Beyond Synthetic Noise: Deep Learning on Controlled Noisy Labels

Lu Jiang[1] **Di Huang**[2] **Mason Liu**[3] **Weilong Yang**[1]

Abstract

Performing controlled experiments on noisy data is essential in understanding deep learning across noise levels. Due to the lack of suitable datasets, previous research has only examined deep learning on controlled synthetic label noise, and real-world label noise has never been studied in a controlled setting. This paper makes three contributions. First, we establish the first benchmark of controlled real-world label noise from the web. This new benchmark enables us to study the web label noise in a controlled setting for the first time. The second contribution is a simple but effective method to overcome both synthetic and real noisy labels. We show that our method achieves the best result on our dataset as well as on two public benchmarks (CIFAR and WebVision). Third, we conduct the largest study by far into understanding deep neural networks trained on noisy labels across different noise levels, noise types, network architectures, and training settings. The data and code are released at the following link http://www.lujiang.info/cnlw.html.

1. Introduction

Performing experiments on controlled noise is essential in understanding Deep Neural Networks (DNNs) trained on noisy labeled data. Previous work performs controlled experiments by injecting a series of synthetic label noises into a well-annotated dataset such that the dataset's noise level can vary, in a controlled manner, to reflect different magnitudes of label corruption in real applications. Through studying controlled synthetic label noise, researchers have discovered theories and methodologies that have greatly fostered the development of this field.

However, due to the lack of suitable datasets, previous work has only examined DNNs on controlled synthetic label noise, and real-world label noise has never been studied in a controlled setting. This leads to two major issues. First, as synthetic noise is generated from an artificial distribution, a tiny change in the distribution may lead to inconsistent or even contradictory findings. For example, contrary to the common understanding that DNNs trained on synthetic noisy labels generalize poorly (Zhang et al., 2017), Rolnick et al. (2017) showed that DNNs can be robust to massive label noise when the noise distribution is made slightly different. Due to the lack of datasets, these findings, unfortunately, have not yet been verified beyond synthetic noise in a controlled setting. Second, the vast majority of previous studies prefer to verify robust learning methods on a spectrum of noise levels because the goal of these methods is to overcome a wide range of noise levels. However, current evaluations are limited because they are conducted only on synthetic label noise. Although there do exist datasets of real label noise *e.g.* WebVision (Li et al., 2017a), Clothing-1M (Xiao et al., 2015), *etc*, they are not suitable for controlled evaluation in which a method must be systematically verified on multiple different noise levels, because the training images in these datasets are not manually labeled and hence their data noise level is fixed and unknown.

In this paper, we study a realistic type of label noise in a controlled setting called web labels. "Webly-labeled" images are commonly used in the literature (Bootkrajang & Kabán, 2012; Li et al., 2017a; Krause et al., 2016; Chen & Gupta, 2015), in which both images and labels are crawled from the web and the noisy labels are automatically determined by matching the images' surrounding text to a class name during web crawling or equivalently by querying the search index afterward. Unlike synthetic labels, web labels follow a realistic label noise distribution but have not been studied in a controlled setting.

We make three contributions in this paper. First, we establish the first benchmark of controlled web label noise, where each training example is carefully annotated to indicate whether the label is correct or not. Specifically, we automatically collect images by querying Google Image Search using a set of class names, have each image annotated by 3-5 workers, and create training sets of ten controlled noise levels. As the primary goal of our annotation is to identify

---

[1]Google Research, Mountain View, United States [2]Google Cloud AI, Sunnyvale, United States [3]Cornell University, Ithaca, United States. Correspondence to: Lu Jiang <lujiang@google.com>.

images with incorrect labels, to obtain a sufficient number of these images we have to collect a total of about 800,000 annotations over 212,588 images. The new benchmark enables us to go beyond synthetic label noise and study web label noise in a controlled setting. For convenience, we will refer it as web label noise (or **_red noise_**) to distinguish it from synthetic label noise (or **_blue noise_**)[1].

Second, this paper introduces a simple yet highly effective method to overcome both synthetic and real-world noisy labels. It is based on a new idea of minimizing the empirical vicinal risk using curriculum learning. We show that it consistently outperforms baseline methods on our datasets and achieves state-of-the-art performance on two public benchmarks of synthetic and real-world noisy labels. Notably, on the challenging benchmark WebVision 1.0 (Li et al., 2017a) that consists of 2.2 million images of real-world noisy labels, it yields a significant improvement of 3% in the top-1 accuracy, achieving the best-published result under the standard training setting.

Finally, we conduct the _largest study_ by far into understanding DNNs trained on noisy labels across a variety of noise types (blue and red), noise levels, training settings, and network architectures. Our study confirms the existing findings of Zhang et al. (2017) and Arpit et al. (2017) on synthetic labels, and brings forward new findings that may challenge our preconceptions about DNNs trained on noisy labels. See the findings in Section 5.2. It is worth noting that these findings along with benchmark results are a result of conducting thousands of experiments using tremendous computation power (hundreds of thousands of V100 GPU hours). We hope our (i) benchmark, (ii) new method, and (iii) findings will facilitate future deep learning research on noisy labeled data. We will release our data and code.

2. Related Work

2.1. Datasets of noisy training labels

While many types of noises exist _e.g._ image corruption noise (Hendrycks & Dietterich, 2019), image registration noise (Mnih & Hinton, 2012), or noise from adversarial attacks (Zhang et al., 2019), this paper focuses on label noise, and in particular web label noise – a common type of label noise used in the literature. To the best of our knowledge, there have been no datasets of controlled web label noise. The closest to ours is the datasets of two types of noises: controlled synthetic label noise and uncontrolled web label noise.

In the dataset of controlled synthetic label noise, a series of synthetic label noises are injected into a well-annotated

dataset in a controlled manner to reflect different magnitudes of label corruption in real applications. The most common one is the symmetric label noise, in which the label of each example is independently and uniformly changed to a random class with a controlled probability. Many works studied the symmetric label in controlled settings and presented their findings, including famous ones, like (Zhang et al., 2017) and (Arpit et al., 2017). The symmetric label is also commonly used as a benchmark to evaluate robust learning methods in a noise-control setting _e.g._ in (Vahdat, 2017; Shu et al., 2019; Jiang et al., 2018; Ma et al., 2018; Han et al., 2018b; Van Rooyen et al., 2015; Li et al., 2019; Arazo et al., 2019; Charoenphakdee et al., 2019). Other types of synthetic label noise were also proposed, including class-conditional noises (Patrini et al., 2017; Rolnick et al., 2017; Reeve & Kabán, 2019; Han et al., 2018a), noises from other datasets (Wang et al., 2018; Seo et al., 2019), _etc_. However, these noises are still synthetically generated from artificial distributions. Moreover, different works might use different parameters to generate such synthetic noises, which may make their results incomparable. See the example of (Rolnick et al., 2017) in the introduction.

In the dataset of uncontrolled web label noise, both images and labels are crawled from the web and the noisy labels are automatically determined by matching the images' surrounding text to a class name. This can be achieved by querying a search index (Krause et al., 2016; Li et al., 2017a; Mahajan et al., 2018). For example, In WebVision, Li et al. (2017a) collected web images with noisy labels by querying Google and Flickr image search using the 1,000 class names from ImageNet. Mahajan et al. (2018) gathered a large scale set of images with noisy labels by searching hashtags on Instagram. However, these datasets do not provide ground-truth labels for training examples. Their noise level is hence fixed and unknown. As a result, they are not suitable for controlled studies in which different noise levels must be systematically examined. Besides, to get controlled web noise, it may not be a feasible option to annotate images in these datasets due to their imbalanced class distribution.

2.2. Robust deep learning methods

Robust learning is experiencing a renaissance in the deep learning era. Nowadays training datasets usually contain noisy examples. The ability of DNNs to memorize all noisy training labels often leads to poor generalization on the clean test data. Recent contributions based on deep learning handle noisy data in multiple directions including dropout (Arpit et al., 2017) and other regularization techniques (Azadi et al., 2016; Noh et al., 2017), label cleaning/correction (Reed et al., 2014; Goldberger & Ben-Reuven, 2017; Li et al., 2017b; Veit et al., 2017; Song et al., 2019), example weighting (Jiang et al., 2018; Ren et al., 2018; Shu et al., 2019; Jiang et al., 2015; Liang et al., 2016),

---

[1]From the red and blue pill in the movie "The Matrix (1999)", where the red pill is used to refer to the truth about reality.

cross-validation (Northcutt et al., 2019), semi-supervised learning (Hendrycks et al., 2018; Vahdat, 2017; Li et al., 2020; Zhang et al., 2020), data augmentation (Zhang et al., 2018; Cheng et al., 2019; 2020; Liang et al., 2020), among others. Different from prior work, we introduce a simple yet effective method to overcome both synthetic and real-world noisy labels. Compared with semi-supervised learning methods, our method learns DNNs without using any clean label.

3. Dataset

Our goal is to create a benchmark of controlled noise that resembles a realistic label noise distribution. Unlike existing datasets such as WebVision or Clothing1M, our benchmark provides *controlled label noise* where every single training example is carefully annotated by several human annotators.

Our benchmark is built on top of two public datasets: Mini-ImageNet (Vinyals et al., 2016) for coarse-grained image classification and Stanford Cars (Krause et al., 2013) for fine-grained image classification. Mini-ImageNet has images of size 84x84 with 100 classes from ImageNet (Deng et al., 2009). We use all 60K images for training and the 5K images in the ILSVRC12 validation set for testing. Stanford Cars contain 16,185 high-resolution images of 196 classes of cars (Make, Model, Year) split 50-50 into training and validation set. The standard train/validation split is used.

3.1. Dataset Construction

We build our datasets to replace synthetic noisy labels with web noisy labels in a controlled manner. To recap, let us revisit the construction of existing datasets of noisy labels. Synthetic noisy datasets are generated beginning with a well-labeled dataset. The most common type of synthetic label noise is called *symmetric label* noise in which the label of each training example is independently changed to a random incorrect class with a probability p called noise level.[2]. The noise level indicates the percentage of training examples with incorrect labels. As the true labels for all images are known, one can enumerate p to obtain training sets of different noise levels and use them in noise-control studies. On class-balanced datasets, this process is equivalent to sampling $p\%$ training images from each class and then replacing their labels with the labels uniformly drawn from other classes. The drawback is that the synthetic labels are artificially created and do not follow the distribution of real-world label noise.

On the other hand, there exist a few datasets of uncontrolled web label noise (Xiao et al., 2015; Li et al., 2017a; Krause

et al., 2016). In these datasets, the images are crawled from the web and their labels are automatically assigned by matching the images' surrounding text to a class name. This can be achieved by querying a search index. These datasets contain noisy web labels. However, as their training images are not manually labeled, their label noise level is fixed and unknown, rendering existing datasets unsuitable for controlled studies.

For our benchmark, we follow the construction of synthetic datasets with one important difference – instead of changing the labels of the sampled clean images, we replace the clean images with incorrectly labeled web images while leaving the label unchanged. The advantage of this approach is that we closely match the construction of synthetic datasets while still being able to introduce controlled web label noise.

3.2. Noisy Web Label Acquisition

We collect images with incorrect web labels in three steps: (1) images collection, (2) deduplication, and (3) manual annotation. In the first step, we combine images independently retrieved by Google image search from two sources: text-to-image and image-to-image search. For the text-to-image search, we formulate a text query for each class using its class name and broader category to retrieve the top images. For image-to-image search, we query the search engine using every training image in Mini-ImageNet and Stanford Cars. Finally, we union the two search results, where the text-to-image search results account for 82% of our final benchmark. Note that the rationale for including a small amount of image-to-image results is to enrich the types of web label noises in our benchmark. We show that an alternative way to construct the dataset by removing all image-to-image results leads to consistent results in the Appendix C.

In the second step of deduplication, following (Kornblith et al., 2019), we run a CNN-based duplicate detector over all images to remove near-duplicates to any of the images in the validation set. All images are retrieved under the usage rights "free to use or share" [3]. But we still recommend checking their actual usage right for the image.

Finally, the images are annotated using the Google Cloud labeling service. The annotators are asked to provide a binary question: "is the label correct for this image?". Every image is independently annotated by 3-5 workers and the final label is reached by majority voting. Statistics show annotators disagree only on a small proportion (11%) of the total images. The remainder have unanimous labels agreed by at least 3 annotators.

---

[2]This is slightly different from (Zhang et al., 2017) and is the same as (Jiang et al., 2018). We do not allow examples to be label flipped to their true labels. It makes p denote the exact noise level and independent of the total number of classes.

---

[3]https://support.google.com/websearch/answer/29508

Table 1. Overview of our datasets of controlled red (web) label noise. Blue (synthetic) label noise is also included for comparison.

Dataset	#Class	Noise Source	Train Size	Val Size	Controlled Noise Levels (%)
Red Mini-ImageNet	100	image search label	50,000	5,000	0, 5, 10, 15, 20, 30, 40, 50, 60, 80
Blue Mini-ImageNet		symmetric label flipping	60,000		0, 5, 10, 15, 20, 30, 40, 50, 60, 80
Red Stanford Cars	196	image search label	8,144	8,041	0, 5, 10, 15, 20, 30, 40, 50, 60, 80
Blue Stanford Cars		symmetric label flipping	8,144		0, 5, 10, 15, 20, 30, 40, 50, 60, 80

3.3. Dataset Overview

In total, we collect about 800,000 annotations over 212,588 images, out of which there are 54,400 images with incorrect labels on Mini-ImageNet and 12,629 images on Stanford Cars. The remainder of the images have correct labels. Using web images with incorrect labels, we replace $p\%$ of the original training images in the two datasets, and enumerate p in 10 different levels to create the controlled web label noise: $\{0\%, 5\%, 10\%, 15\%, 20\%, 30\%, 40\%, 50\%, 60\%, 80\%\}$. Similar to synthetic noise, p is made uniform across classes, *e.g.* $p = 20\%$ means that every class has roughly 20% incorrect labels. In Appendix C, we show constructing the dataset only using web images leading to consistent results.

Table 1 summarizes our benchmark. For comparison, we also include synthetic (symmetric) labels of the same 10 noise levels. We use ***blue noise*** to denote the synthetic label noise and ***red noise*** for the web label noise. The sizes of the red and blue training sets are made similar to better compare their difference[4]. Only a subset of web images with incorrect labels is used in our dataset. But we release all 212,588 images along with their annotations, which can be downloaded at `http://www.lujiang.info/cnlw.html` under the license of Creative Commons.

For lack of space, we use Fig. 1 in the Appendix to illustrate noisy images in each dataset. In summary, there are three clear distinctions between images with the synthetic and web label noise. Images with label noise from the web (or red noise) (1) a higher degree of similarity to the true positive images, (2) exist at the instance-level, and (3) come from an open vocabulary outside the class vocabulary of Mini-ImageNet or Stanford Cars.

4. Method

In this section, we introduce a simple method called *MentorMix* to overcome both synthetic and web label noise. As its name suggests, our idea is inspired by MentorNet (Jiang et al., 2018) and Mixup (Zhang et al., 2018). The main idea is to design a new robust loss to overcome noisy labels using curriculum learning and vicinal risk minimization.

---

[4]As existing findings on synthetic noise hold on both the full (60K) and subset (50K) of Blue Mini-ImageNet, we choose to report the results on the 60K full set. This results in a slightly larger Blue Mini-ImageNet but may not affect our main contributions

4.1. Background on MentorNet and Mixup

Consider a classification problem with training set $\mathcal{D} = \{\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i$ denotes the ith training image and $y_i \in [1, m]$ is an integer-valued noisy label over m possible classes and $\mathbf{y}_i$ is the corresponding one-hot label. Note that no clean labels are allowed to be used in training. Let $g_s(\mathbf{x}_i; \boldsymbol{w})$ denote the prediction of a DNN, parameterized by $\boldsymbol{w} \in \mathbb{R}^d$. MentorNet (Jiang et al., 2018) minimizes the following objective:

$$\boldsymbol{w}^* = \underset{\boldsymbol{w} \in \mathbb{R}^d, \mathbf{v} \in [0,1]^n}{\operatorname{argmin}} \mathbb{F}(\mathbf{v}, \boldsymbol{w})$$
$$= \frac{1}{n} \sum_{i=1}^{n} v_i \ell(g_s(\mathbf{x}_i; \boldsymbol{w}), y_i) + \theta \|\boldsymbol{w}\|_2^2 + G(\mathbf{v}; \gamma) \quad (1)$$

where $\ell(g_s(\mathbf{x}_i; \boldsymbol{w}), y_i)$, or $\ell(\mathbf{x}_i, y_i)$ for short, is the cross-entropy loss with the softmax function. θ is the weight decay parameter on the l_2 norm of the model parameters. For convenience, we make the weight decay regularization, data augmentation and dropout all subsumed inside g_s.

Eq. (1) introduces the latent weight variable $\mathbf{v} \in [0, 1]^n$ for every training example. The regularization term G determines a curriculum (Jiang et al., 2015; 2014; Fan et al., 2017) or equivalently a weighting scheme to compute the latent weight v_i to each example. In (Jiang et al., 2018), the weighting scheme is computed by a neural network called MentorNet. In training, $\boldsymbol{w}$ and $\mathbf{v}$ are alternatively minimized inside a mini-batch, one at a time while the other is held fixed. Only $\boldsymbol{w}$ is used at test time.

Mixup (Zhang et al., 2018) minimizes the empirical vicinal risk calculated from:

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{n} \sum_{j=1}^{n} \mathbb{E}_{\lambda}[\ell(g_s(\tilde{\mathbf{x}}_{ij}; \boldsymbol{w}), \tilde{\mathbf{y}}_{ij})] \quad (2)$$

$\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ are computed by the mixup function:

$$\tilde{\mathbf{x}}_{ij} = \lambda \mathbf{x}_i + (1 - \lambda)\mathbf{x}_j \quad (3)$$
$$\tilde{\mathbf{y}}_{ij} = \lambda \mathbf{y}_i + (1 - \lambda)\mathbf{y}_j \quad (4)$$

where λ is drawn from the Beta distribution $\text{Beta}(\alpha, \alpha)$ controlled by hyperparameter α. In practice, only the examples from the same mini-batch are mixed up during training.

4.2. MentorMix

In the proposed MentorMix, we minimize the empirical vicinal risk using curriculum learning. For simplicity, the self-paced regularizer $G(\mathbf{v}) = -\gamma\|\mathbf{v}\|_1$ (Kumar et al., 2010; Jiang et al., 2015) is used and we have:

$$\mathbb{F}(\tilde{\mathbf{v}}, \boldsymbol{w}) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbb{E}_{\lambda} [\tilde{v}_{ij}\ell(\tilde{\mathbf{x}}_{ij}, \tilde{\mathbf{y}}_{ij}) - \gamma\tilde{v}_{ij}], \quad (5)$$

where γ is a hyperparameter. It is easy to derive the optimal weighting scheme when the network parameter $\boldsymbol{w}$ is fixed:

$$\tilde{v}_{ij}^* = \underset{\tilde{\mathbf{v}} \in [0,1]^{n \times n}}{\mathrm{argmin}} \mathbb{F}_{\boldsymbol{w}}(\tilde{\mathbf{v}}) = \mathbf{1}(\ell(\tilde{\mathbf{x}}_{ij}, \tilde{\mathbf{y}}_{ij}) \leq \gamma), \quad (6)$$

where $\mathbf{1}$ is the indicator function and $\mathbb{F}_{\boldsymbol{w}}$ denotes the objective when $\boldsymbol{w}$ is fixed.

Although Eq. (6) gives a closed-form solution for computing the optimal weight, it is intractable to compute as this requires enumerating all pairs of training examples. We therefore resort to importance sampling to find the "important" examples. To do so, we define a stratum for each $\mathbf{x}_i$ and draw an example from the following distribution:

$$P_{\mathbf{v}}(v_i = 1 | \mathbf{x}_i, \mathbf{y}_i) = \frac{\exp(v_i^*/t)}{\sum_{j=1}^{n} \exp(v_j^*/t)}, \quad (7)$$

where t is the temperature in the softmax function and is fixed to 1 in our experiments. $P_{\mathbf{v}}$ specifies a density function over individual training examples. In theory, the distribution is defined over all training examples but, in practice to enable mini-batch training, we compute the distribution within each mini-batch (See Algorithm 1). v_i^* is the optimal weight for $\mathbf{x}_i$. v_i^* is calculated from $v_i^* = \mathrm{argmin}_{v_i} \mathbb{F}_{\boldsymbol{w}}(\mathbf{v})$ in Eq. (1) and can be conveniently obtained by MentorNet. As the optimal v_{ij}^* can only have binary values according to Eq. (6), under the importance sampling we rewrite part of the objective in Eq. (5) as:

$$\sum_{(\mathbf{x}_i,\mathbf{y}_i)\sim\mathcal{D}} \sum_{(\mathbf{x}_j,\mathbf{y}_j)\sim\mathcal{D}} \mathbb{E}_{\lambda}[v_{ij}\ell(\tilde{\mathbf{x}}_{ij}, \tilde{\mathbf{y}}_{ij}) - \gamma v_{ij}]$$
$$= \sum_{(\mathbf{x}_i,\mathbf{y}_i)\sim\mathcal{D}} \sum_{(\mathbf{x}_j,\mathbf{y}_j)\sim P_{\mathbf{v}}} \mathbb{E}_{\lambda}[\ell(\tilde{\mathbf{x}}_{ij}, \tilde{\mathbf{y}}_{ij})] - \gamma \quad (8)$$

where the constant γ will be dropped during training. According to Eq. (6), our goal is to find the mixed-up examples of smaller loss. For a given example $(\mathbf{x}_i, \mathbf{y}_i)$, the loss of the mixed-up example $\ell(\tilde{\mathbf{x}}_{ij}, \tilde{\mathbf{y}}_{ij})$ tends to be smaller when $\ell(\mathbf{x}_j, \mathbf{y}_j)$ is small. Inspired by this idea, we sample $\mathbf{x}_j$ from $P_{\mathbf{v}}$ with respect to the weight v_j^* that is monotonically decreasing with its loss $\ell(\mathbf{x}_j, y_j)$. In this way examples of lower loss are more likely to be selected in the mixup.

Algorithm 1 shows the four key steps to compute the loss for a mini-batch: weight (Step 2-4), sample (Step 5 and 8),

---

Algorithm 1 The proposed MentorMix method.

Input : mini-batch $\mathcal{D}_m$; two hyperparameters γ_p and α
Output : the loss of the mini-batch
1   For every $(\mathbf{x}_i, y_i)$ in $\mathcal{D}_m$ compute $\ell(\mathbf{x}_i, y_i)$
2   Set $\ell_p(\mathcal{D}_m)$ to be the γ_p-th percentile of the loss $\{\ell(\mathbf{x}_i, y_i)\}$.
3   $\gamma \leftarrow \mathrm{EMA}(\ell_p(\mathcal{D}_m))$ // update the moving average
4   $v_i^* \leftarrow \mathrm{MentorNet}(\ell(\mathbf{x}_i, y_i), \gamma)$ // MentorNet weight
5   Compute $P_{\mathbf{v}} = \mathrm{softmax}(\mathbf{v}^*)$, where $\mathbf{v}^* = [v_1^*, \cdots, v_{|\mathcal{D}_m|}^*]$
6   Stop gradient
7   **foreach** $(\mathbf{x}_i, \mathbf{y}_i)$ **do**
8   Draw a sample $(\mathbf{x}_j, \mathbf{y}_j)$ with replacement from $P_{\mathbf{v}}$
9      $\lambda \leftarrow Beta(\alpha, \alpha)$
10     $\lambda \leftarrow v_i^* \max(\lambda, 1-\lambda) + (1-v_i^*)\min(\lambda, 1-\lambda)$
11     $\tilde{\mathbf{x}}_{ij} \leftarrow \lambda\mathbf{x}_i + (1-\lambda)\mathbf{x}_j$
12     $\tilde{\mathbf{y}}_{ij} \leftarrow \lambda\mathbf{y}_i + (1-\lambda)\mathbf{y}_j$
13   Compute $\ell_i = \ell(\tilde{\mathbf{x}}_{ij}, \tilde{\mathbf{y}}_{ij})$
14   Weight ℓ_i using a separate MentorNet   // optional
15   **end**
16   **return** $(1/|\mathcal{D}_m|)\sum_{i=1}^{|\mathcal{D}_m|} \ell_i$

---

mixup (Step 9-12), and weight again (Step 14), where the weighting is achieved by the MentorNet. Following our previous work (Jiang et al., 2018), we avoid directly setting γ and adopt a moving average (in Step 3) to track the exponential moving average of the γ_p-th percentile of the mini-batch loss, in which γ_p becomes the new hyperparameter. Step 4 computes the weight for individual example using a fixed MentorNet. The simplified MentorNet is used in our paper which is equivalent to computing the weight by a thresholding function $v_i^* = \mathbf{1}(\ell(\mathbf{x}_i, y_i) \leq \gamma)$. In Step 10, we assign a bigger (binary) weight between λ and $1 - \lambda$ to $\mathbf{x}_i$ unless its v_i^* is small. This trick is to stabilize importance sampling by encouraging each stratum to receive a bigger weight in the mixup. It leads to marginal performance gains but makes the training more robust to the choice of hyperparameters. In Step 14, the second weighting can be applied optionally using the weights produced by a separate MentorNet. This step is optional for low noise levels but is useful for high noise levels. Our algorithm has the same time and space complexity as that of the Mixup algorithm.

5. Experiments

This section verifies the proposed method on four datasets and presents new findings on web label noise. Specifically, in Section 5.1 we first verify the proposed method on our dataset and then compare it with the state-of-the-art on two public benchmarks of synthetic and real-world noisy labels. In Section 5.2, we empirically examine DNNs trained on controlled noisy labels under various settings and present our findings that challenge our previous understandings.

5.1. Method Comparison

Evaluation metrics: Following prior works, the *peak accuracy* is used as the primary evaluation metric that denotes the

Table 2. Peak accuracy (%) of the best trial of each method averaged across 10 noise levels. – denotes the method failed to train.

| Method | Mini-ImageNet | | | | Stanford Cars | | | |
| | Fine-tuned | | Trained from scratch | | Fine-tuned | | Trained from scratch | |
	Blue	Red	Blue	Red	Blue	Red	Blue	Red
Vanilla	82.3±1.9	81.6±1.9	58.3±10.3	64.9±5.2	70.0±16.8	82.4±6.9	53.8±24.4	77.7±10.4
WeightDecay	81.9±1.8	81.5±1.8	—	—	72.2±17.5	84.3±6.6	—	—
Dropout	82.8±1.3	81.8±1.8	59.3±9.5	65.7±5.0	71.7±16.9	83.8±6.6	62.8±23.5	84.1±6.7
S-Model	82.3±1.8	82.0±1.9	58.7±10.2	64.6±5.1	69.7±16.8	82.4±7.1	53.9±23.5	77.6±10.2
Bootstrap	83.1±1.6	82.7±1.8	60.1±9.7	65.5±4.9	71.7±16.9	82.8±6.7	55.6±23.9	78.9±9.6
Mixup	81.7±1.8	82.4±1.7	60.7±9.8	66.0±4.9	73.1±16.6	85.0±6.2	64.2±21.6	82.5±8.0
MentorNet	82.9±1.7	82.4±1.7	61.8±10.3	65.1±5.0	75.9±16.8	82.6±6.6	56.8±23.1	78.9±8.9
Our MentorMix	**84.2±0.7**	**83.3±1.9**	**70.9±3.4**	**67.0±5.0**	**78.2±16.2**	**86.9±5.5**	**67.7±23.0**	**83.6±7.5**

maximum accuracy on the clean validation set throughout the training. In addition, the *final accuracy* is also reported in the Appendix D *i.e.* the validation accuracy after training has converged at the final training step.

Training setting: On our benchmark, all methods are trained on the noisy training sets of two noise types (blue and red) under 10 noise levels (from 0% to 80%), and tested on the same clean validation set. Two training settings are considered: (i) training from scratch and (ii) fine-tuning from an ImageNet checkpoint where the checkpoint is pre-trained on the ImageNet training data. See details in the Appendix. On our datasets, Inception-ResNet-v2 (Szegedy et al., 2017) is used as the default backbone for all methods, where we upsample the images in the Mini-ImageNet dataset from 84x84 to 299x299 so that we can keep using the same pretrained ImageNet checkpoint. On the public benchmarks in Section 5.1.2, we train networks from scratch using ResNet-32 for CIFAR and Inception-ResNet-v2 for WebVision.

Baselines and our method: On our dataset, MentorMix is compared against the following baselines. We extensively search the hyperparameter for each method on every noise level. *Vanilla* is the standard training using l_2 weight decay, dropout, and data augmentation. **Weight Decay** and **Dropout** (Srivastava et al., 2014) are classical regularization methods. We search the hyperparameter for the weight decay in $\{e^{-5}, e^{-4}, e^{-3}, e^{-2}\}$ and the dropout ratio in $\{0.9, 0.8, 0.7, 0.6, 0.5\}$ as suggested in (Arpit et al., 2017). **Bootstrap** (Reed et al., 2014) corrects the loss with the learned label. The soft version is used and the hyperparameter for the learned label is tuned in $\{0.05, 0.25, 0.5, 0.7\}$. **S-model** (Goldberger & Ben-Reuven, 2017) is another way to "correct" the predictions by appending a new layer to a DNN to learn noise transformation. **MentorNet** (Jiang et al., 2018) is an example-weighting method. We employ the predefined MentorNet and search the hyperparameter p-percentile in $\{85\%, 75\%, 55\%, 35\%\}$. **Mixup** (Zhang et al., 2018) is a robust learning method that minimizes the empirical vicinal risk. Following the advice in (Zhang et al.,

2018) its hyperparameter α is searched in $\{1, 2, 4, 8\}$.

We implement MentorMix in Algorithm 1 in TensorFlow. We search two hyperparameters α in the γ_p in the range $\alpha = \{0.4, 1, 2\}$ and $\gamma_p = \{90\%, 80\%, 70\%\}$. The code will be released to reproduce our results.

5.1.1. BASELINE COMPARISON

We first show the comparison to the baseline methods on our dataset. For the lack of space, Table 2 shows a high-level summary of the comparison result, in which each cell shows a method's average best peak accuracy (and 95% confidence interval) across all 10 noise levels. Table 4 - Table 7 in the Appendix D list detailed results on each noise level.

As shown, the proposed method consistently outperforms the baseline methods across noise types (red and blue) and training settings (finetuning and training from scratch). This is desirable as baseline methods that work well on blue noise may not show consistent improvement over red noise, or vice versa. The red noise appears to be less harmful. Yet it is more difficult to overcome, which suggests the need for a new benchmark for a more comprehensive evaluation. Nevertheless, our method yields consistent improvements on both synthetic and web noisy labels.
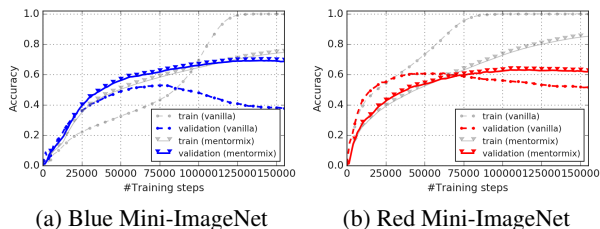

(a) Blue Mini-ImageNet　　(b) Red Mini-ImageNet

Figure 1. Comparison of training and validation accuracy during training. The dataset is Mimi-ImageNet at the 50% noise level.

The primary reason for our superior performance is that MentorMix can leverage MentorNet and Mixup in a complementary way. Technically, it uses MentorNet weight to identify examples with "cleaner" labels and encourages them to be used in the mixup operation. (Jiang et al., 2018)

showed that MentorNet optimizes an underlying robust loss in empirical risk minimization. From this perspective, our MentorMix introduces a new robust loss to minimize the vicinal risk which turns out to be more resilient to noisy training labels. For example, Fig. 1 compares Vanilla and MentorMix on the blue and red Mini-ImageNet at 50% noise level. It shows that MentorMix's loss is more robust to noisy labels, and MentorMix improves the peak accuracy of Vanilla by 16.3% on blue noise and 2.4% on red noise.

As the noise levels span across a wide range, we find the hyperparameters of robust learning methods are important. For the same method, a careful hyperparameter search could well be the difference between good and bad performance. As shown in the Appendix, we find that our method is relatively less sensitive to the choice of hyperparameters.

5.1.2. COMPARISON TO THE STATE-OF-THE-ART

In this subsection, we compare MentorMix on two public benchmarks of synthetic and real-world noisy labels. Table 3 compares with the state-of-the-art on the CIFAR dataset with symmetric label noise, where the top shows the classification accuracy on the clean validation set of CIFAR-100 and the bottom is for CIFAR-10. As all methods are trained using networks of similar capacity (ours is ResNet-32), we cite the numbers reported in their papers except for MentorNet and Mixup. Our method achieves the best accuracy across all noise levels. The results validate that our method is effective for synthetic noisy labels.

Table 3. Comparison with the state-of-the-art in terms of the validation accuracy on CIFAR-100 (top) and CIFAR-10 (bottom).

Data	Method	Noise level (%)			
		20	40	60	80
CIFAR100	Arazo et al. (2019)	73.7	70.1	59.5	39.5
	Zhang & Sabuncu (2018)	67.6	62.6	54.0	29.6
	MentorNet (2018)	73.5	68.5	61.2	35.5
	Mixup (2018)	73.9	66.8	58.8	40.1
	Huang et al. (2019)	74.1	69.2	39.4	-
	Ours (MentorMix)	**78.6**	**71.3**	**64.6**	**41.2**
CIFAR10	Arazo et al. (2019)	94.0	92.8	90.3	74.1
	Zhang & Sabuncu (2018)	89.7	87.6	82.7	67.9
	Lee et al. (2019)	87.1	81.8	75.4	-
	Chen et al. (2019)	89.7	-	-	52.3
	Huang et al. (2019)	92.6	90.3	43.4	-
	MentorNet (2018)	92.0	91.2	74.2	60.0
	Mixup (2018)	94.0	91.5	86.8	76.9
	Ours (MentorMix)†	**95.6**	**94.2**	**91.3**	**81.0**

We then compare with the state-of-the-art on the challenging benchmark WebVision 1.0 (Li et al., 2017a) that contains 2.4 million training images of noisy labels from the web. It uses the same 1,000 classes in the ImageNet ILSVRC12 challenge and thus is evaluated on the same validation set. Following prior studies, we train our method on both the full training set (2.4M images on 1K classes) and the mini subset (61K images on 50 classes), and test it on two clean

validation sets from ImageNet ILSVRC12 and WebVision.

Table 4 shows the comparison results, where the method marked by † uses extra clean labels during training. As shown, the proposed MentorMix improves the prior state-of-the-art by about 3% in the top-1 accuracy on the ILSVRC12 validation set without using any extra labels. It is worth noting that 3% is a significant improvement on the ILSVRC12 validation (Deng et al., 2009). To the best of our knowledge, it achieves the best-published result on the WebVision 1.0 benchmark under the same training setting. The results show that our method is effective for real-world noisy labels. We also apply our method on the Clothing-1M dataset where we train only on the 1M noisy training examples. Our model gets 74.3% accuracy which is competitive to recent published works.

Table 4. Comparison with the state-of-the-art on the clean validation set of ILSVRC12 and WebVision. The number outside (inside) the parentheses denotes the top-1 (top-5) classification accuracy(%). † marks the method trained using extra clean labels.

Data	Method	ILSVRC12	WebVision
Full	Lee et al. (2018)†	61.0(82.0)	69.1(86.7)
Full	Vanilla	61.7(82.4)	70.9(88.0)
Full	MentorNet (2018)†	64.2(84.8)	72.6(88.9)
Full	Guo et al. (2018)†	64.8(84.9)	72.1(89.2)
Full	Saxena et al. (2019)	—	67.5(——)
Full	Ours (MentorMix)	**67.5(87.2)**	**74.3(90.5)**
Mini	MentorNet (2018)	63.8(85.8)	—
Mini	Chen et al. (2019)	61.6(85.0)	65.2(85.3)
Mini	Ours (MentorMix)	**72.9(91.1)**	**76.0(90.2)**

5.2. Understanding DNNs trained on noisy labels

In this subsection, we conduct a large study into understanding DNNs trained on noisy labels across noise levels, noise types, training settings, and network architectures. We focus on three important findings (Zhang et al., 2017; Arpit et al., 2017; Kornblith et al., 2019). These works examine vanilla DNN training either on controlled synthetic labels (the former two) or on clean training labels (the last one). Our goal is to revisit them on our benchmark in a controlled setting where the noise in training sets varies from completely clean (0%) to the level where 80% of training labels are incorrect.

As in these works, we learn DNNs using vanilla training which allows us to compare their findings. Training DNNs using robust learning methods would probably lead to different findings but that would make the findings undesirably depend on the specific method being used. Regarding the network architectures, by default we use Inception-ResNet-v2 and also compare six other architectures: EfficientNet-B5 (Tan & Le, 2019), MobileNet-V2 (Sandler et al., 2018), ResNet-50 and ResNet-101 (He et al., 2016), Inception-V2 (Ioffe & Szegedy, 2015), and Inception-V3 (Szegedy et al., 2016). We select the above architectures to be representative of diverse capacities, the accuracy of which on the
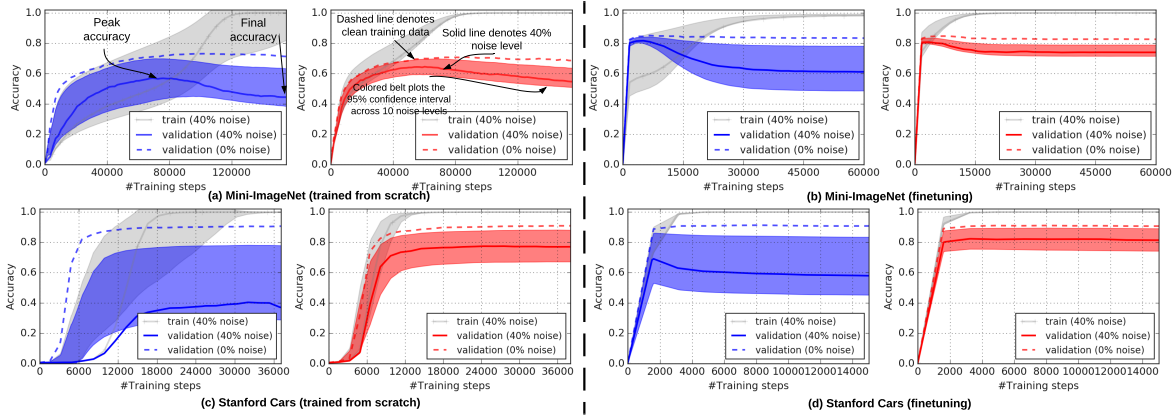
Figure 2. DNNs trained on synthetic (blue) and web label noise (red) on Mini-ImageNet (top) and Stanford Cars (bottom).

ILSVRC 2012 validation covers a wide range from 71.6% to 83.6%. See more details in the Appendix B.

DNNs generalize much better on red label noise. Zhang et al. (2017) found that the generalization performance of DNNs drops sharply as the level of noisy training labels increases. This pivotal finding that DNNs generalize poorly on noisy training labels has influenced many works. We first confirm Zhang et al. (2017)'s finding on blue noise in Fig 2, where the training and validation accuracies are shown along with the training steps. The dashed and solid curves represent the validation accuracies for 0% (or clean) and 40% noise levels, respectively, and the color belt plots the 95% confidence interval of the 10 noise levels.

After the training converges in Fig 2, the difference between the dashed and solid curves (in blue) indicates a palpable performance degradation between the clean (0%) and noisy (40%) labels. This can also be seen from the greater width of the blue belt which denotes the accuracy's confidence interval over all 10 noise levels. This confirms Zhang et al. (2017)'s finding on synthetic noisy labels. However, the difference is much smaller on the red curves, suggesting DNNs generalize much better on the red noise. This pattern is consistent across our two datasets using both fine-tuning and training from scratch. We hypothesize that DNNs are more robust to web labels because they are more relevant (visually or semantically) to the clean training images. See Fig. 1 in the Appendix.
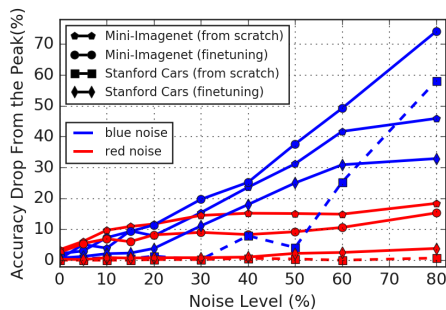

Figure 3. Performance drop from the peak accuracy at different noise levels. Colors are used to differentiate noise types.

DNNs may not learn patterns first on red label noise. Arpit et al. (2017) found that DNNs learn patterns first, revealing an interesting property that DNNs are able to automatically learn generalizable "patterns" in the early training stage before memorizing all noisy training labels. This can be manifested by the gap between the peak and final accuracy, as shown in Fig. 2(a). A larger drop suggests a better pattern is found during the early training stage. For better visualization, Fig. 3 computes the relative difference, namely the drop, between the peak and final accuracy across noise levels. As shown, the blue curves show a significant drop as the noise level grows. This is consistent with Arpit et al. (2017)'s finding on synthetic label noise.

Interestingly, the drop on the red noise is considerably smaller and even approaches zero on the Stanford Cars dataset. This suggests DNNs may not learn patterns first on red label noise at least for the fine-grained classification task. Our hypothesis is that images of real-world label noise are more complicated than those of the synthetic noise because they are sampled non-uniformly from an infinite number of classes. Therefore, it is much more difficult for DNNs to capture meaningful patterns automatically.

ImageNet architectures generalize on noisy labels when the networks are fine-tuned. Kornblith et al. (2019) found that fine-tuning better architectures trained on ImageNet tend to perform better on downstream tasks of clean training labels. It is important to verify whether this holds on noisy training labels because if so, one can conveniently transfer better architectures to better overcome the noisy labels.

Following (Kornblith et al., 2019), in Fig. 4, we compare the fine-tuning performance using ImageNet architectures, where the x-axis is the accuracy of the pretrained architectures on ImageNet, and the y-axis denotes the peak accuracy on our datasets. The bar plots the 95% confidence interval across 10 noise levels, where the center dot marks the mean. As it shows, there is a reasonable correlation between the ImageNet accuracy and the validation accuracy on both red and blue noisy labels. The Pearson correlation for the red noise
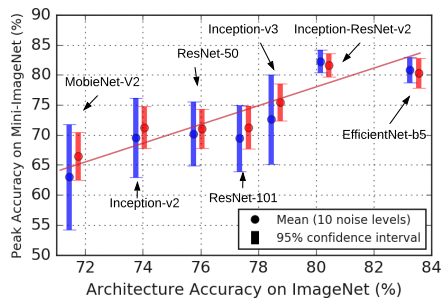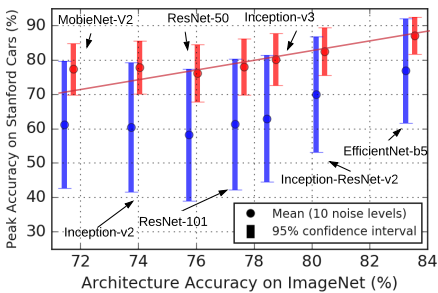
(a) Mini-ImageNet ($r = 0.91$)

(b) Stanford Cars ($r = 0.88$)

Figure 4. Fine-tuning seven ImageNet architectures on the red and blue datasets. The number in parentheses is the Pearson correlation between the architecture's ImageNet accuracy and the fine-tuning accuracy on our dataset of red noise.

is 0.91 on Mini-ImageNet and 0.88 on Stanford Cars. This indicates a better-pretrained architecture is likely to perform better even when it is fine-tuned on noisy training labels. We do not find such correlation when these architectures are trained from scratch. These results extend Kornblith et al. (2019)'s finding to noisy training data, and suggest when possible one may use more advanced pretrained architectures to overcome noisy training labels.

6. Conclusions

In this paper, we study web label noise in a controlled setting. We make three contributions. First, we establish the first benchmark of controlled web noise obtained from image search. Second, a simple but effective method is proposed to overcome both synthetic and real-world noisy labels. Our method achieves state-of-the-art results on multiple datasets.

Finally, we conduct the largest study by far into understanding deep learning on noisy data across a variety of settings. Our studies reveal several new findings: (1) DNNs generalize much better on web label noise; (2) DNNs may not learn patterns first on web label noise in which early stopping may not be very effective; (3) when networks are fine-tuned, ImageNet architectures generalize well on both symmetric and web label noise; (4) methods that perform well on synthetic noise may not work as well on the real-world noisy labels from the web; (5) the proposed method yields consistent improvements on both synthetic and real-world noisy labels

from the web.

Based on our observations, we arrive at the following recommendations for training deep neural networks on noisy training data.

- A simple way to deal with noisy labels is to fine-tune a model that is pre-trained on clean datasets, like ImageNet. The better the pre-trained model is, the better it may generalize on downstream noisy training tasks.

- Early stopping may not be effective on the label noise from the web, especially on the fine-grained classification task.

- Methods that perform well on synthetic noise may not work as well on the real-world noisy labels from the web. The proposed MentorMix can better overcome both synthetic and real-world web noisy labels.

- The real-world label noise from the web appears to be less harmful, yet it is more difficult for our current robust learning methods to tackle. This encourages more future research to be carried out on controlled label noise from the web.

Acknowledgements

References

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., and McGuinness, K. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning (ICML)*, 2019.

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

Azadi, S., Feng, J., Jegelka, S., and Darrell, T. Auxiliary image regularization for deep cnns with noisy labels. In *International Conference on Learning Representations (ICLR)*, 2016.

Bootkrajang, J. and Kabán, A. Label-noise robust logistic regression and its applications. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2012.

Charoenphakdee, N., Lee, J., and Sugiyama, M. On symmetric losses for learning from corrupted labels. *International Conference on Machine Learning (ICML)*, 2019.

Chen, P., Liao, B., Chen, G., and Zhang, S. Understanding and utilizing deep neural networks trained with noisy labels. *International Conference on Machine Learning (ICML)*, 2019.

Chen, X. and Gupta, A. Webly supervised learning of convolutional networks. In *International Conference on Computer Vision (ICCV)*, 2015.

Cheng, Y., Jiang, L., and Macherey, W. Robust neural machine translation with doubly adversarial inputs. In *Annual Conference of the Association for Computational Linguistics (ACL)*, 2019.

Cheng, Y., Jiang, L., Macherey, W., and Eisenstein, J. Advaug: Robust data augmentation for neural machine translation. In *Annual Conference of the Association for Computational Linguistics (ACL)*, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Fan, Y., He, R., Liang, J., and Hu, B. Self-paced learning: an implicit regularization perspective. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

Goldberger, J. and Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Guo, S., Huang, W., Zhang, H., Zhuang, C., Dong, D., Scott, M. R., and Huang, D. Curriculumnet: Weakly supervised learning from large-scale web images. In *European Conference on Computer Vision (ECCV)*, 2018.

Han, B., Yao, J., Niu, G., Zhou, M., Tsang, I., Zhang, Y., and Sugiyama, M. Masking: A new perspective of noisy supervision. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018a.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018b.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. Using trusted data to train deep networks on labels corrupted by severe noise. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

Huang, J., Qu, L., Jia, R., and Zhao, B. O2u-net: A simple noisy label detection approach for deep neural networks. In *International Conference on Computer Vision (ICCV)*, 2019.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

Jiang, L., Meng, D., Yu, S.-I., Lan, Z., Shan, S., and Hauptmann, A. Self-paced learning with diversity. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

Jiang, L., Meng, D., Zhao, Q., Shan, S., and Hauptmann, A. G. Self-paced curriculum learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *International Conference on Machine Learning (ICML)*, 2018.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Krause, J., Deng, J., Stark, M., and Fei-Fei, L. Collecting a large-scale dataset of fine-grained cars. In *Second Workshop on Fine-Grained Visual Categorization*, 2013.

Krause, J., Sapp, B., Howard, A., Zhou, H., Toshev, A., Duerig, T., Philbin, J., and Fei-Fei, L. The unreasonable effectiveness of noisy data for fine-grained recognition. In *European Conference on Computer Vision (ECCV)*, 2016.

Kumar, M. P., Packer, B., and Koller, D. Self-paced learning for latent variable models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2010.

Lee, K., Yun, S., Lee, K., Lee, H., Li, B., and Shin, J. Robust inference via generative classifiers for handling noisy labels. *International Conference on Machine Learning (ICML)*, 2019.

Lee, K.-H., He, X., Zhang, L., and Yang, L. Cleannet: Transfer learning for scalable image classifier training with label noise. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Li, J., Wong, Y., Zhao, Q., and Kankanhalli, M. S. Learning to learn from noisy labeled data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Li, J., Socher, R., and Hoi, S. C. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Li, W., Wang, L., Li, W., Agustsson, E., and Van Gool, L. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017a.

Li, Y., Yang, J., Song, Y., Cao, L., Li, J., and Luo, J. Learning from noisy labels with distillation. In *International Conference on Computer Vision (ICCV)*, 2017b.

Liang, J., Jiang, L., Meng, D., and Hauptmann, A. G. Learning to detect concepts from webly-labeled video data. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

Liang, J., Jiang, L., and Hauptmann, A. Simaug: Learning robust representations from simulation for trajectory prediction. In *European Conference on Computer Vision (ECCV)*, 2020.

Ma, X., Wang, Y., Houle, M. E., Zhou, S., Erfani, S. M., Xia, S.-T., Wijewickrema, S., and Bailey, J. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning (ICML)*, 2018.

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and van der Maaten, L. Exploring the limits of weakly supervised pretraining. In *European Conference on Computer Vision (ECCV)*, 2018.

Mnih, V. and Hinton, G. E. Learning to label aerial images from noisy data. In *International Conference on Machine Learning (ICML)*, 2012.

Noh, H., You, T., Mun, J., and Han, B. Regularizing deep neural networks by noise: its interpretation and optimization. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

Northcutt, C. G., Jiang, L., and Chuang, I. L. Confident learning: Estimating uncertainty in dataset labels. *arXiv preprint arXiv:1911.00068*, 2019.

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Reeve, H. W. and Kabán, A. Fast rates for a knn classifier robust to unknown asymmetric label noise. *International Conference on Machine Learning (ICML)*, 2019.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning (ICML)*, 2018.

Rolnick, D., Veit, A., Belongie, S., and Shavit, N. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Saxena, S., Tuzel, O., and DeCoste, D. Data parameters: A new family of parameters for learning a differentiable curriculum. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Seo, P. H., Kim, G., and Han, B. Combinatorial inference against label noise. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Song, H., Kim, M., and Lee, J.-G. Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning (ICML)*, 2019.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958, 2014.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

Tan, M. and Le, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

Vahdat, A. Toward robustness against label noise in training deep discriminative neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

Van Rooyen, B., Menon, A., and Williamson, R. C. Learning with symmetric label noise: The importance of being unhinged. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., and Belongie, S. Learning from noisy large-scale datasets with minimal supervision. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

Wang, Y., Liu, W., Ma, X., Bailey, J., Zha, H., Song, L., and Xia, S.-T. Iterative learning with open-set noisy labels. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. Learning from massive noisy labeled data for image classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. *International Conference on Machine Learning (ICML)*, 2019.

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

Zhang, Z., Zhang, H., Arik, S. O., Lee, H., and Pfister, T. Distilling effective supervision from severe label noise. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.