Proceedings of the 8th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'22)
Prague, Czech Republic - July 28- 30, 2022
Paper No. MVML 106
DOI: 10.11159/mvml22.106

Investigating the Interaction between Data and Algorithms

Daniel Pototzky[1,2], Azhar Sultan[1], Lars Schmidt-Thieme[2]
[1]Robert Bosch GmbH
Hildesheim, Germany
daniel.pototzky@de.bosch.com
[2]University of Hildesheim
Hildesheim, Germany

Abstract – Research in computer vision is centered on algorithmic improvements, for example, by developing better models. Thereby, the data is considered fixed. This is in contrast to many real-world applications of computer vision systems in which algorithms and data co-evolve. To address this shortcoming of previous research, we study the properties of the data and their interaction with deep learning algorithms. Thereby, we investigate the size of the data, the share of mislabels, class imbalance and the presence of unlabeled data which can be leveraged using semi-supervised learning. In experiments on 100 classes from ImageNet, we show that a tiny network architecture outperforms a much more powerful one it if has access to only a little bit more data. Only if vast amounts of data are available so that adding even more images has little effect on performance, large architectures dominate smaller ones. If little data is provided, adding a few labeled images has a huge effect on accuracy. Once accuracy saturates, massive amounts of additional data are needed to achieve even small improvements. Furthermore, we find that mislabels severely reduce performance. To fix that, we propose a cost-efficient way of identifying mislabels which is especially beneficial if many images are already available. Conversely, if little data is available, labeling more images is more advantageous than cleaning existing annotations. In the case of imbalanced data, we illustrate that labeling more instances from rare classes has a much greater effect on performance than only increasing dataset size. Moreover, we show that leveraging unlabeled images by semi-supervised learning offers a consistent benefit even if the labeled subset contains significant label noise.

Keywords**:** Dataset Size, Mislabels, Class Imbalances, Unlabeled Data, Semi-Supervised Learning

1. Introduction

Machine learning research has seen great advances in recent years. In computer vision, the performance on many challenging benchmarks including ImageNet [1] and MS COCO [2] has greatly improved. This was achieved by developing better models and superior training strategies. At the same time, aspects of the dataset have received much less attention. In particular, the interaction between algorithms and the properties of the data has barely been studied. This is in stark contrast to industrial applications, where algorithms and data often co-evolve.

In this work, we study different aspects of datasets as well as their interaction with machine learning algorithms. We cover the size of the dataset, mislabels, imbalanced classes as well as access to additional unlabeled images and their interaction with different models. In experiments on 100 classes from ImageNet, we show that increasing the dataset size does improve performance, but even small models outperform larger ones if they have access to a little bit more data. Furthermore, we propose a simple method for correcting mislabels. We find that if a large, noisy dataset is available, the best strategy is to remove the noise whereas in the case of a small noisy dataset additional images should be labeled. Moreover, we illustrate that class imbalance in the dataset is very detrimental. In such a situation labeling instances from rare classes is very beneficial and much more important than increasing the overall data size. In addition, we show experiments for leveraging unlabeled data using FixMatch [3], a semi-supervised learning algorithm that leads to consistent improvements. This is the case even if the labeled portion of the dataset contains a large number of mislabels. Overall, our contributions can be summarized as follows:

- We study the interaction between data and algorithms and provide guidelines for computer vision researchers on what actions to take in different situations.
- We show that less powerful models often outperform larger ones if they have access to only a little bit more data

- We present a simple method for finding potential mislabels in an existing dataset. This reduces costs for cleaning datasets which in many settings is more important than increasing the dataset size.
- We show that reducing class imbalance by labeling instances from rare classes has a greater effect on performance than only increasing the dataset size.
- We show that leveraging unlabeled data using semi-supervised learning consistently improves performance even if the labeled subset contains many mislabels.

2. Related Work

In this section, we provide an overview of related areas. First, we cover existing works about dealing with mislabeled data. Second, we include research on imbalanced data. Third, we provide a short overview of semi-supervised image classification.

2.1. Dealing With Mislabeled Data

Label noise in image classification is mostly studied on small datasets like Cifar10 and Cifar100 in a synthetic setting [4][5]. For example, labels are randomly flipped by a certain probability or, in what is called asymmetry flipping, only similar classes get mixed up, e.g., bird and plane in Cifar10. In many cases, it remains unclear how well these methods that are optimized for synthetic label noise work in a real-world setting.

2.2. Imbalanced Data

Imbalances in the data are a common issue in many machine learning settings. Most notably, the class distribution of datasets varies. In OpenImages, the ratio between the most frequent and the least frequent class is 100,000 [6]. In MS COCO [2] the ratio is 1,000. Similarly, in object detection, there is usually a background-foreground imbalance [7]. Imbalance problems can be addressed by adding weights in the loss function or by upsampling rare classes.

2.3. Semi-Supervised Image Classification

Methods for semi-supervised image classification can broadly be divided into three groups, consistency-based methods [8] [9], pseudo-label methods [10] [11] and those that integrate ideas from self-supervised learning [12][13].

Consistency-based methods typically add a loss term that enforces consistent predictions for an image and perturbed versions of it. The underlying idea is that even if the true label is unknown, a model should still make the same prediction regardless of how the image is perturbed. Of particular importance is the method Unsupervised Data Augmentation (UDA) [8], which shows the benefits of applying strong data augmentation to unlabeled images for image perturbation.

The majority of methods in semi-supervised image classification generate pseudo-labels on unlabeled images. In the simplest form, a pseudo-label is created by making a prediction on an unlabeled image, converting it to a one-hot representation, and then using it as a target in training [14]. Some recent methods use pseudo-labels to implicitly enforce consistency in the model output [3].

3. Methods

3.1. Supervised Learning

We use a common supervised learning protocol and minimize the cross-entropy loss between predictions and targets. We use the same hyperparameters for optimization as in FixMatch [3]. Therefore, the learning rate is set to 0.003, weight decay to 0.0005 and SGD with Nesterov momentum are used for optimization. Furthermore, we use a batch size of 32.

3.2. FixMatch

FixMatch [3] is a semi-supervised learning algorithm that generates pseudo-labels on unlabeled images. Weakly-augmented images are input into the model for pseudo-label generation and a strongly augmented version of the same image is used in training. By doing so, output consistency is implicitly enforced, as the network is trained to make the same prediction on a strongly augmented image as it did on a weakly-augmented version for pseudo-label generation.

4. Experiments

In this section, we conduct a broad range of experiments to study the interaction between data and deep learning algorithms. We focus our investigations on the first 100 classes from ImageNet. This subset of ImageNet is large (130,000 images) and contains a large number of similar classes (e.g., different breeds of dogs) which makes it a challenging test ground for our experiments.

4.1 The Size of the Dataset

First, we investigate the effect that varying the size of the dataset has on performance. We compare two different network architectures, ResNet-50 [15] and SqueezeNet-1.0 [16]. ResNet-50 is one of the most frequently used architectures which achieves an accuracy of 76.5% on full ImageNet by using approximately 25 million parameters. Conversely, SqueezeNet-1.0 is a tiny architecture known for reasonable classification accuracy (55% on full ImageNet) while using only around 1 million parameters. As can be seen in Figure 1, increasing the dataset size leads to improvements in performance for both architectures. Thereby, ResNet-50 consistently outperforms SqueezeNet-1.0 on the same number of images. However, if SqueezeNet-1.0 has access to only a little bit more data (e.g., 2% of images instead of 1%), it achieves higher accuracy than ResNet-50 (at least if little data is available overall). This example of a tiny architecture outperforming a larger and more powerful one illustrates the importance that the available data has compared to the network architecture. Despite this, the vast majority of research focuses on improving models and algorithms. Only if many images are available (i.e. at least 40% of this subset, equivalent to more than 50,000 images), does ResNet-50 outperform SqueezeNet-1.0 even if SqueezeNet-1.0 is trained on more data.

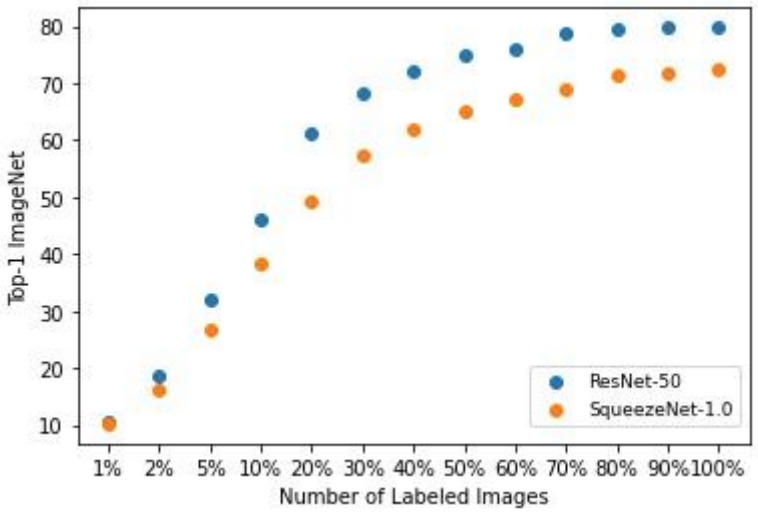

Fig. 1: Top-1 accuracy on the first 100 classes of ImageNet given varying numbers of labeled images. The more data is available, the higher the performance of both ResNet-50 and SqueezeNet-1.0. For the same number of images, ResNet-50 outperforms SqueezeNet-1.0, which only uses approximately 1 million parameters. However, if SqueezeNet-1.0 has access to just a little more data (e.g. 2% of images instead of 1%) it outperforms ResNet-50, which uses almost 25 times as many parameters. Only if vast amounts of data are available so that adding even more images has little effect on performance, does ResNet-50 dominate SqueezeNet-1.0.

Figure 2 shows the labeling costs measured in hours of labeling effort that are necessary to reach a certain accuracy on ImageNet with a ResNet-50. For simplicity, we assume that each image is annotated by a single labeler, no quality checks are conducted and labeling one image takes around 50 seconds, which is the time reported for labeling one bounding box on ImageNet [1]. As can be seen, the first hours of manual labeling lead to a huge increase in top-1 accuracy. However, this rise quickly saturates and further improvements in accuracy require a large amount of additional labeling.

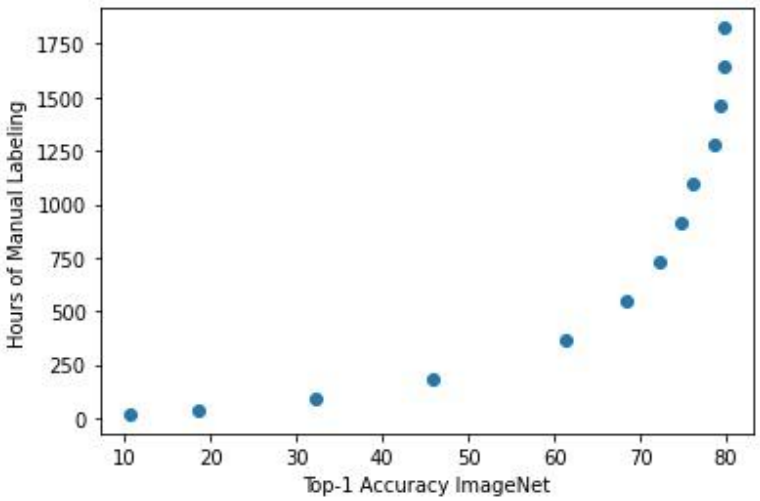

Fig. 2: Labeling costs measured in hours of manual labeling for reaching a given accuracy on ImageNet with a ResNet-50. Initially, adding a few labeled images at little cost has a huge effect on accuracy. Once accuracy saturates, massive amounts of additional data are needed to achieve small improvements.

4.2 Mislabels in the Dataset

Acquiring a cleanly-labeled dataset is very difficult in practice. This is especially the case if visually similar classes are included. For example, the first 100 classes of ImageNet contain a variety of different fish that only domain experts can distinguish. To ensure high label quality, ImageNet was annotated using at least 10 labelers per image. Images that did not reach a high agreement among labelers were discarded. Due to financial limitations, it is not always possible to have multiple labelers per image. As a result, many real-world datasets contain some degree of wrong labels. Figure 3 shows results if a certain percentage of the training data has a wrong label. Mislabels are generated by randomly assigning a label from related classes. For the first 100 classes of ImageNet, this means, for example, that one fish gets the label from another kind of fish assigned (class indices: 1-7 fish, 8-25 birds, 26-80 amphibians, 81-100 birds). The more mislabels, the greater the drop in performance. For example, if 10% of the data is available, 20% of mislabels lead to a drop in performance from 46.0% to 33.8% accuracy.

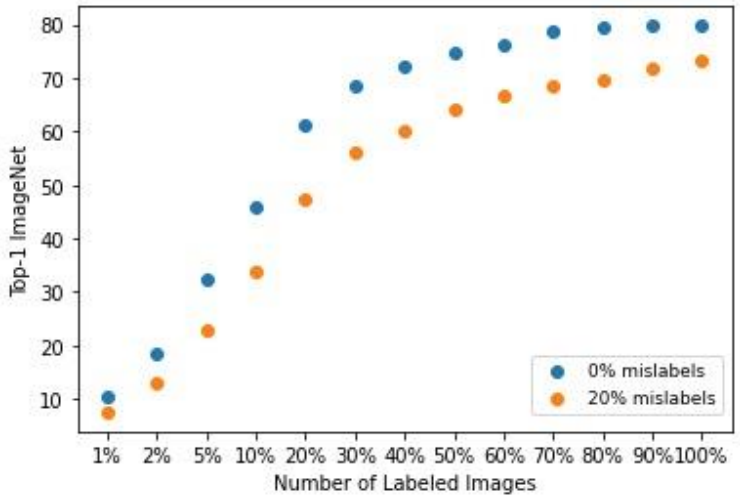

Fig. 3: Performance on ImageNet with a ResNet-50 depending on the number of labeled images and whether the labels are noisy. The presence of mislabels greatly decreases performance.

4.3 Cost-Efficient Mislabel Correction

In this section, we investigate if an existing noisy dataset should be cleaned or if a given labeling budget should be spent on annotating additional images. Thereby, we also suggest a cost-efficient method for reducing mislabels. The naïve way of cleaning a dataset would be to quality check each image individually and to correct mislabels. Instead, we propose to automatically pre-select images that are likely to contain a mislabel. For this, we divide the noisy training set into k different parts. To determine mislabels in one of the k parts (called the hold-out part), we train a model on the remaining k-1 parts and create predictions on the unseen hold-out part. Each image for which the model prediction is substantially different from the label is selected for quality checking. By focusing on images that likely contain mislabels, we reduce the number of images for quality checks. This approach significantly reduces the overall costs for quality checking.

In Figure 4 we conduct the experiment about what to do if 10% or 60% of the ImageNet dataset are available with 20% of mislabels each. We compare adding more images, naïve quality checking of images as well as the improved method for quality control that we suggest. For simplicity, we set k=2, although a larger k would probably lead to even greater cost savings. Unfortunately, no exact numbers are available for how long quality checking and labeling takes for ImageNet per-image labels. However, [17] reports that labeling bounding box labels on ImageNet takes around 50.8 seconds per instance and quality checking requires 21.9 seconds. In the absence of exact numbers for image-level labels, we use the time measures for bounding box labels on ImageNet. This means, that labeling a new image requires 50.8 seconds. Quality checking with label correction needs 72.2 seconds (21.9 seconds + 50.8 seconds) and only quality checking takes 21.9 seconds. As can be seen, if the initial noisy dataset is small (10% of ImageNet; left part of Figure 4) simply adding more data is similarly beneficial as the optimized quality checking strategy. However, if the initial dataset is large (60% of ImageNet; right part of Figure 4) our improved method for quality checking is the best strategy to employ.

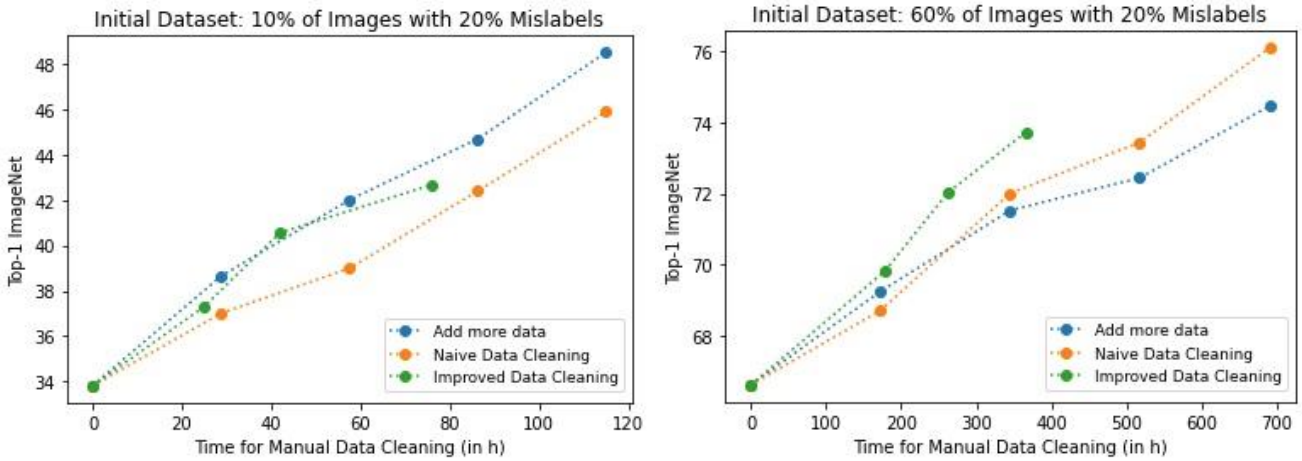

Fig. 4: Used time for manual data cleaning and top-1 accuracy on ImageNet. We assume that initially 10% (left Figure) or 60% (right Figure) of the ImageNet dataset with 20% mislabels are available. If little data is available adding more data is at least as good as cleaning labels. Conversely, if a lot of data is provided, removing mislabels is more promising than adding more images.

4.4 Imbalanced Classes

We study the impact that class imbalances have on performance. Many commonly used datasets are severely imbalanced. For example, the ratio between the most frequent and least frequent class in MS COCO [2] is 1,000 and in OpenImages [6] it is even 100,000. By default, the ImageNet dataset has even-sized classes. This allows us to model the effect that class imbalance has on performance. We model imbalance for each class by drawing a value from a uniform distribution within the interval [a, b]. We fix a at 0.1 and by modifying b we simulate the imbalance. For each class, we

then determine the number of images by dividing the sampled value by the sum of all sampled values from all classes and multiplying this value by the number of images to include.

In Figure 5 we conduct experiments for a varying share of images if there is a class imbalance or not. Severe class imbalance denotes the case where the ratio between the most frequent and least frequent classes is approximately 100. This is much less than in MS COCO and OpenImages, however, it still does result in substantially reduced performance compared to a balanced dataset. If we then double the dataset size (e.g. by labeling more images) and do not pay attention to class imbalances, performance increases somewhat (e.g. orange dot at 10% of images to orange dot at 20% of images). Conversely, only labeling images of rare classes dramatically improves performance (see black arrows).

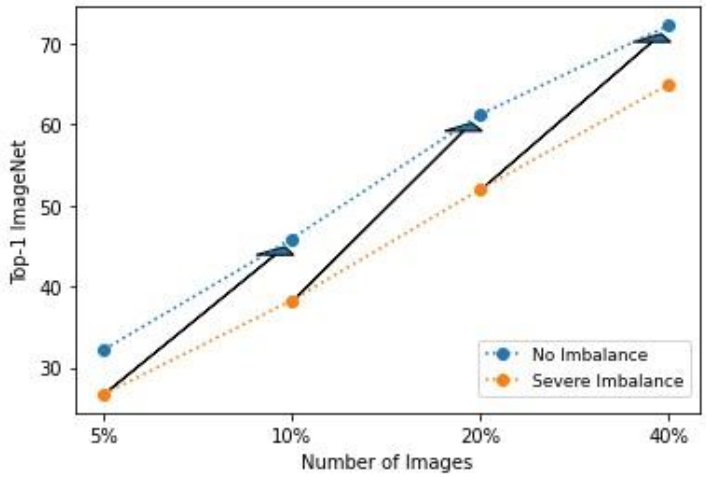

Fig. 5: The effect of class imbalances on performance. Severe class imbalance (defined as a ratio of 100:1 between the most frequent and least frequent class) significantly reduces top-1 accuracy. If more images are labeled with the same class imbalance performance increases only slightly. However, if only rare classes are newly labeled, top-1 accuracy rises substantially, as indicated by the black arrows.

4.5 Semi-Supervised Learning

In many cases, unlabeled data is available in addition to a labeled subset. If we apply supervised learning, we essentially discard these unlabeled images which might contain additional information to learn from. To leverage unlabeled data, we use a semi-supervised learning algorithm, FixMatch. Figure 6 shows results for FixMatch compared to supervised learning. FixMatch greatly improves upon supervised learning by incorporating additional unlabeled images. Notably, by using only 25% of the labels, FixMatch almost matches supervised learning on the full dataset (77.3% vs. 79.8%).

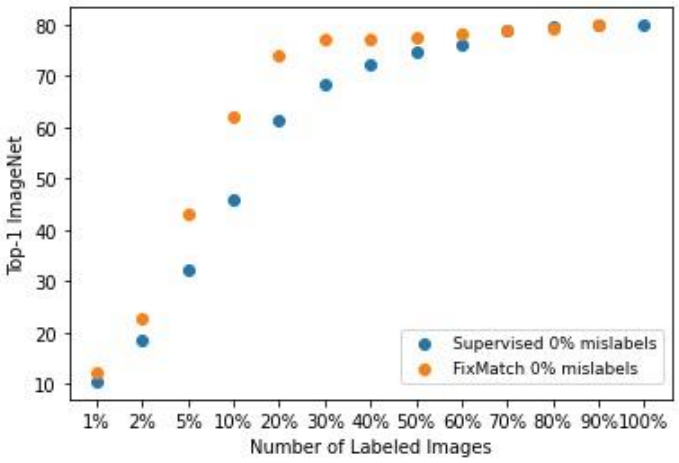

Fig. 6: Leveraging unlabeled images using semi-supervised learning. FixMatch, a semi-supervised learning algorithm consistently improves upon supervised learning. Both setups use a ResNet-50 architecture.

Moreover, Figure 7 investigates FixMatch and supervised learning if the labeled data contains a significant number of wrong labels (i.e. 20%). Even then, FixMatch leads to consistent improvements in performance.

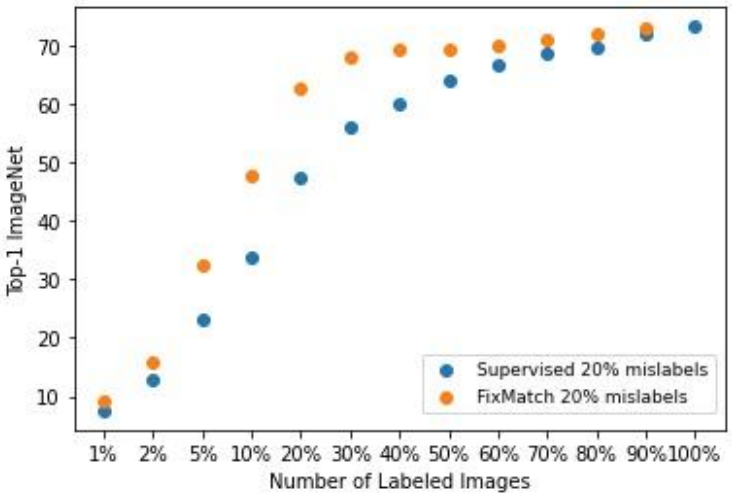

Fig. 7: Semi-Supervised Learning if the labeled dataset contains 20% of mislabels. Even then, semi-supervised learning greatly improves upon supervised learning. Again, both setups use a ResNet-50 architecture.

4. Discussion and Conclusion

This paper investigates several properties of a dataset and their effect on model performance. We find that tiny architectures can outperform much more powerful ones if they have access to just a little bit more data. Only if large amounts of data are available, large architectures offer consistent improvements. If only a few images are available, labeling more instances is a cost-efficient way of improving performance. Conversely, if a lot of data is provided already, further performance improvements require large amounts of additional labels. Mislabels significantly reduce performance. If a small noisy dataset is provided, labeling additional images is a good strategy whereas in the case of a large noisy dataset, the labeling budget should be spent on reducing mislabels. Furthermore, imbalanced classes reduce performance. If additional labels are acquired for rare classes, overall accuracy significantly improves. Finally, we find that semi-

supervised learning offers consistent improvements over supervised learning even if the annotated dataset contains a substantial amount of mislabels.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," IEEE Conference on Computer Vision and Pattern Recognition, pp. 248-255, 2009.

[2] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár and C. Lawrence Zitnick, "Microsoft COCO: Common Objects in Context," arXiv:1405.0312, 2014.

[3] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang and Colin Raffel, "FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence," Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020.

[4] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang and Masashi Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," Advances in Neural Information Processing Systems (NeurIPS), 2018.

[5] Eran Malach, Shalev-Shwartz Shai, "Decoupling when to update from how to update," Advances in Neural Information Processing Systems (NeurIPS), 2017.

[6] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig and Vittorio Ferrari, "The Open Images Dataset V4," International Journal of Computer Vision, 2020.

[7] Kemal Oksuz, Baris Can Cam, Sinan Kalkan and Emre Akbas, "Imbalance Problems in Object Detection: A Review," arXiv:1909.00169, 2019.

[8] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong and Quoc V. Le, "Unsupervised Data Augmentation for Consistency Training," arXiv:1904.12848, 2019.

[9] Samuli Laine and Timo Aila, "Temporal Ensembling for Semi-Supervised Learning," International Conference on Learning Representations (ICLR), 2017.

[10] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver and Colin Raffel, "MixMatch: A Holistic Approach to Semi-Supervised Learning," arXiv:1905.02249, 2019.

[11] Hieu Pham, Zihang Dai, Qizhe Xie, Minh-Thang Luong and Quoc V. Le, „Meta Pseudo Labels," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[12] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov and Lucas Beyer, "S4L: Self-Supervised Semi-Supervised Learning," Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[13] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas and Michael Rabbat, "Semi-Supervised Learning of Visual Features by Non-Parametrically Predicting View Assignments With Support Samples," Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021.

[14] Lee, Dong-Hyun, "Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks," ICML 2013 Workshop: Challenges in Representation Learning (WREPL). 2013.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[16] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally and Kurt Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," arXiv:1602.07360, 2016.

[17] Hao Su, Jia Deng and Li Fei-Fei, "Crowdsourcing annotations for visual object detection," Human Computation - Papers from the 2012 AAAI Workshop, Technical Report, 2012.