is the finest-level of the hierarchy. We focus on leveraging prior knowledge available in the hierarchical structure of data. Thus, for an image labeled at the hierarchical-level i as y^i, we assume the image also has label information available for all coarser $i-1$ levels as $\{y^j\}_{j=1}^{i-1}$.

Conventional semi-supervised learning frameworks use a set of unlabeled data and a set of labeled data. Labels for images in the labeled set are available only at the finest level of the hierarchy. In our setting of hierarchical semi-supervised learning, we have a set of labeled data for every level of the hierarchy and a set of unlabeled data.

Let $\mathcal{X}^h = \{(x_b, y_b^h) : b \in \mathcal{B}^h\}$ be the set of labeled data where $\mathcal{B}^h$ is the index set of all images that have labels at level $h \in [1, 2, \ldots, H]$ and y_b^h takes one of K^h values. Each image x_b was annotated at some level $j \geq h$ as y_b^j, which is its label at the j-th level of the hierarchy. The knowledge of labels $\{y_b^1, \ldots, y_b^h, \ldots, y_b^j\}$ for x_b follows from our assumption about the knowledge of the hierarchical structure of labels. Thus x_b will feature in all sets $\mathcal{X}^1, \ldots, \mathcal{X}^j$. Let the set of unlabeled data samples be $\mathcal{U} = \{x_b : b \in \mathcal{B}^U\}$ where analogously $\mathcal{B}^U$ is the index set of *unlabeled* images. These samples do not have label information at any hierarchical level.

We denote the feature extractor by $\mathcal{F}(\cdot)$ and the label classifiers corresponding to different hierarchical levels by $\mathcal{G}^1(\cdot), \mathcal{G}^2(\cdot), \ldots, \mathcal{G}^H(\cdot)$, each of which fork from the output of $\mathcal{F}(\cdot)$. The labeled data for classifier $\mathcal{G}^\ell$ consists of all the samples that have a label at the ℓ-th hierarchical level. Unlabeled data for a classifier $\mathcal{G}^\ell$ comprises of all unlabeled samples $\mathcal{U}$ and samples labeled at levels coarser than ℓ. This

follows from the assumption that a sample labeled at a particular level j does not have labels for any level finer than j. Thus, such a sample will be treated as an unlabeled sample by all the finer classifiers i.e. $\{\mathcal{G}^{j+1}, \ldots, \mathcal{G}^{H}\}$. We collect these samples in a set defined as

$$\mathcal{U}^{\ell} = \left(\mathcal{X}^{1} \setminus \mathcal{X}^{l}\right) \cup \mathcal{U} \tag{1}$$

We build upon the hierarchical classification framework proposed by Chang et al. [7] discussed in section 3.2. Akin to their work, we disentangle the feature vectors used for predictions by the independent label classifiers. For an input image x, let the obtained feature vector $f = \mathcal{F}(x)$, which is divided into H equal parts $\{f^1, \ldots, f^H\}$. Each vector f^h corresponds to the h-th hierarchical level. Fine-grained features assist coarse-level classifiers in classification, however, during backpropagation gradients are stopped from flowing to the finer features to avoid biasing them with coarse-level information. Thus, the final feature vector which the label classifier $\mathcal{G}^h(\cdot)$ uses for prediction is

$$f^h_{inp} = \text{CONCAT}(f^h, \Gamma(f^{h+1}), \ldots, \Gamma(f^H)) \tag{2}$$

where $\Gamma(\cdot)$ represents stopping of gradients during backpropagation. For each of the hierarchical classifiers, any semi-supervised learning algorithm can be applied. We use MixMatch [3] and FixMatch [24] as examples of SSL algorithms to validate our claims. We present the complete steps of HIERMATCH in Algorithm 1.

4. Experiments

4.1. Datasets

We evaluate HIERMATCH on the following datasets:

CIFAR-100 [18] - is a standard benchmark dataset which is used by many recent SSL techniques [2, 3, 24]. It contains images of size 32×32 belonging to one of 100 classes. Primarily, we use a two-level hierarchy, having 20 classes at level 1 and 100 classes at level 2 for our experiments. To demonstrate the robustness of our method to different choices of hierarchy, we also report results on a three-level hierarchy [12], an extension to the two-level hierarchy, with 8 classes at level 1, 20 classes at level 2, and 100 classes at level 3.

North American Birds [26] - is a popular benchmark dataset commonly used to evaluate the performance of fine-grained visual classification methods. The NABirds dataset has a total of 555 classes. We perform experiments with two different hierarchies. The first is a two level hierarchy with 555 classes at level 2 and 404 classes at level 1. The second is an extension of the first with 555 classes at level 3, 404 classes at level 2, and 50 classes at level 1. To our knowledge, this paper is the first to evaluate semi-supervised learning methods on the large-scale fine-grained NABirds dataset.

4.2. Experimental Setup

For both datasets, we split the training data into train and validation sets. We use 5000 samples of CIFAR-100 and 4972 samples of NABirds as validation sets. For CIFAR-100, we stratify our split to maintain an equal number of samples per class in the labeled set. NABirds is a highly imbalanced dataset with number of samples in a class ranging from as low as 4 to a maximum of 60. We split the training set into labeled and unlabeled sets while ensuring every class has at least one sample in the labeled set. When distributing the labeled samples across hierarchies in NABirds, we ensure that there is at least one sample per class at the finest level. We report the final accuracy on the test set for the model with the best validation accuracy.

On the CIFAR-100 dataset, we show results using both MixMatch [3] and FixMatch [24] as SSL algorithms in HIERMATCH framework and denote them as HIERMATCH (M) and HIERMATCH (F) respectively. We use the WideResNet-28-8 architecture [30] having 23.5 million parameters as our backbone network, following the state-of-the-art SSL methods [3, 24].

For a fair comparison, we use the same experimental settings on CIFAR-100 for HIERMATCH (M) and HIERMATCH (F) as is used for baseline MixMatch [3] and FixMatch [24] respectively. We follow [24] and use RandAugment [8] for data augmentation.

For NABirds, we use the WideResNet-50-2 [30] (67.87 million parameters), pretrained on ImageNet [9], as our backbone network. We use only MixMatch [3] as semi-supervised technique in HIERMATCH framework. We present further details of the complete set of hyperparameters used in the supplementary material. All our experiments were done using PyTorch [23], and we used Weights & Biases [5] for experiment tracking, visualizations, and hyperparameter sweeps.

4.3. Baselines

We set out to show that exploiting hierarchies can reduce annotation costs in semi-supervised learning frameworks without compromising performance. We evaluate our approach against the following methods:

Fully-supervised baseline (*Sup.-only*)- The backbone network with a single label classifier trained on all available labeled training samples. This setting represents the highest accuracy a model can achieve *without* utilizing any hierarchical information in a purely supervised setting.

Fully-supervised using limited labeled samples and no unlabeled samples (*Sup.-only*) - The backbone network with a single label classifier trained on a subset of the labeled dataset. This is the maximum performance one can achieve *without* leveraging neither the hierarchical structure nor the unlabeled data.

Fully-supervised baseline using hierarchical network

Algorithm 1: HIERMATCH Algorithm

Input: $\forall h \in \{1, \ldots, H\} \mathcal{X}^h = \{(x_b^h, y_b^h) : b \in \mathcal{B}^h\}$ set of labeled data with labels till h-th level of hierarchies; $\mathcal{U} = \{x_b : b \in \mathcal{B}^U\}$ set of unlabeled data points; Unsupervised loss weight λ_u; number of hierarchical levels in the data H.

1 **for** ℓ=$\{1 \ldots H\}$ **do**
2 | $\mathcal{U}^\ell = \left(\mathcal{X}^1 \setminus \mathcal{X}^l\right) \cup \mathcal{U}$ `// Concatenate to form unlabeled data for level` ℓ
3 **end**
4 **for** e=$\{1 \ldots MaxEpochs\}$ **do**
5 | **for** t=$\{1 \ldots MaxIters\}$ **do**
6 | | $\mathcal{L}_t = 0$
7 | | **for** ℓ=$\{1 \ldots H\}$ **do**
8 | | | $\mathcal{L}_{\mathcal{X}_t^\ell}, \mathcal{L}_{\mathcal{U}_t^\ell}$ = SSLAlgo $(\mathcal{X}_t^\ell, \mathcal{U}_t^\ell)$
| | | | `// Apply SSL algo. at level` ℓ `to calculate sup. & unsup. loss`
9 | | | $\mathcal{L}_t$ += $(\mathcal{L}_{\mathcal{X}_t^\ell} + \lambda_u \mathcal{L}_{\mathcal{U}_t^\ell})$ `// Calculate total loss`
10 | | **end**
11 | | θ = Optimize $(\mathcal{L}_t, \theta)$ `// Update model parameters`
12 | **end**
13 **end**
14 Use $\mathcal{G}^H(f^H)$ for classification at the finest-level.

(*Hierarchical Sup.*) - we train the hierarchical network [7] by utilizing the hierarchical structure inherent in the dataset, on the entire labeled data. This represents the highest achievable accuracy *using the hierarchical coarse labels*.

MixMatch [3], FixMatch [24] - The backbone network with a single label classifier trained using one of the SSL Algorithms MixMatch and FixMatch. In this setting, we use a subset of the entire available data as labeled, with labels at the finest-level only, and all of the unlabeled data.

4.4. Evaluation Protocols

Unlike semi-supervised methods that use a fraction of data as a labeled set, our approach requires labels with different degrees of fineness. Therefore, we validate our approach by performing experiments with varying amounts of labeled data at different levels. We stick to other standard settings from SSL [22, 3, 2, 24]. These include using a small validation set, using the same backbone network and keeping the same experimental settings across all the techniques per dataset. As is typical in the semi-supervised learning setting, we report Top-1 and Top-5 accuracies on three-different folds of labeled data.

5. Results and Analysis

In the following section, an experiment is referred to using a tuple (a, b, c), where a is the number (percentage) of labeled samples at level 3 (finest level), b is the number (percentage) of samples labeled at level 2, and c is the number (percentage) of samples labeled at level 1 (coarsest level). A sample labeled at a particular level ℓ contributes to the corpus of labeled samples at all levels coarser than the level ℓ. Thus, for an experiment (a, b, c), we have a total of a labeled samples at level 3, $a + b$ labeled samples at level 2, and $a + b + c$ labeled samples at level 1. A dash sign '-' in a tuple indicates the absence of that level of the hierarchy. For example, the tuple $(a, b, -)$ denotes a two level hierarchy with a samples labeled at level 3 and b samples labeled only at level 2. However, the tuple $(a, b, 0)$ denotes a three level hierarchy with no additional samples labeled at level 1. In the absence of a hierarchy level ℓ', the losses corresponding to that level ($\mathcal{L}_{\mathcal{X}^{\ell'}}$, $\mathcal{L}_{\mathcal{U}^{\ell'}}$ in Line 8 of Algo. 1) do not contribute at all.

Methods	Labeled Samples	Top-1 (%)	Top-5 (%)
Sup.-only	(45k, -, -)	79.98	93.91
Sup.-only	(10k, -, -)	61.37	83.24
Hierarchical Sup.	(45k, 0, -)	78.13	93.69
MixMatch	(10k, -, -)	71.69 ± 0.33	90.53
HIERMATCH (M)	(10k, 0, -)	**73.42 ± 0.08**	**91.20 ± 0.10**
FixMatch	(10k, -, -)	77.40 ± 0.12	94.3
HIERMATCH (F)	(10k, 0, -)	**77.61 ± 0.06**	**94.70 ± 0.14**

Table 1. Comparisons on the test sets of CIFAR-100 [18] with two-level hierarchy on the baseline methods. Top-1 and Top-5 accuracy are reported on the level 3 (finest) classifier. Approaches that do not use a hierarchical level are indicated with a dash mark '-'.

5.1. Comparison with the baseline-methods

Tables 1 and 5 present the comparisons of our proposed algorithm against the baselines. HIERMATCH outperforms

the standard MixMatch algorithm on both datasets. Using the 2-level hierarchy for CIFAR-100, HIERMATCH increases top-1 accuracy by 1.73% while using the same number of samples labeled at the finest level. Using the 3-level hierarchy on NAbirds, the improvement in the top-1 accuracy is 1.02%.

Can HIERMATCH maintain performance, by leveraging hierarchical labels, despite reduction in number of samples labeled at the finest level?

We use MixMatch and FixMatch with 10k labeled samples at the finest level as our baselines to answer this question. We reduce the number of samples labeled at the finest level and use an equal number of samples labeled at coarser levels instead. We report results on CIFAR-100 for HIERMATCH (M) in Table 2 and for HIERMATCH (F) in Table 4. In the `(8k, 2k, -)` setting, using 2k less labeled samples at the finest level, HIERMATCH (M) improves upon the `(10k, -, -)` (i.e., the baseline MixMatch) in top-1 accuracy by 0.93% and in top-5 accuracy by 0.3%. In the `(5k, 5k, -)` setting, despite using 50% less labeled samples at the finest level, our technique drops by only 0.59%. In the `(7k, 2k, 1k)` setting, we obtain nearly similar performance as that of the baseline MixMatch. In domains where obtaining labels at the finest level is extremely expensive leveraging our technique can significantly reduce annotation costs whilst achieving similar performance.

The results for NABirds are reported in Table 6. We observe trends similar to those observed with CIFAR-100. In the settings `(15%, 5%, -)`, we reduce the number of samples at the finest level by 995 as compared to the baseline setting `(20%, -, -)`, and improve on the top-1 and top-5 accuracy by 0.23% and 0.59% respectively. However, similar performance improvement is not observed in the much harder setting `(15%, 0, 5%)` since 5% samples from level 3 (finest level) are replaced by an equal number of samples at level 1 (coarsest level). This can be attributed to the fact that we only have 50 categories at level 1 which results in orders of magnitude less coverage of the input space. We observe high variations in the HIERMATCH (M) experiments of NABirds as total number of samples in fine-grained categories are in the range from 4 to 60. The encouraging performance of HIERMATCH (M) on NABirds, a challenging and a highly imbalanced dataset, speaks to the robustness of our algorithm.

How effective is HIERMATCH if additional coarse-level labels are provided?

We conduct a study on CIFAR-100 using MixMatch as SSL algorithm to establish the effectiveness of our algorithm with additional coarse-level labels, results of which are present in Table 7. By keeping the number of samples constant at the finest level, we compare the results by varying the number of coarse-level samples. On all four settings, HIERMATCH (M) is able to achieve significant improvements when additional coarse-level labels are leveraged. This result confirms our claim that by incorporating additional hierarchical level information in an SSL framework with limited labels can boost its performance significantly, and obtaining such information is cost-effective.

Labeled Samples	Top-1 (%)	Top-5 (%)
(10k, -, -)	71.69 ± 0.33	90.53
(10k, 0, -)	73.42 ± 0.08	91.20 ± 0.10
(10k, 0, 0)	72.85 ± 0.56	91.20 ± 0.20
(8k, 2k, -)	72.62 ± 0.10	90.83 ± 0.27
(7k, 2k, 1k)	71.68 ± 0.59	90.41 ± 0.11
(6k, 4k, -)	71.93 ± 0.26	90.34 ± 0.09
(6k, 2k, 2k)	72.04 ± 0.37	90.33 ± 0.26
(5k, 5k, -)	71.10 ± 0.18	90.00 ± 0.09
(5k, 3k, 2k)	70.99 ± 0.24	89.73 ± 0.24

Table 2. Performance comparison on CIFAR-100 [18] using MixMatch as SSL Algorithm. Accuracy on the level 3 (finest) classifier using different number of labeled samples from different hierarchies while keeping the total number of labeled samples as 10,000 across the two-levels of hierarchy. Approaches that do not use a hierarchical level are indicated with a dash mark '-'.

Labeled Samples	Top-1 (%)	Top-5 (%)
(400, -, -)	24.31 ± 0.22	37.87 ± 0.62
(400, 0, -)	31.39 ± 1.86	47.73 ± 1.50
(300, 100, -)	28.38 ± 0.65	45.32 ± 1.39

Table 3. Performance comparison on low-data CIFAR-100 [18] using MixMatch as SSL Algorithm. Accuracy on the level 2 (finest) classifier using different number of labeled samples from different hierarchies while keeping the total number of labeled samples as 400 across the two-levels of hierarchy. Approaches that do not use a hierarchical level are indicated with a dash mark '-'.

Labeled Samples	Top-1 (%)	Top-5 (%)
(10k, -, -)	77.40 ± 0.12	94.3
(10k, 0, -)	77.61 ± 0.06	94.70 ± 0.14
(8k, 2k, -)	77.10 ± 0.16	94.53 ± 0.08
(5k, 5k, -)	76.06 ± 0.15	94.12 ± 0.12

Table 4. Performance comparison on CIFAR-100 [18] using FixMatch as SSL Algorithm. Accuracy on the level 3 (finest) classifier using different number of labeled samples from different hierarchies while keeping the total number of labeled samples as 10,000 across the two-levels of hierarchy. Approaches that do not use a hierarchical level are indicated with a dash mark '-'.

How does HIERMATCH perform in a small labeled data setting?

To test the performance of HIERMATCH in low-data setting, we use a total of 400 labeled samples from CIFAR-100

Methods	Labeled Samples	Top-1(%)	Top-5(%)
Sup.-only	(18957, -, -)	69.94	88.37
Sup.-only	(4972 , -, -)	55.71	50.86
Hierarchical Sup.	(18957, 0, 0)	71.39	88.78
MixMatch	(4972, -, -)	61.31 $\pm$ 0.23	81.86 $\pm$ 0.09
HIERMATCH (M)	(4972, 0, 0)	**62.33** $\pm$ **0.24**	**82.20** $\pm$ **0.34**

Table 5. Comparisons on the test sets of NABirds [26] with three-level hierarchy on the baseline methods. Top-1 and Top-5 accuracy are reported on the fine-grained classifier. Approaches that do not use a hierarchical level are indicated with a dash mark '-'.

dataset i.e. 4 samples per finest-class are used. We record mean and standard deviation of label-scarce setting in Table 3. Using baseline MixMatch [3] as SSL algorithm, we achieved an accuracy of 24.31%. Using additional coarse-level labels of these 400 samples with HIERMATCH under the setting `(400, 0, -)` gives a significant boost of 7.08% and 9.86% in top-1 and top-5 accuracy metrics respectively. In the experiment setting `(300, 100, -)`, where we use only 3 samples per-fine grained class and an additional 100 coarse-labeled samples were used, we get an accuracy of 28.38% which is nearly 4% improvement over baseline MixMatch [3]. These improvements confirm that reducing fine-grained samples, and instead using coarser-samples can improve performance in low labeled data scenarios too.

5.2. Effect of number of hierarchical levels

In this subsection, we analyze the effect of varying the number of hierarchical levels for both CIFAR-100 and NABirds. We conduct experiments with hierarchical information from two and three hierarchical levels. The results of these are presented in Table 2. For the experiments `(10k, -, -)`, `(10k, 0, -)` and `(10k, 0, 0)` where 10k samples remain the same at the finest-hierarchical level, the top-5 accuracy improves as we leverage more hierarchical information. We reduce 1000 finest-level samples from `(8k, 2k, -)`, increase them in the coarsest-level in `(7k, 2k, 1k)`, and observe a slight performance drop of 0.34% in the top-1 accuracy.

The experiments `(6k, 4k, -)` and `(6k, 2k, 2k)` marginally differ by 0.11% in top-1 and 0.01% in top-5 accuracies respectively. The next set of experiments `(5k, 5k, -)` and `(5k, 3k, 2k)`, involving dropping of 2k samples from level 2 and increasing them in level 1, also differ marginally by 0.11% and 0.27% in top-1 and top-5 accuracies respectively.

6. Conclusion

We introduced HIERMATCH, a technique that leverages hierarchical structure often available with real-world datasets to empower existing semi-supervised learning methods. Through experiments for two different choices of hierarchies on both CIFAR-100 and NABirds we found that with fewer number of fine-grained labeled samples, HIERMATCH was able to achieve competitive or better performance than the baseline MixMatch technique. This work establishes that using hierarchical priors boosts the performance of state-of-the-art semi-supervised learning methods on large datasets. For future work, we are interested in exploring other ideas from semi-supervised learning that might be more suitable for hierarchical structure of data as well as validating our technique on more complex datasets.

Labeled Samples	Top-1 (%)	Top-5 (%)
(20%, -, -)	61.31 $\pm$ 0.23	81.86 $\pm$ 0.09
(20%, 0, -)	62.83 $\pm$ 0.48	83.36 $\pm$ 0.29
(20%, 0, 0)	62.33 $\pm$ 0.24	82.20 $\pm$ 0.34
(15%, 5%, -)	61.54 $\pm$ 0.22	82.45 $\pm$ 0.25
(15%, 5%, 0)	60.49 $\pm$ 0.81	80.03 $\pm$ 1.68
(15%, 0, 5%)	59.33 $\pm$ 1.04	78.80 $\pm$ 1.82
(13%, 7%, -)	60.03 $\pm$ 0.60	81.01 $\pm$ 0.89
(13%, 4%, 3%)	59.39 $\pm$ 0.82	79.03 $\pm$ 1.50

Table 6. Performance comparison on NABirds [26]. Top-1 and Top-5 Accuracy (%) are reported on the level 3 classifier using different number of labeled samples from different hierarchies while keeping the total number of labeled samples as 20% across the three-levels of hierarchy. % indicate the respective number of samples used 20% - 4972, 15% - 3977, 13% - 3475, 7% - 1497, 4% - 991, and 3% - 506. Approaches that do not use a hierarchical level are indicated with a dash mark '-'.

Labeled Samples	Top-1 (%)	Labeled Samples	Top-1 (%)
(5k, 0, -)	67.92	(5k, 5k, -)	71.21
(6k, 0, -)	69.65	(6k, 4k, -)	71.72
(8k, 0, -)	70.78	(8k, 2k, -)	72.61
(10k, 0, -)	73.51	(10k, 5k, -)	74.91

Table 7. Ablation study on CIFAR-100 [18] to show that the additional coarse-class level labels can indeed help improve the performance. Samples from fine-grained class (level 2) are kept constant while an increase in the additional number of coarse-grained (level 1) samples improve the performance. Accuracy is reported on the fine-grained classifier.

Acknowledgement

This work was supported partly by SERB, Govt. of India, under grant no. CRG/2020/006049 and partly by the Infosys Center for Artificial Intelligence at IIIT-Delhi.

References

[1] Björn Barz and Joachim Denzler. Hierarchy-based image embeddings for semantic image retrieval. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 638–647. IEEE, 2019. 2, 3

[2] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*, 2020. 1, 3, 5, 6

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1, 2, 3, 4, 5, 6, 8

[4] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A. Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 3

[5] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com. 5

[6] Alsallakh Bilal, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. Do convolutional neural networks learn class hierarchy? *IEEE transactions on visualization and computer graphics*, 24(1):152–162, 2017. 3

[7] Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Your "flamingo" is my "bird": Fine-grained, or not. In *Computer Vision and Pattern Recognition*, 2021. 2, 4, 5, 6

[8] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 5

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[10] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. 3

[11] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4438–4446, 2017. 1

[12] Vivien Sainte Fare Garnot and Loic Landrieu. Leveraging class hierarchies with metric-guided prototype learning. *arXiv preprint arXiv:2007.03047*, 2020. 5

[13] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, pages 529–536, 2004. 3

[14] Kristen Grauman, Fei Sha, and Sung Hwang. Learning a tree of metrics with disjoint visual features. *Advances in neural information processing systems*, 24:621–629, 2011. 2

[15] Sung Hwang, Kristen Grauman, and Fei Sha. Semantic kernel forests from multiple taxonomies. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 1, 2

[16] Shyamgopal Karthik, Ameya Prabhu, Puneet K. Dokania, and Vineet Gandhi. No cost likelihood manipulation at test time for making better mistakes in deep networks. In *International Conference on Learning Representations*, 2021. 2, 3

[17] Nikos Komodakis and Spyros Gidaris. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018. 3

[18] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 2, 5, 6, 7, 8

[19] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 1, 3, 4

[20] Marcin Marszalek and Cordelia Schmid. Semantic hierarchies for visual object recognition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2007. 2

[21] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998. 1

[22] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 6

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[24] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 596–608. Curran Associates, Inc., 2020. 1, 2, 3, 4, 5, 6

[25] Antti Tarvainen and Harri Valpola. Weight-averaged consistency targets improve semi-supervised deep learning results. *CoRR*, abs/1703.01780, 2017. 1, 3

[26] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015. 1, 2, 5, 8

[27] Sinan Wang, Xinyang Chen, Yunbo Wang, Mingsheng Long, and Jianmin Wang. Progressive adversarial networks for fine-grained domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9213–9222, 2020. 3, 4

[28] Hui Wu, Michele Merler, Rosario Uceda-Sosa, and John R Smith. Learning to make better mistakes: Semantics-aware visual food recognition. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 172–176, 2016. 3

[29] Jincheng Xu and Qingfeng Du. Learning neural networks for text classification by exploiting label relations. *Multimedia Tools and Applications*, 79:22551–22567, 2020. 2

[30] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. 5

[31] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1476–1485, 2019. 3

[32] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 3, 4

[33] Peiqin Zhuang, Yali Wang, and Yu Qiao. Learning attentive pairwise interaction for fine-grained classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13130–13137, 2020. 1