Image Classification with Deep Learning in the Presence of Noisy Labels: A Survey

Görkem Algan*†, Ilkay Ulusoy*
*Middle East Technical University, Electrical-Electronics Engineering
{e162565, ilkay}@metu.edu.tr
†ASELSAN
galgan@aselsan.com.tr

***Abstract*—Image classification systems recently made a big leap with the advancement of deep neural networks. However, these systems require an excessive amount of labeled data in order to be trained properly. This is not always feasible due to several factors, such as expensiveness of labeling process or difficulty of correctly classifying data even for the experts. Because of these practical challenges, label noise is a common problem in datasets and numerous methods to train deep networks with label noise are proposed in the literature. Although deep networks are known to be relatively robust to label noise, their tendency to overfit data makes them vulnerable to memorizing even total random noise. Therefore, it is crucial to consider the existence of label noise and develop counter algorithms to fade away its negative effects to train deep neural networks efficiently. Even though an extensive survey of machine learning techniques under label noise exists, literature lacks a comprehensive survey of methodologies centered explicitly around deep learning in the presence of noisy labels. This paper aims to present these algorithms while categorizing them into one of the two subgroups: noise model based and noise model free methods. Algorithms in the first group aim to estimate the structure of the noise and use this information to avoid the negative effects of noisy labels during training. On the other hand, methods in the second group try to come up with algorithms that are inherently noise robust by using approaches like robust losses, regularizers or other learning paradigms.**

***Index Terms*—deep learning, label noise, classification with noise, noise robust, noise tolerant**

I. Introduction

Recent advancement in deep learning has led to great improvements on many different domains, such as image classification [1]–[3], object detection [4]–[6], semantic segmentation [7], [8] and others. Despite their impressive ability to generalize [9], [10], it is shown that these powerful models can memorize even complete random noise [11]. Various works are devoted to better explain this phenomenon [12], [13], yet regularizing deep neural networks (DNNs), while avoiding memorization, stays to be an important challenge. It gets even more crucial when there exists noise in data. Therefore, various methods are proposed in the literature to effectively train deep networks in the presence of noise.

There are two kinds of noise in literature, namely: feature and label noise [14]. Feature noise corresponds to corruption in the observed features of data, while label noise means the change of label from its true class. Even though both noise types may cause a significant decrease in performance [15], [16], label noise is considered to be more harmful [14], [17] and shown to deteriorate the performance of classification systems in a broad range of problems [14], [18]–[20]. This is due to several factors; label is unique for each data while features are multiple, and the importance of each feature varies while the label always has a significant impact [15], [20]. This work focuses on label noise; therefore, noise and label noise is used synonymously throughout the article.

The necessity of an excessive amount of labeled data for supervised learning is a major drawback since it requires an expensive dataset collection and labeling process. To overcome this issue, cheaper alternatives have emerged. For example, an almost unlimited amount of data can be collected from the web with the help of search engines or from social media. Similarly, the labeling process can be crowdsourced with the help of systems like Amazon Mechanical Turk (AMT) [1], Crowdflower [2], which decrease the cost of labeling notably. Another widely used approach is to label data with automated systems. However, all these approaches led to a common problem; label noise. Besides these methods, label noise can occur even in the case of expert annotators. Labelers may lack the necessary experience, or data can be too complex to be correctly classified even for the experts. Moreover, label noise can also be introduced to data for data adversarial poisoning purposes [21], [22]. Being a natural outcome of dataset collection and labeling makes label noise robust algorithms an essential topic for the development of efficient computer vision systems.

Supervised learning with label noise is an old phenomenon with three decades of history [23]. An extensive survey about relatively old machine learning techniques under label noise is available [15], [16]; however, no work is proposed to provide a comprehensive survey on classification methods centered around deep learning by label noise. This work focuses explicitly on filling this absence. Even though deep networks are considered to be relatively robust to label noise [9], [10], they have an immense capacity to overfit data [11]. Therefore, preventing DNNs to overfit noisy data is very important, especially for fail-safe applications, such as automated medical diagnosis systems. Considering the significant success of deep

[1] http://www.mturk.com
[2] http://crowdflower.com

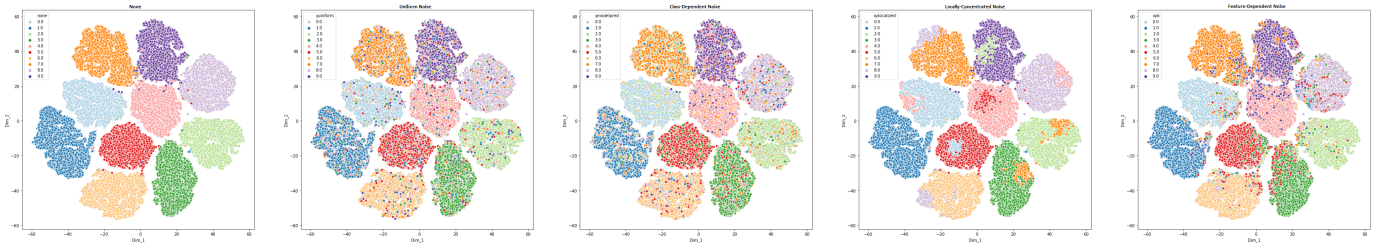

Fig. 1: T-SNE plot of data distribution of MNIST dataset in feature space for 25% noise ratio. a) clean data b) random noise c) y-dependent noise which is still randomly distributed in feature domain d) xy-dependent noise in locally concentrated form e) xy-dependent noise that is concentrated on decision boundaries

learning over its alternatives, it is a topic of interest, and many works are presented in the literature. Throughout the paper, these methods are briefly explained and grouped to provide the reader a clear overview of the literature.

This paper is organized as follows. Section II explains several concepts that are used throughout the paper. Proposed solutions in literature are categorized into two major groups, and these methods are discussed in section III - section IV. Finally, section V concludes the paper.

II. Preliminaries

In this section, firstly the problem statement for supervised learning in the presence of noisy labels is given. Secondly, types of label noises are presented and finally sources of label noise are discussed.

A. Problem Statement

Classical supervised learning consists of an input dataset $\mathcal{S} = \{(x_1, y_1), ..., (x_N, y_N)\} \in (X, Y)^N$ drawn according to an unknown distribution $\mathcal{D}$, over (X, Y). Task is to find the best mapping function $f : X \rightarrow Y$ among family of functions $\mathcal{F}$, where each function is parametrized by θ.

One way of evaluating the performance of a classifier is the so called loss function, denoted as $l : \mathcal{R} \times Y \rightarrow \mathcal{R}^+$. Given en example $(x_i, y_i) \in (X, Y)$, $l(f_\theta(x_i), y_i)$ evaluates how good is classifier prediction. Then, for any classifier f, expected risk is defined as follows

$$R_{l,\mathcal{D}}(f_\theta) = E_\mathcal{D}[l(f_\theta(x), y)] \tag{1}$$

where E denotes the expectation over distribution $\mathcal{D}$. Since it is not generally feasible to have complete knowledge over distribution $\mathcal{D}$, as an approximation, the empirical risk is used

$$\hat{R}_{l,\mathcal{D}}(f_\theta) = \frac{1}{N} \sum_{i=1}^{N} l(f_\theta(x_i), y_i) \tag{2}$$

Various methods of learning a classifier may be seen as minimizing the empirical risk subjected to network parameters

$$\theta^\star = \arg\min_\theta \hat{R}_{l,\mathcal{D}}(f_\theta) \tag{3}$$

In the presence of the noise, dataset turns into $\mathcal{S}_n = \{(x_1, \tilde{y}_1), ..., (x_N, \tilde{y}_N)\} \in (X, Y)^N$ drawn according to a noisy distribution $\mathcal{D}_n$, over (X, Y). Then, risk minimization results in

$$\theta_n^\star = \arg\min_\theta \hat{R}_{l,\mathcal{D}_n}(f_\theta) \tag{4}$$

As a result, obtained parameters by minimizing over $\mathcal{D}_n$ are different than desired optimal classifier parameters

$$\theta^\star \neq \theta_n^\star$$

Classical supervised learning aims to find the best estimator parameters $\theta^\star$ for given distribution $\mathcal{D}$, while iterating over $\mathcal{D}$. However, in noisy label setup, task is still finding $\theta^\star$ while working on distribution $\mathcal{D}_n$. Therefore, classical risk minimization is insufficient in the presence of label noise, since it would result in $\theta_n^\star$. As a result, variations of classical risk minimization methods are proposed in literature and they will be further evaluated in the upcoming sections.

B. Label Noise Models

A detailed taxonomy of label noise is provided in [15]. In this work, we follow the same taxonomy with a little abuse of notation. Label noise can be affected by three factors: data features, the true label of data, and the labeler characteristics. According to the dependence of these factors, label noise can be categorized into three subclasses.

Random noise is totally random and does not depend on either instance features nor its true class. With a given probability p_e label is changed from its true class. *Y-dependent noise* is independent of image features but depends on its class; $p_e = p(e|y)$. That means data from a particular classes are more likely to be mislabeled. For example, in a handwritten digit recognition task, ”3” and ”8” are much more likely to be confused with each other rather than ”3” and ”5”. *XY-dependent noise* depends on both image features and its class; $p_e = p(e|x, y)$. As in the y-dependent case, objects from a particular class may be more likely to be mislabeled. Moreover, the chance of mislabeling may change according to data features. If an instance has similar features to another instance from another class, it is more likely to be mislabeled. All these types of noises are illustrated in Figure 1

What is not considered here is the case of multi-labeled data, in which each instance has multiple labels given by different annotators. In that scenario, works shows that modeling the characteristics of each labeler and using this information during training, significantly boosts the performance [24]. However, various characteristics of different labelers can be explained with given noise models. For example, in a crowd-sourced dataset, some labelers can be total spammers who label with a random selection [25]; therefore, they can be modeled as random noise. On the other hand, labelers with better accuracies then random selection can be modeled by y-dependent or xy-dependent noise. As a result, the characteristic of the labeler is not introduced as an extra ingredient in these definitions.

C. *Sources of Label Noise*

As mentioned, label noise is a natural outcome of dataset collection process and can occur in various domains, such as medical imaging [24], [26], [27], semantic segmentation [28]–[30], crowd-sourcing [31], social network tagging [32], financial analysis [33] and many more. This work focuses on various solutions to such problems, but it may be helpful to investigate the causes of label noise in order to understand the phenomenon better.

Firstly, with the availability of the immense amount of data on the web and social media, it is a great interest of computer vision community to make use of that [34]–[39]. However, labels of these data are coming from messy user tags or automated systems used by search engines. These processes of obtaining datasets are well known to result in noisy labels.

Secondly, the dataset can be labeled by multiple experts resulting in a multi-labeled dataset. Each labeler has a varying level of expertise, and their opinions may commonly conflict with each other, which results in noisy label problem [25]. There are several reasons to get data labeled by more than one expert. Opinions of multiple labelers can be used to double-check each other's predictions for challenging datasets, or crowd-sourcing platforms can be used to decrease the cost of labeling for big data; such as Amazon Mechanical Turk, Crowdflower and more. Despite its cheapness, labels obtained from non-experts are commonly noisy with a differentiating rate of error. Some labelers even can be a total spammer who labels with random selection [25].

Thirdly, data can be too complicated for even the experts in the field, e.g., medical imaging. For example, to collect gold standard validation data for retinal images, annotations are gathered from 6-8 different experts [40], [41]. This can be due to the subjectiveness of the task for human experts or the lack of experience in annotator. Considering the fields where the true diagnosis is of crucial importance, overcoming this noise is of great interest.

Lastly, label noise can intentionally be injected in purpose of regularizing [42] or data poisoning [21], [22].

D. *Methodologies*

There are many possible ways to group proposed methods in the literature. For example, one possible way to distinguish algorithms is according to their need for a noise-free subset of data or not. Alternatively, they can be divided according to the noise type they are dealing with, or label type such as singly-labeled or multi-labeled. However, these are not handy to understand the main approaches behind the proposed algorithms; therefore, different sectioning is proposed as noise model based and noise model free methods.

Noise model based methods aim to model the noise structure so that this information can be used during training to come through noisy labels. In general, approaches in this category aim to extract noise-free information contained withing the dataset by either neglecting or de-emphasizing information coming from noisy samples. Furthermore, some methods attempt to reform the dataset by correcting noisy labels to increase the quality of the dataset for the classifier. The performance of these methods is heavily dependent on the accurate estimate of the underlying noise. The advantage of noise model based methods is the decoupling of classification and label noise estimation, which helps them to work with the classification algorithm at hand. Another good side is in the case of prior knowledge about the noise structure, noise model based methods can easily be head-started with this extra information inserted to the system.

Differently, noise model free methods aim to come up with inherently noise robust methods without explicit modeling of the noise structure. These kinds of approaches assume that classifier is not too sensitive to the noise, and performance degradation is a result of overfitting. Therefore, the main focus is given to overfit avoidance by regularizing the network training procedure.

Both of the mentioned approaches are discussed and further categorized in section III and section IV. Table I presents all mentioned methods to provide a clear picture as a whole. It should be noted that most of the time there are no sharp boundaries among the methods, and they may belong to more than one category. However, for the sake of integrity, they are placed in the subclass of most resemblance.

III. Noise Model Based Methods

In the presence of noisy labels, the task is to find the best estimator for hidden distribution $\mathcal{D}$, while iterating over distribution $\mathcal{D}_n$. If the mapping function $M : \mathcal{D} \to \mathcal{D}_n$ is known, it can be used to reverse the effect of noisy samples. Algorithms under this section simultaneously try to find underlying noise structure and train the base classifier with estimated noise parameters. They need a better estimate of M to train better classifiers and better classifiers to estimate M accurately. Therefore, they usually suffer from a chicken-egg problem. Approaches belonging to this category are explained in the following subsections.

A. *Noisy Channel*

The general setup for the noisy channel is illustrated in Figure 2. Methods belonging to this category minimize the following risk

$$\hat{R}_{l,\mathcal{D}}(f) = \frac{1}{N}\sum_{i=1}^{N} l(Q(f_\theta(x_i)), \tilde{y_i}) \tag{5}$$

Noise Model Based Methods	**1. Noisy Channel** *a.Explicit calculation*: predictions on noisy data [43], predictions on clean data [44], easy data [45] *b.Iterative calculation*: EM [26], [46], [47], fully connected layer [48], anchor point estimate [49], Drichlet-distribution [50] *c.Complex noisy channel*: noise type estimation [51], relevance estimation [52]
	2. Label Noise Cleansing *a.Using data with clean labels*: train on clean set [53], ensemble [54], graph-based [55] *b.Using data with both clean and noisy labels*: iteratively correct [56], correct for fine-tune [57] *c.Using data with just noisy labels*: calculate posterior [58], posterior with compatibility [59], consistency with model [60], [61], ensemble [62], prototypes [63], quality embedding [64], partial labels [65]
	3. Dataset Pruning *a.Data pruning* according to noise rate [66], transfer learning [67], cyclic state [68] *b.Label pruning* semi-supervised learning [69]–[71], relabeling [72]–[74]
	4. Sample Choosing a.*Curriculum Learning*: Screening loss [75], teacher-student [76], selecting uncertain samples [77], curriculum loss [78], data complexity [79], consistency with model [80] b.*Multiple Classifiers*: Consistency of networks [81], co-teaching [82]–[85]
	5. Sample Importance Weighting Meta task [86]–[88], siamese network [89], pLOF [27], abstention [90], estimate noise rate [91], [92], similarity loss [93], transfer learning [94], θ-distribution [95]
	6. Labeler Quality Assessment EM [25], [96], [97], trace regularizer [98], crowd-layer [99], image difficulty estimate [100], consistency with network prediction [101], omitting probability variable [102] , softmax layer per labeler [24]
Noise Model Free Methods	**1. Robust Losses** Non-convex loss functions [103]–[105], 0-1 loss surrogate [106], MAE [107], IMEA [108], Generalized cross-entropy [109], symmetric loss [110], unbiased estimator [111], modified cross-entropy for omission [112], information theoric loss [113], linear-odd losses [114], classification calibrated losses [115], SGD with robust losses [116]
	2. Meta Learning Choosing best methods [117], pumpout [118], noise tolerant parameter initialization [119], knowledge distillation [120], [121], gradient magnitude adjustment [122], [123]
	3. Regularizers Dropout [124], adversarial training [125], mixup [126], label smoothing [127], [128], pre-training [129], dropout on final layer [124], checking dimensionality [130], auxiliary image regularizer [131]
	4. Ensemble Methods LogitBoost&BrownBoost [132], noise detection based AdaBoost [133], rBoost [134], RBoost1&RBoost2 [135], robust multi-class AdaBoost [136]
	5. Others Complementary labels [137], [138], autoencoder reconstruction error [139], minimum covariance determinant [140], less noisy data [141], data quality [142], prototype learning [143], [144], multiple instance learning [145], [146]

TABLE I: Existing methods to deal with label noise in the literature

where $Q(f_\theta(x_i)) = p(\tilde{y}_i|f_\theta(x_i))$ is the mapping from network predictions to given noisy labels. If Q adapts the noise structure $p(\tilde{y}|y)$, then network will be forced to learn true mapping $p(y|x)$.

Q can be formulated with a *noise transition matrix* T so that $Q(f_\theta(x_i)) = Tf_\theta(x_i)$ where each element of the matrix represents the transition probability of given true label to noisy label, $T_{ij} = p(\tilde{y} = j|y = i)$. Since T is composed of probabilities, weights coming from a single node should sum to one $\sum_j T_{ij} = 1$. This procedure of correcting predictions to match given label distribution is also called *loss-correction* [43].

A common problem in noisy channel estimation is scalability. As the number of classes increases, the size of the noise transition matrix increases exponentially, making it intractable to calculate. This can be partially avoided by allowing connections only among the most probable nodes [47] or predefined nodes [147]. These restrictions are determined by human experts, which allows additional noise information to be inserted into the training procedure.

The noisy channel is used only in the training phase. In the evaluation phase, the noisy channel is simply removed to get noise-free predictions of the base classifier. In these kinds of approaches, performance heavily depends on accurate estimation of noisy channel parameters; therefore, works mainly focus on the estimation of Q. Various ways of formulating the noisy channel are explained below.

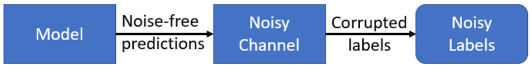

Fig. 2: Noise can be modeled as a noisy channel on top of base classifier.

1) Explicit calculation: Noise transition matrix is calculated explicitly, and then base classifier is trained using this matrix. Assuming dataset is balanced in terms of clean representative samples and noisy samples, so that there exists samples for each class with $p(y = \tilde{y}_i|x_i) = 1$, [43] constructs T just based on noisy class probability estimates of a pre-trained model, so-called *confusion matrix*. A similar approach is followed in [44]; however, the noise transition matrix is calculated from confusion matrix of network on the clean subset of data. Two datasets are gathered in [45], namely: easy data and hard data. The classifier is first trained on the easy data to extract similarity relationships among classes. Afterward, the calculated similarity matrix is used as the noise transition matrix. For noisy data, another method proposed in [48] calculates the confusion matrix on both noisy data and clean data. Then, the difference between these two confusion matrices gives T.

2) Iterative calculation: Noise transition matrix is estimated incrementally during training of the base classifier. In [46], [47] expectation-maximization (EM) [148] is used to iteratively train network to match given distribution and estimate noise transition matrix given the model prediction. The same approach is used on medical data with noisy labels in [26]. [149] and [48] add a linear fully connected layer as a last layer of the base classifier, which is trained to adapt noise behavior. In order to avoid this additional layer to converge the identity matrix and base classifier overfitting the noise, the weight decay regularizer is applied to this layer. [49] suggests using class probability estimates on anchor points, data points that belong to a specific class almost surely, to construct the noise transition matrix. In the absence of a noise-free subset of data, anchor points are extracted from data points with high noisy class posterior probabilities. Then, the matrix is updated iteratively to minimize loss during training. Instead of using softmax probabilities, [50] models noise transition matrix in Bayesian form by projecting it into a Dirichlet-distributed space.

3) Complex noisy channel: Different then simple confusion matrix, some works formalize the noisy channel as a more complex function. This enables noisy channel parameters to be calculated not just by using network outputs, but additional information about the content of data. For example, three types of label noises are defined in [51], namely: no noise, random noise, structured noise. An additional convolutional neural network (CNN) is used to interpret the noise type of each sample. Finally, the noisy layer aims to match predicted labels to noisy labels with the help of predicted noise type. Another work in [52] proposes training an extra network as a relevance estimator, which attains the label's relevance to the given instance. Predicted labels are mapped to noisy labels with the consideration of relevance. If relevance is low, in case of noise, the classifier can still make predictions of true class and doesn't get penalized much for it.

B. Label Noise Cleansing

An obvious solution to noisy labels is to identify and correct suspicious labels to their corresponding true class. Cleaning the whole dataset manually can be costly; therefore some works propose to pick only suspicious samples to be sent to a human annotator for the purpose of reducing the cost [39]. However, this is still not a scalable approach, and as a result, various algorithms are proposed in the literature. Including the label correction algorithm, the empirical risk takes the following form

$$\hat{R}_{l,\mathcal{D}}(f) = \frac{1}{N}\sum_{i=1}^{N} l(f_\theta(x_i), G(\tilde{y_i}, x_i)) \tag{6}$$

where $G(\tilde{y_i}, x_i) = p(y_i|\tilde{y_i}, x_i)$ represents the label cleansing algorithm. Label cleansing algorithms rely on a feature extractor to map data to feature domain for the investigation of noisiness. While some works use a pretrained network as feature extractor, others use base classifier as it gets more and more accurate during training. This results in an iterative framework: as classifier gets better label cleansing is more accurate, and as label quality gets better base classifier gets better. From this point of view, label cleansing can be viewed as a dynamically evolving component of the system, instead of a preprocessing of data. Such methods usually tackle the difficulty of distinguishing informative hard samples from those with noisy labels [15]. As a result, they can end up removing too many samples or changing labels in a delusional way. Approaches for label cleansing can be separated according to their need for clean data or not.

1) Using data with clean labels: In the existence of a clean subset of data, the aim is to fuse noise-free label structure to noisy labels for correction. If the clean subset is large enough to train a network, one obvious way is to relabel noisy labels by predictions of the network trained on clean data. For relabeling, [53] uses alpha blending of given noisy labels and predicted labels. An ensemble of networks which are trained with different subsets of dataset are used in [54]. If they all agree on the label, it is changed to the predicted label; otherwise, the label is set to a random label. Instead of keeping the noisy label, setting it randomly helps to break the structure in noise and makes noise more uniformly distributed in label space. In [55] a graph-based approach is used, where relation among noisy labels and clean labels are extracted by a conditional random field.

2) Using data with both clean and noisy labels: Some works rely on a subset of data, for which both clean and noisy labels are provided. Then label noise structure is extracted from these conflicting labels and used to correct noisy data. In [56], the label cleaning network gets two inputs: extracted features of instances by the base classifier and corresponding noisy labels. Label cleaning network and base classifier trained jointly, so that label cleansing network learns to correct labels on the clean subset of data and provides corrected labels for base classifier on noisy data. Same approach is decoupled in [57] in teacher-student manner. First, the student is trained on noisy data. Then features are extracted from clean data via the student model, and the teacher learns the structure of noise depending on these extracted features. Afterward, the teacher predicts soft labels for noisy data, and the student is again trained on these soft labels for fine-tuning.

3) Using data with just noisy labels: Noise-free data is not always available, so the main approach in this situation is to incrementally estimate cleaner posterior label distribution. However, there is a possible undesired solution to this approach so that all labels are attained to a single class and base network predicting constant class, which would result in top delusional training accuracy. Therefore additional regularizers are commonly used to make label posterior distributed evenly. A joint optimization framework for both training base classifier and propagating noisy labels to cleaner labels is presented in [58]. Using expectation-maximization, both classifier parameters and label posterior distribution is estimated in order to minimize the loss. A similar approach is used in [59] with additional compatibility loss condition on label posterior. Considering noisy labels are in the minority, this term assures posterior label distribution is not diverged too much from given noisy label distribution, so that majority of the clean label contribution is not lost. [60] deploys a confidence policy where labels are determined by either network output or given noisy labels. With the increasing number of epochs, more confidence is given to the network output, since it gets better at estimating labels. In [61] arguing that, in case of noisy labels, model first learns correctly labeled data and then overfits to noisy data, the probability of a sample being noisy or not can be extracted from its loss value. In order to achieve this, the loss of each sample is fitted by a beta mixture model, which in turn models the label noise in an unsupervised manner. [62] proposes a two-level approach. In the first stage, with any chosen inference algorithm, the ground truth labels are determined, and data is divided into two subsets as noisy and clean. In the second stage, an ensemble of weak classifiers are trained on clean data to predict true labels of noisy data. After that, these two subsets of data are merged to create the final enhanced dataset. [63] constructs prototypes that are able to represent deep feature distribution of the corresponding class, for each class. Then corrected labels are found by checking similarity among data samples and prototypes. [64] introduces a new parameter, namely *quality embedding*, which represents the trustworthiness of data and is estimated by a neural network. Depending on two latent variables, true class probabilities and quality embedding, an additional network tries to extract the true class of each instance. In multi-labeled dataset, where each instance has multiple labels representing its content, some labels may be partially missing resulting in partial labels. In the case of partial labels, [65] uses one network to find and estimate easy missing labels and other network to be trained on this corrected data. [150] formulates video anomaly detection as a classification with label noise problem and trains a graph convolutional label noise cleaning network depending on features and temporal consistency of video snippets.

C. Dataset Pruning

Instead of correcting noisy labels to their true classes, an alternative approach is to remove them. While this would result in loss of information, preventing negative impact of noise may result in better performance. In these methods, there is a risk of removing too many samples. Therefore, it is crucial to remove as few samples as possible to prevent unnecessary data loss.

One option is to remove noisy data completely, and prune dataset to a smaller clean dataset. In [66], with the help of a probabilistic classifier, training data divided into two subsets: confidently clean and noisy. Noise rates are estimated according to sizes of these subsets. Finally, relying on output confidence of base network on data instances, number of most un-confident samples are removed in accordance with noise rate estimated. In [67], transfer learning is used, so that network trained on a clean dataset from a similar domain is fine-tuned on the noisy dataset for relabeling. Afterward, the network is again trained on relabeled data to re-sample the dataset to construct a final clean dataset. In [68], learning rate is adjusted cyclicly to change network status between underfitting and overfitting. Since, while underfitted, noisy samples cause more loss, samples with large noise during this cyclic process are removed.

An alternative option is to just remove labels while keeping data instances. This procedure results in labeled and unlabeled data, as a result semi-supervised learning methods [151] can be used to train on this new dataset [69] or relabel unlabeled data [72]. Alternatively, label removing can be done iteratively in each epoch to dynamically update dataset for better utilization of semi-supervised learning. [70] uses consistency among given label and moving average of model predictions to evaluate if the given label is noisy or not. Then model is trained on clean samples on the next iteration. This procedure continues until convergence to the best estimator. Same approach is used in [71] with a little tweak. Instead of comparing with given labels, moving average of predictions are compared with predicted labels in the current epoch.

Another approach in this class is to train a network on labeled and unlabeled data, and then use it to relabel noisy data [74]. Assuming that correctly labeled data account for majority, [73] proposes to randomly split dataset to labeled and unlabeled subgroups. Then, labels are propagated to unlabeled data using similarity index among instances. This procedure repeated to produce multiple labels per instance and then final label is set with majority voting.

D. Sample Choosing

A widely used approach to overcome label noise is to manipulate the input stream to the classifier. Guiding the network with choosing the right instances to feed can help classifier finding its way easier in the presence of noisy labels.

$$\hat{R}_{l,\mathcal{D}}(f) = \frac{1}{N}\sum_{i=1}^{N} V(x_i, y_i) l(f_\theta(x_i), \tilde{y_i})) for V(x_i, y_i) \in \{0, 1\} \tag{7}$$

In the above formula, $V(x_i, y_i)$ is the binary operator that decides to use the given data x_i or not. If $V(x_i, y_i) = 1$ for all data, then it turns out to be classical risk minimization (7). If V happens to be a static function, which means choosing the same samples during whole training according to a predefined rule, then it turns out to be dataset pruning, as explained in subsection III-C. Differently, sample choosing methods continuously monitor the base classifier and select samples to be trained on for the next training iteration. The task can be seen as drawing a path through data that would mimic the noise-free distribution of $\mathcal{D}$. Since these methods operate outside of the existing system, they are easier to attach to the existing algorithm at hand by just manipulating the input stream. However, it is vital to keep the balance the same as dataset pruning so that system does not ignore unnecessarily large quantities of data. Additionally, these methods prioritize low loss samples, which results in a slow learning rate since hard informative samples are considered only in the later stages of training. Two major approaches under this group are discussed in the following paragraphs.

1) Curriculum Learning: Curriculum learning (CL) [152], inspired from human cognition, proposes to start from easy samples and go through harder samples to guide training. This is also called *self-paced learning* [153], when prior to sample hardness is not known and inferred from loss of current model on that sample. In noisy label framework, clean labeled data can be accepted as an easy task while noisily labeled data is a harder task. Therefore, the idea of CL can be transferred to label noise setup as starting from confidently clean instances and go through more likely to be noisier samples as the classifier gets better. Various screening loss functions are proposed in [75] to sort instances according to their noisiness level. Teacher-student approach is implemented in [76], where the task of the teacher is to choose most probably to be clean samples for the student. Instead of using a predefined curriculum, the teacher constantly updates its curriculum depending on the outputs from the student. Arguing that CL slows down the learning speed, since it focuses on easy samples, [77] suggests choosing uncertain samples, which are predicted incorrectly sometimes and correctly on others, in training. These samples assumed to be probably not noisy since noisy samples should be predicted incorrectly all the time. Arguing that it is hard to optimize 0-1 loss, *curriculum loss* that chooses samples with low loss values for loss calculation, is proposed as an upper bound for 0-1 loss in [78]. In [79], data is split into subgroups according to their complexities, which are extracted by a network pre-trained on the full dataset. Since less complex data groups expected to have more clean labels, training will start from less complex data and go through more complex instances as the network gets better. Next samples to be trained on can be chosen by checking the consistency of the label with the network prediction. In [80], if both label and model prediction of the given sample is consistent, it is used in the training set. Otherwise, the model has a right to disagree. Iteratively this provides better training data and better model. However, there is a risk of the model being too skeptical and choosing labels in a delusional way; therefore, consistency balance should be established.

2) Multiple Classifiers: Some works use multiple classifiers to help each other to choose the next batch of data to train on. This is different than the teacher-student approach since none of the networks is supervising the other but they rather help each other out. This can provide robustness since networks can correct each other's mistakes due to their differences in learned representations. For this setup to work, the initialization of the classifiers is important. They are most likely to be initialized with a different subset of the data. If they are both the same, then there happens no update since they will both agree to disagree with labels. In [81] label is assumed to be noisy if both networks disagree with the given label, and update on model weights happens only when the prediction of two networks conflicts. The paradigm of *co-teaching* is introduced in [82], where two networks select the next batch of data for each other. The next batch is chosen as the data batch, which has small loss values according to pair network. It is claimed that using one network accumulates the noise-related error, whereas two networks filter noise error more successfully. The idea of co-teaching is further improved by iterating over data where two networks disagree, to prevent two networks converging each other with increasing number of epochs [83], [84]. Another work using co-teaching first trains two networks on a selected subset for a given number of epochs and then moves to full dataset [85].

E. Sample Importance Weighting

Similar to sample choosing, training can be made more effective by assigning weights to instances according to their estimated noisiness level. This has an effect of emphasizing cleaner instances for better update on model weights. Following empirical risk is minimized by these algorithms

$$\hat{R}_{l,\mathcal{D}}(f) = \frac{1}{N}\sum_{i=1}^{N}\beta(x_i,y_i)l(f_\theta(x_i),\tilde{y_i})) \tag{8}$$

where $\beta(x_i,y_i)$ determines the instance dependent weight. If β would be binary, then formulation is the same with sample choosing, as explained in subsection III-D. Differently, here β is not binary and changes values for each instance. Just like in sample choosing algorithms, β is a dynamic function, which means weights for instances keep changing during the training. Therefore, it is commonly a challenge to prevent β changing too rapidly and sharply, such that it disrupts the stabilized training loop. Moreover, these methods commonly suffer from accumulated errors so that they can easily get biased towards a certain subset of data. There are various methods proposed to obtain optimal β to fade away the negative effects of noise.

The simplest approach would be, in case of availability of both clean and noisy data, weighting clean data more [48]. However, this utilizes information poorly; moreover, clean data is not always available. Works of [86] and [87], uses meta-learning paradigm to determine the weighting factor. In each iteration, gradient descent step on given mini-batch for weighting factor is performed, so that it minimizes the loss on clean validation data. A similar method is adopted in [88], but instead of implicit calculation of the weighting factor, multi layer perceptron (MLP) is used to estimate the weighting function. *Open-set noisy labels*, where data samples associated with noisy labels might belong to a true class that is not present in the training data, are considered in [89]. Siamese network is trained to detect noisy labels by learning discriminative features to apart clean and noisy data. Noisy samples are iteratively detected and pulled from clean samples. Then, each iteration weighting factor is recalculated for noisy samples, and the base classifier is trained on whole dataset. [27] also iteratively separates noisy samples and clean samples. On top of that, not to miss valuable information from minority and hard samples, noisy data are weighted according to their noisiness level, which is estimated by pLOF [154]. [90] introduces *abstention*, which gives option to abstain samples, depending on their loss value, with an abstention penalty. Therefore, the network learns to abstain from confusing samples, and with the abstention penalty, the tendency to abstain can be adjusted. In [91], weighting factor is conditioned on distribution of training data, $\beta(X, Y) = P_{\mathcal{D}}(X, Y)/P_{\mathcal{D}_n}(X, \tilde{Y})$. The same methodology is extended to the multi-class case in [92]. In [93], a feature extractor network is used to map instances to feature domain. Later, the weighting factor is determined by checking instance similarity to its representative class prototype in the feature domain. [94] formulates the problem as transfer learning where the source domain is noisy data and target domain is a clean subset of data. Then weighting in source domain is arranged in a way to minimize target domain loss. [95] uses θ values of samples in θ-distribution to calculate their probability of being clean and use this information to weight clean samples more in training.

F. Labeler Quality Assessment

As explained in subsection II-C, there can be several reasons for dataset to be labeled by multiple annotators. Each labeler may have different level of expertise and their labels may occasionally contradict with each other. This is a common case in crowd-sourced data [155]–[157] or datasets which requires high level of expertise such as medical imaging [19]. Therefore, modeling and using labeler characteristic can significantly increase performance [24].

In this setup there are two unknowns namely; noisy labeler characteristic and ground truth labels. One can estimate both with expectation-maximization algorithm [25], [96], [97]. If noise is assumed to be y-dependent, labeler characteristic can be modeled with noise transition matrix, just like subsection III-A. [98] adds a regularizer to loss function which is the sum of traces of annotator confusion matrices, in order to force sparsity on matrices. Similar approach is implemented in [99], where crowd-layer is added to the end of network. In [100], xy-dependent noise is also considered by taking image complexities into account as well. Human annotators and computer vision system are used mutually in [101], where consistency among predictions of these two components are used to evaluate the reliability of labelers. [102] deals with the noise when labeler omits a tag in the image. Therefore, instead of noise transition matrix for labelers, omitting probability variable is used, which is estimated together with true class using expectation-maximization algorithm. Separate softmax layers are trained for each annotator in [24] and an additional network to predict the true class of data depending on the outputs of labeler specific networks and features of data. This setup enables to model each labeler and their overall noise structure in separate networks.

G. Discussion

Noise model based methods are heavily dependent on the accurate estimate of the noise structure. This brings a dilemma. For better noise model one needs better estimators, and for better estimators it is necessary to have a better estimate of underlying noise. Therefore, many approaches can be seen as an expectation-maximization of both noise estimate and classification. However, it is essential to prevent the system diverging from reality, therefore regularizing noise estimates and not letting it getting delusional is important. In order to achieve this, works in literature commonly make assumptions about the underlying noise structure, which damages their applicability to different setups. On the other hand, this lets any prior information about the noise to be inserted to the system for an head-start. It is also useful to handle domain-specific noise. One another advantage of these algorithms is they decouple noise estimation and classification tasks. Therefore, they are easier to implement on an existing classification algorithm at hand.

IV. Noise Model Free Methods

These methods aim to achieve label noise robustness without explicitly modeling it, but rather designing robustness in the proposed algorithm. Noisy data is treated as anomaly and therefore these methods are in similar line with overfit avoidance. They commonly rely on internal noise tolerance of the classifier and aim to boost performance by regularizing undesired memorization of noisy data. Various methodologies are presented in the following subsections.

A. Robust Losses

A loss function is said to be noise robust if the classifier learned with noisy and noise-free data, both achieve the same classification accuracy [103]. Algorithms under this section aims to design loss function in such a way that the noise would not decrease the performance. However, it is shown that noise can badly affect the performance even for the robust loss functions [15]. Moreover, these methods treat both noisy and clean data in the same way, which prevents the utilization of any prior information over data distribution.

In [103], it is shown that certain non-convex loss functions, such as 0-1 loss, has noise tolerance much more than commonly used convex losses. Extending this work [104], [105] derives sufficient conditions for a loss function to be noise tolerant for uniform noise. Their work shows that, if the given loss function satisfies $\sum_k l(f_\theta(x), k) = C, \forall x \in X$ where C is a constant value, then loss function is tolerant to uniform noise. In this content, they empirically show that none of the standard convex loss functions has noise robustness while 0-1 loss has, up to a certain noise ratio. However, 0-1 loss is non-convex and non-differentiable; therefore, surrogate loss of 0-1 loss is proposed in [106], which is still noise sensitive. Widely used *categorical cross entropy (CCE)* loss is compared with *mean absolute value of error (MAE)* in the work of [107], where it is shown empirically that mean absolute value of error is more noise tolerant. [108] shows that the robustness of MEA is due to its weighting scheme. While CCE is sensitive to abnormal samples and produces bigger gradients in magnitude, MAE treats all data points equally, which would result in an underfitting of data. Therefore, *Improved mean absolute value of error (IMAE)*, which is an improved version of MAE, is proposed in [108], where gradients are scaled with a hyper-parameter to adjusts weighting variance of MAE. [109] also argues that MAE provides a much lower smaller learning rate than CCE; therefore, a new loss function is suggested, which combines the robustness of MAE and implicit weighting of CCE. With a tuning parameter, characteristic of the loss function can be adjusted in a line from MAE to CCE. Loss functions are commonly not symmetric, meaning that $l(f_\theta(x_i), y_i) \neq l(y_i, f_\theta(x_i))$. Inspired from the idea of symmetric KL-divergence, [110] proposes symmetric cross entropy loss $l_{SCE}(f_\theta(x_i), y_i) = l(f_\theta(x_i), y_i) + l(y_i, f_\theta(x_i))$ to battle noisy labels.

Given that noise prior is known, [111] provides two surrogate loss functions using the prior information about label noise, namely, unbiased and weighted estimator of the loss function. [112] considers asymmetric omission noise for the binary classification case, where the task is to find road pixels from a satellite map image. Omission noise makes the network less confident about its predictions, so they modified cross-entropy loss to penalize network less for making wrong but confident predictions since these labels are more likely to be noisy. Instead of using distance-based loss, [113] proposes to use information-theoretic loss, in which determinant based mutual information [158] between given labels and predictions are evaluated for loss calculation. Weakly supervised learning with noisy labels are considered in [114], and necessary conditions for loss to be noise tolerant are drawn. [115] shows that classification-calibrated loss functions are asymptotically robust to symmetric label noise. Stochastic gradient descent with robust losses are analyzed in general [116] and shown to be more robust to label noise than its counterparts.

B. Meta Learning

With the recent advancements of deep neural networks, the necessity of hand-designed features for computer vision systems are mostly eliminated. Instead, these features are learned autonomously via machine learning techniques. Even though these algorithms are able to learn complex functions on their own, there still remains many hand-designed parameters such as network architecture, loss function, optimizer algorithm and so on. Meta learning aims to eliminate these necessities by learning not just the required complex function for the task, but also learning the learning itself [159], [160]. In general, the biggest drawback of these methods is their computational cost. Since they require nested loops of gradient computations for each training loop, they are several times slower than straightforward training.

Designing a task beyond classical supervised learning in meta learning fashion has been used to deal with label noise as well. A meta task is defined as predicting the most suitable method, among family of methods, for a given noisy dataset in [117]. *Pumpout* [118] presents a meta objective as recovering the damage done by noisy samples by erasing their effect on model via *scaled gradient ascent*. As a meta learning paradigm, model-agnostic-meta-learning (MAML) [160] seeks for weight initialization, which can easily be fine-tuned. A similar mentality is used in [119] for noisy labels, which aims to find noise-tolerant model parameters that are less prone to noise under teacher-student training framework [161], [162]. Multiple student networks are fed with data corrupted by synthetic noise, and meta objective is defined to maximize consistency with teacher outputs, which are obtained from raw data without synthetic noise. Therefore, student networks are forced to find most noise robust weight initialization such that weight update will still be consistent after training an epoch on synthetically corrupted data. Then, final classifier weights are set as an exponential moving average of student networks. Alternatively, in the case of available clean data, a meta objective can be defined to utilize this information. The approach used in [120] is to train a teacher network in a clean dataset and transfer its knowledge to student network for the purpose of guiding training process in the presence of mislabeled data. They used *distillation* technique proposed in [163] for controlled transfer of knowledge from teacher to student. A similar methodology of using *distillation* together with label correction in human pose estimation task is implemented in [121]. In [122], [123] the target network is trained on excessive noisy data, and the confidence network is trained on clean subset. Inspiring from [159], the confidence network's task is to control the magnitude of gradient updates to the target network so that noisy labels are not resulting in updating gradients.

C. Regularizers

Regularizers are well known to prevent DNNs from overfitting noisy labels. From this perspective, these methods treat performance degradation due to noisy data as overfitting to noise. Even though this assumption is mostly valid in random noise, it may not be the case for more complex noises. Some widely used techniques are weight decay, dropout [124], adversarial training [125], mixup [126], label smoothing [127], [128]. [129] shows that pre-training has a regularization effect in the presence of noisy labels. In [164] an additional softmax layer is added, and dropout regularization is applied to this layer, arguing that it provides more robust training and prevents memorizing noise due to randomness of dropout [124]. [130] proposes a complexity measure to understand if the network starts to overfit. It is shown that learning consists of two steps: 1) dimensionality compression, that models low-dimensional subspaces which closely match the underlying data distribution, 2) dimensionality expansion, that steadily increases subspace dimensionality in order to overfit the data. The key is to stop before the second step. *Local intrinsic dimensionality* [165] is used to measure complexity of trained model and stop before it starts to overfit. [131] takes a pre-trained network on a different domain and fine-tunes it for the noisy labeled dataset. Groups of image features are formed, and group sparsity regularization is imposed so that model is forced to choose relative features and up-weights the reliable images.

D. Ensemble Methods

It is well known that bagging is more robust to label noise than boosting [166]. Boosting algorithms like AdaBoost puts too much weight on noisy samples, resulting in overfitting the noise. However, the degree of label noise robustness changes for the chosen boosting algorithm. For example, it is shown that BrownBoost and LogitBoost are more robust than AdaBoost [132]. Therefore, noise-robust alternatives of AdaBoost is proposed in literature, such as noise detection based AdaBoost [133], rBoost [134], RBoost1&RBoost2 [135] and robust multi-class AdaBoost [136].

E. Others

Complementary labels, defines classes that observations do not belong to. For example, in the case of ten classes, there is one true class for an instance and nine complementary classes. Since annotators are less likely to mislabel, some works propose to work in complementary label space [137], [138]. [139] uses reconstruction error of autoencoder to discriminate noisy data from clean data, arguing that noisy data tend to have bigger reconstruction error. In [140], first base model is trained with noisy data. An additional generative classifier is trained on top of feature space generated by the base model. By estimating its parameters with *minimum covariance determinant*, noise-robust decision boundaries are aimed to be found. In [141], a special setup is considered where dataset consists of noisy and *less-noisy* data for binary classification task. [142] aims to extract the quality of data instances. Assuming that the training dataset is generated from a mixture of target distribution and other unknown distributions, it estimates the quality of data samples by checking the consistency between generated and target distributions.

Prototype learning aims to construct prototypes, that can represent features of a class, in order to learn clean representations. Some works in the literature [143], [144] propose to create clean representative prototypes for noisy data, so that base classifier can be trained on them instead of noisy labels.

In multiple-instance learning, data are grouped in clusters, called bags, and each bag is labeled as positive if there is at least one positive instance in it and negative otherwise. The network is fed with a group of data and produces a single prediction for each bag by learning the inner discriminative representation of data. Since the group of images is used and one prediction is produced, existence of noisy labels along with true labels in a bag has less impact on learning. In [145], authors propose to effectively choose training samples from each bag by minimizing the total bag level loss. Extra model is trained in [146] as attention model, which chooses parts of the images to be focused on. Aim is to focus on few regions on correctly labeled image and not focus on any region for mislabeled images.

F. Discussion

Methods belonging to this category, in overall, treat noisy data as an anomaly. Therefore, they are in a similar line with overfit avoidance and anomaly detection. Even though this assumption may be quite valid for random noise, it loses its validity in case of more complicated and structured noises. Since noise modeling is not decoupled from classification task explicitly, proposed methods are, in the general sense, embedded into the existing algorithm. This prevents their quick deployment to the existing system at hand. Moreover, algorithms belonging to meta-learning and ensemble methods can be computationally costly since they require multiple iterations of training loops.

V. Conclusion

Throughout this paper, it is shown that label noise is an important obstacle to deal with in order to achieve desirable performance from real-world datasets. Besides its importance for supervised learning in practical applications, it is also an important step to collect datasets from the web [167], [168], design networks that can learn from unlimited web data with no human supervision [35]–[38]. Furthermore, beside image classification, there are more fields where dealing with mislabeled instances is important, such as generative networks [169], [170], semantic segmentation [28]–[30], sound classification [171] and more. All these factors make dealing with label noise an important step through self-sustained learning systems.

Different approaches to come through noisy label phenomenon are proposed in the literature. All methods have their advantages and disadvantages, so one can choose the most appropriate algorithm for the use case. However, in order to draw a generic line, we make the following suggestions. If the noise structure is domain-specific and there is prior information or assumption about its structure, noise model based methods are more appropriate. Among these models, one can choose the best appropriate method according to need. For example, if noise can be represented as noise transition matrix, noisy channel or labeler quality assessment for multi labeler case can be chosen. If the purpose is to purify the dataset as a preprocessing stage, then dataset pruning or label noise cleansing methods can be employed. Sample choosing or sample importance weighting algorithms are handy if instances can be ranked according to their informativeness on training. Different from noise model based algorithms, noise model free methods do not depend on any prior information about the structure of the noise. Therefore, they are easier to implement if noise is assumed to be random, and performance degradation is due to overfitting since they do not require the hassle of implementing an external algorithm for noise structure estimation. If there is no clean subset of data, robust losses or regularizers are appropriate options since they treat all samples the same. Meta-learning techniques can be used in the presence of a clean subset of data since they can easily be adapted to utilize this subset.

Even though an extensive amount of research is conducted for machine learning techniques [15], deep learning by noisy labels is certainly an understudied problem. Considering its dramatic effect on DNNs [11], there still are many open research topics in the field. For example, truly understanding the effects of label noise on deep networks can be a fruitful future research topic. As mentioned in [20], it is believed first layers of CNNs extract features from data, while last layers learn to interpret labels from these features. Understanding which part of the network is highly affected by label noise may help to achieve transfer learning effectively. Alternatively, the question of how to train in the existence of both attribute and label noise is an understudied problem with significant potential on practical applications [53]. [110] shows noisy labels degrades the learning, especially for hard samples. So instead of overfitting, that may be the reason for performance degradation, which is an open question to be answered in the future. Another possible research direction may be on the effort of breaking the structure of the noise to make it uniformly distributed in the feature domain [73]. This approach would be handy, where labelers have a particular bias.

A widely used approach for quick testing of proposed algorithms is to create noisy datasets by adding synthetic label noise to benchmarking toy datasets [172]–[176]. However, this prevents fair comparison and evaluation of algorithms since each work adds its own type of noise. Some large datasets with noisy labels are proposed in literature [9], [93], [177], [178]. These datasets are collected from the web, and labels are attained from noisy user tags. Even though these datasets provide a useful domain for benchmarking proposed solutions, their noise rates are mostly unknown and they are biased in terms of data distribution for classes. Moreover, one can not adjust the noise rate for testing under extreme or moderate conditions. From this perspective, we believe literature lacks a noisy dataset where a major part of it is verified; thus, noise rate can be adjusted as desired.

Very small attention is given to the learning from a noisy labeled dataset when there is a small amount of data. This can be a fruitful research direction considering its potential in fields where harvesting dataset is costly. For example, in medical imaging, collecting a cleanly annotated large dataset is not feasible most of the time [53], due to its cost or privacy of the data. Effectively learning from a small amount of noisy data with no ground truth can have a significant effect on autonomous medical diagnosis systems. Even though some pioneer researches are available [24], [26], [27], there is still much more to be explored.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[7] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3194–3203.

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

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

[10] A. Drory, S. Avidan, and R. Giryes, "How do neural networks overcome label noise?" 2018.

[11] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[12] D. Krueger, N. Ballas, S. Jastrzebski, D. Arpit, M. S. Kanwal, T. Maharaj, E. Bengio, A. Fischer, and A. Courville, "Deep nets don't learn via memorization," 2017.

[13] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio *et al.*, "A closer look at memorization in deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 233–242.

[14] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artificial intelligence review*, vol. 22, no. 3, pp. 177–210, 2004.

[15] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2014.

[16] B. Frénay, A. Kabán *et al.*, "A comprehensive introduction to label noise," in *ESANN*, 2014.

[17] R. Hataya and H. Nakayama, "Investigating cnns' learning representation under label noise," 2018.

[18] D. F. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artificial intelligence review*, vol. 33, no. 4, pp. 275–306, 2010.

[19] M. Pechenizkiy, A. Tsymbal, S. Puuronen, and O. Pechenizkiy, "Class noise and supervised learning in medical domains: The effect of feature extraction," in *19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)*. IEEE, 2006, pp. 708–713.

[20] D. Flatow and D. Penner, "On the robustness of convnets to training on noisy labels," 2017.

[21] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik, "Data poisoning attacks on factorization-based collaborative filtering," in *Advances in Neural Information Processing Systems*, 2016, pp. 1893–1901.

[22] J. Steinhardt, P. W. Koh, and P. Liang, "Certified defenses for data poisoning attacks," in *Advances in Neural Information Processing Systems*, vol. 2017-Decem, 2017, pp. 3518–3530.

[23] D. Angluin and P. Laird, "Learning from noisy examples," *Machine Learning*, vol. 2, no. 4, pp. 343–370, 1988.

[24] M. Y. Guan, V. Gulshan, A. M. Dai, and G. E. Hinton, "Who said what: Modeling individual labelers improves classification," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[25] A. Khetan, Z. C. Lipton, and A. Anandkumar, "Learning from noisy singly-labeled data," *arXiv preprint arXiv:1712.04577*, 2017.

[26] Y. Dgani, H. Greenspan, and J. Goldberger, "Training a neural network based on unreliable human annotation of medical images," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, 2018, pp. 39–42.

[27] C. Xue, Q. Dou, X. Shi, H. Chen, and P.-A. Heng, "Robust Learning at Noisy Labeled Medical Images: Applied to Skin Lesion Classification," in *arxiv.org*, 2019, pp. 1280–1283.

[28] Z. Lu, Z. Fu, T. Xiang, P. Han, L. Wang, and X. Gao, "Learning from weak and noisy labels for semantic segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 3, pp. 486–500, 2016.

[29] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro, "Improving Semantic Segmentation via Video Propagation and Label Relaxation," 2018.

[30] D. Acuna, A. Kar, and S. Fidler, "Devil is in the edges: Learning semantic boundaries from noisy annotations," in *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11 075–11 083.

[31] P. Welinder, S. Branson, P. Perona, and S. J. Belongie, "The multidimensional wisdom of crowds," in *Advances in neural information processing systems*, 2010, pp. 2424–2432.

[32] Y. Cha and J. Cho, "Social-network analysis using topic models," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 565–574.

[33] Y. Aït-Sahalia, J. Fan, and D. Xiu, "High-frequency covariance estimates with noisy and asynchronous financial data," *Journal of the American Statistical Association*, vol. 105, no. 492, pp. 1504–1517, 2010.

[34] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting image databases from the web," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 4, pp. 754–766, 2010.

[35] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from internet image searches," *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1453–1466, 2010.

[36] X. Chen, A. Shrivastava, and A. Gupta, "NEIL: Extracting visual knowledge from web data," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1409–1416.

[37] S. K. Divvala, A. Farhadi, and C. Guestrin, "Learning everything about anything: Webly-supervised visual concept learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3270–3277.

[38] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache, "Learning visual features from large weakly supervised data," in *European Conference on Computer Vision*. Springer, 2016, pp. 67–84.

[39] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei, "The unreasonable effectiveness of noisy data for fine-grained recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 301–320.

[40] J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. ODonoghue, D. Visentin *et al.*, "Clinically applicable deep learning for diagnosis and referral in retinal disease," *Nature medicine*, vol. 24, no. 9, p. 1342, 2018.

[41] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros *et al.*, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *Jama*, vol. 316, no. 22, pp. 2402–2410, 2016.

[42] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian, "Disturblabel: Regularizing cnn on the loss layer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4753–4762.

[43] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1944–1952.

[44] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," in *Advances in neural information processing systems*, 2018, pp. 10 456–10 465.

[45] X. Chen and A. Gupta, "Webly supervised learning of convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1431–1439.

[46] A. J. Bekker and J. Goldberger, "Training deep neural-networks based on unreliable labels," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2682–2686.

[47] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," 2016.

[48] S. Sukhbaatar and R. Fergus, "Learning from noisy labels with deep neural networks," *arXiv preprint arXiv:1406.2080*, vol. 2, no. 3, p. 4, 2014.

[49] X. Xia, T. Liu, N. Wang, B. Han, C. Gong, G. Niu, and M. Sugiyama, "Are anchor points really indispensable in label-noise learning?" in *Advances in Neural Information Processing Systems*, 2019, pp. 6835–6846.

[50] J. Yao, H. Wu, Y. Zhang, I. W. Tsang, and J. Sun, "Safeguarded Dynamic Label Regression for Noisy Supervision," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9103–9110, jul 2019.

[51] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2691–2699.

[52] I. Misra, C. Lawrence Zitnick, M. Mitchell, and R. Girshick, "Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2930–2939.

[53] J. Lee, D. Yoo, J. Y. Huh, and H.-E. Kim, "Photometric Transformer Networks and Label Adjustment for Breast Density Prediction," may 2019.

[54] B. Yuan, J. Chen, W. Zhang, H. S. Tai, and S. McMains, "Iterative cross learning on noisy labels," in *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, vol. 2018-Janua, 2018, pp. 757–765.

[55] A. Vahdat, "Toward robustness against label noise in training deep discriminative neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 5596–5605.

[56] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie, "Learning from noisy large-scale datasets with minimal supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 839–847.

[57] M. Dehghani, A. Mehrjou, S. Gouws, J. Kamps, and B. Schölkopf, "Fidelity-weighted learning," *arXiv preprint arXiv:1711.02799*, 2017.

[58] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5552–5560.

[59] K. Yi and J. Wu, "Probabilistic end-to-end noise correction for learning with noisy labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7017–7025.

[60] X. Liu, S. Li, M. Kan, S. Shan, and X. Chen, "Self-Error-Correcting Convolutional Neural Network for Learning with Noisy Labels," 2017.

[61] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Unsupervised label noise modeling and loss correction," *arXiv preprint arXiv:1904.11238*, 2019.

[62] J. Zhang, V. S. Sheng, T. Li, and X. Wu, "Improving crowdsourced label quality using noise correction," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1675–1688, 2017.

[63] J. Han, P. Luo, and X. Wang, "Deep self-learning from noisy labels," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5138–5147.

[64] J. Yao, J. Wang, I. W. Tsang, Y. Zhang, J. Sun, C. Zhang, and R. Zhang, "Deep learning from noisy image labels with quality embedding," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1909–1922, 2018.

[65] T. Durand, N. Mehrasa, and G. Mori, "Learning a Deep ConvNet for Multi-label Classification with Partial Labels," 2019.

[66] C. G. Northcutt, T. Wu, and I. L. Chuang, "Learning with confident examples: Rank pruning for robust classification with noisy labels," in *Uncertainty in Artificial Intelligence - Proceedings of the 33rd Conference, UAI 2017*, may 2017.

[67] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, 2018.

[68] J. Huang, L. Qu, R. Jia, and B. Zhao, "O2u-net: A simple noisy label detection approach for deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3326–3334.

[69] Y. Ding, L. Wang, D. Fan, and B. Gong, "A semi-supervised two-stage approach to learning from noisy labels," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1215–1224.

[70] D. T. Nguyen, T.-P.-N. Ngo, Z. Lou, M. Klar, L. Beggel, and T. Brox, "Robust Learning Under Label Noise With Iterative Noise-Filtering," 2019.

[71] D. T. Nguyen, C. K. Mummadi, T. P. N. Ngo, T. H. P. Nguyen, L. Beggel, and T. Brox, "SELF: Learning to Filter Noisy Labels with Self-Ensembling," oct 2019.

[72] Y. Yan, Z. Xu, I. W. Tsang, G. Long, and Y. Yang, "Robust semi-supervised learning through label aggregation," in *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016, pp. 2244–2250.

[73] J. Jiang, J. Ma, Z. Wang, C. Chen, and X. Liu, "Hyperspectral image classification in the presence of noisy labels," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 851–865, 2019.

[74] J. Sahota, D. Shanmugam, J. Ramanan, S. Eghbali, and M. Brubaker, "An energy-based framework for arbitrary label noise correction," 2018.

[75] B. Han, I. W. Tsang, L. Chen, P. Y. Celina, and S.-F. Fung, "Progressive stochastic learning for noisy labels," *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–13, 2018.

[76] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," *arXiv preprint arXiv:1712.05055*, 2017.

[77] H.-S. Chang, E. Learned-Miller, and A. McCallum, "Active bias: Training more accurate neural networks by emphasizing high variance samples," in *Advances in Neural Information Processing Systems*, 2017, pp. 1002–1012.

[78] Y. Lyu and I. W. Tsang, "Curriculum Loss: Robust Learning and Generalization against Label Corruption," may 2019.

[79] S. Guo, W. Huang, H. Zhang, C. Zhuang, D. Dong, M. R. Scott, and D. Huang, "Curriculumnet: Weakly supervised learning from large-scale web images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 135–150.

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

[81] E. Malach and S. Shalev-Shwartz, "Decoupling" when to update" from" how to update"," in *Advances in Neural Information Processing Systems*, 2017, pp. 960–970.

[82] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Advances in Neural Information Processing Systems*, 2018, pp. 8527–8537.

[83] X. Yu, B. Han, J. Yao, G. Niu, I. W. Tsang, and M. Sugiyama, "How does Disagreement Help Generalization against Label Corruption?" 2019.

[84] X. Wang, S. Wang, J. Wang, H. Shi, and T. Mei, "Co-mining: Deep face recognition with noisy labels," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9358–9367.

[85] P. Chen, B. Liao, G. Chen, and S. Zhang, "Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels," may 2019.

[86] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," *arXiv preprint arXiv:1803.09050*, 2018.

[87] S. Jenni and P. Favaro, "Deep bilevel learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 618–633.

[88] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, and D. Meng, "Meta-weight-net: Learning an explicit mapping for sample weighting," in *Advances in Neural Information Processing Systems*, 2019, pp. 1917–1928.

[89] Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song, and S.-T. Xia, "Iterative learning with open-set noisy labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8688–8696.

[90] S. Thulasidasan, T. Bhattacharya, J. Bilmes, G. Chennupati, and J. Mohd-Yusof, "Combating Label Noise in Deep Learning Using Abstention," Tech. Rep., 2019.

[91] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 38, no. 3, pp. 447–461, 2015.

[92] R. Wang, T. Liu, and D. Tao, "Multiclass learning with partially corrupted labels," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2568–2580, 2018.

[93] K.-H. Lee, X. He, L. Zhang, and L. Yang, "Cleannet: Transfer learning for scalable image classifier training with label noise," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5447–5456.

[94] O. Litany and D. Freedman, "Soseleto: A unified approach to transfer learning and training with noisy labels," *arXiv preprint arXiv:1805.09622*, 2018.

[95] W. Hu, Y. Huang, F. Zhang, and R. Li, "Noise-Tolerant Paradigm for Training Face Recognition CNNs," 2019.

[96] V. C. Raykar, S. Yu, L. H. Zhao, A. Jerebko, C. Florin, G. H. Valadez, L. Bogoni, and L. Moy, "Supervised learning from multiple experts: whom to trust when everyone lies a bit," in *Proceedings of the 26th Annual international conference on machine learning*. ACM, 2009, pp. 889–896.

[97] Y. Yan, R. Rosales, G. Fung, R. Subramanian, and J. Dy, "Learning from multiple annotators with varying expertise," *Machine Learning*, vol. 95, no. 3, pp. 291–327, jun 2014.

[98] R. Tanno, A. Saeedi, S. Sankaranarayanan, D. C. Alexander, and N. Silberman, "Learning From Noisy Labels By Regularized Estimation Of Annotator Confusion," 2019.

[99] F. Rodrigues and F. C. Pereira, "Deep learning from crowds," in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018, pp. 1611–1618.

[100] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Advances in neural information processing systems*, 2009, pp. 2035–2043.

[101] S. Branson, G. Van Horn, and P. Perona, "Lean crowdsourcing: Combining humans and machines in an online system," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 6109–6118.

[102] H. Izadinia, B. C. Russell, A. Farhadi, M. D. Hoffman, and A. Hertzmann, "Deep classifiers from image tags in the wild," in *Proceedings of the 2015 Workshop on Community-Organized Multimodal Mining: Opportunities for Novel Solutions*. ACM, 2015, pp. 13–18.

[103] N. Manwani and P. Sastry, "Noise tolerance under risk minimization," *IEEE transactions on cybernetics*, vol. 43, no. 3, pp. 1146–1151, 2013.

[104] A. Ghosh, N. Manwani, and P. Sastry, "Making risk minimization tolerant to label noise," *Neurocomputing*, vol. 160, pp. 93–107, 2015.

[105] N. Charoenphakdee, J. Lee, and M. Sugiyama, "On symmetric losses for learning from corrupted labels," *arXiv preprint arXiv:1901.09314*, 2019.

[106] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006.

[107] A. Ghosh, H. Kumar, and P. Sastry, "Robust loss functions under label noise for deep neural networks," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[108] X. Wang, E. Kodirov, Y. Hua, and N. M. Robertson, "Improved Mean Absolute Error for Learning Meaningful Patterns from Abnormal Training Data," Tech. Rep., 2019.

[109] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in neural information processing systems*, 2018, pp. 8778–8788.

[110] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric Cross Entropy for Robust Learning with Noisy Labels," 2019.

[111] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Advances in neural information processing systems*, 2013, pp. 1196–1204.

[112] V. Mnih and G. E. Hinton, "Learning to label aerial images from noisy data," in *Proceedings of the 29th International conference on machine learning (ICML-12)*, 2012, pp. 567–574.

[113] Y. Xu, P. Cao, Y. Kong, and Y. Wang, "L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise," in *Advances in Neural Information Processing Systems*, 2019, pp. 6222–6233.

[114] G. Patrini, F. Nielsen, R. Nock, and M. Carioni, "Loss factorization, weakly supervised learning and label noise robustness," in *International conference on machine learning*, 2016, pp. 708–717.

[115] B. Van Rooyen, A. Menon, and R. C. Williamson, "Learning with symmetric label noise: The importance of being unhinged," in *Advances in Neural Information Processing Systems*, 2015, pp. 10–18.

[116] B. Han, I. W. Tsang, and L. Chen, "On the convergence of a family of robust losses for stochastic gradient descent," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2016, pp. 665–680.

[117] L. P. Garcia, A. C. de Carvalho, and A. C. Lorena, "Noise detection in the meta-learning level," *Neurocomputing*, vol. 176, pp. 14–25, 2016.

[118] B. Han, G. Niu, J. Yao, X. Yu, M. Xu, I. Tsang, and M. Sugiyama, "Pumpout: A meta approach for robustly training deep neural networks with noisy labels," *arXiv preprint arXiv:1809.11008*, 2018.

[119] J. Li, Y. Wong, Q. Zhao, and M. Kankanhalli, "Learning to learn from noisy labeled data," 2018.

[120] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, "Learning from noisy labels with distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1910–1918.

[121] N. Kato, T. Li, K. Nishino, and Y. Uchida, "Improving Multi-Person Pose Estimation using Label Correction," nov 2018.

[122] M. Dehghani, A. Severyn, S. Rothe, and J. Kamps, "Learning to Learn from Weak Supervision by Full Supervision," *arxiv.org*, 2017.

[123] ——, "Avoiding your teacher's mistakes: Training neural networks with controlled weak supervision," *arXiv preprint arXiv:1711.00313*, 2017.

[124] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[125] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[126] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," oct 2017.

[127] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," *arXiv preprint arXiv:1701.06548*, 2017.

[128] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[129] D. Hendrycks, K. Lee, and M. Mazeika, "Using Pre-Training Can Improve Model Robustness and Uncertainty," jan 2019.

[130] X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. M. Erfani, S.-T. Xia, S. Wijewickrema, and J. Bailey, "Dimensionality-driven learning with noisy labels," *arXiv preprint arXiv:1806.02612*, 2018.

[131] S. Azadi, J. Feng, S. Jegelka, and T. Darrell, "Auxiliary image regularization for deep cnns with noisy labels," *arXiv preprint arXiv:1511.07069*, 2015.

[132] X. Sun and H. Zhou, "An empirical comparison of two boosting algorithms on real data sets with artificial class noise," in *Communications in Computer and Information Science*, vol. 201 CCIS, no. PART 1, 2011, pp. 23–30.

[133] J. Cao, S. Kwong, and R. Wang, "A noise-detection based adaboost algorithm for mislabeled data," *Pattern Recognition*, vol. 45, no. 12, pp. 4451–4465, 2012.

[134] J. Bootkrajang and A. Kabán, "Boosting in the presence of label noise," in *Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference, UAI 2013*, sep 2013, pp. 82–91.

[135] Q. Miao, Y. Cao, G. Xia, M. Gong, J. Liu, and J. Song, "Rboost: label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 11, pp. 2216–2228, 2015.

[136] B. Sun, S. Chen, J. Wang, and H. Chen, "A robust multi-class adaboost algorithm for mislabeled noisy data," *Knowledge-Based Systems*, vol. 102, pp. 87–102, 2016.

[137] X. Yu, T. Liu, M. Gong, and D. Tao, "Learning with biased complementary labels," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 68–83.

[138] Y. Kim, J. Yim, J. Yun, and J. Kim, "NLNL: Negative Learning for Noisy Labels," aug 2019.

[139] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, 2015, pp. 1511–1519.

[140] K. Lee, S. Yun, K. Lee, H. Lee, B. Li, and J. Shin, "Robust determinantal generative classifier for noisy labels and adversarial attacks," 2018.

[141] Y. Duan and O. Wu, "Learning with Auxiliary Less-Noisy Labels," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 7, pp. 1716–1721, 2017.

[142] S. Choi, S. Hong, and S. Lim, "ChoiceNet: Robust Learning by Revealing Output Correlations," may 2018.

[143] W. Zhang, Y. Wang, and Y. Qiao, "Metacleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7373–7382.

[144] P. H. Seo, G. Kim, and B. Han, "Combinatorial inference against label noise," in *Advances in Neural Information Processing Systems*, 2019, pp. 1171–1181.

[145] L. Niu, W. Li, and D. Xu, "Visual recognition by learning from web data: A weakly supervised domain generalization approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2774–2783.

[146] B. Zhuang, L. Liu, Y. Li, C. Shen, and I. Reid, "Attend in groups: a weakly-supervised deep learning framework for learning from web data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1878–1887.

[147] B. Han, J. Yao, G. Niu, M. Zhou, I. Tsang, Y. Zhang, and M. Sugiyama, "Masking: A new perspective of noisy supervision," in *Advances in Neural Information Processing Systems*, 2018, pp. 5836–5846.

[148] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 28, no. 1, pp. 20–28, 1979.

[149] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," *arXiv preprint arXiv:1406.2080*, 2014.

[150] J.-X. Zhong, N. Li, W. Kong, S. Liu, T. H. Li, and G. Li, "Graph Convolutional Label Noise Cleaner: Train a Plug-and-play Action Classifier for Anomaly Detection," 2019.

[151] X. J. Zhu, "Semi-supervised learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2005.

[152] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[153] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Advances in Neural Information Processing Systems*, 2010, pp. 1189–1197.

[154] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: local outlier probabilities," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1649–1652.

[155] J. Vuurens, A. P. de Vries, and C. Eickhoff, "How much spam can you take? an analysis of crowdsourcing results to increase accuracy," in *Proc. ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR11)*, 2011, pp. 21–26.

[156] P. Wais, S. Lingamneni, D. Cook, J. Fennell, B. Goldenberg, D. Lubarov, D. Marin, and H. Simons, "Towards building a high-quality workforce with mechanical turk," *Proceedings of computational social science and the wisdom of crowds (NIPS)*, pp. 1–5, 2010.

[157] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical turk," in *Proceedings of the ACM SIGKDD workshop on human computation*. ACM, 2010, pp. 64–67.

[158] Y. Kong, "Dominantly truthful multi-task peer prediction with a constant number of tasks," in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2020, pp. 2398–2411.

[159] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in *Advances in Neural Information Processing Systems*, 2016, pp. 3981–3989.

[160] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1126–1135.

[161] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Advances in neural information processing systems*, 2015, pp. 3546–3554.

[162] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in neural information processing systems*, 2017, pp. 1195–1204.

[163] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[164] I. Jindal, M. Nokleby, and X. Chen, "Learning deep networks from noisy labels with dropout regularization," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 967–972.

[165] M. E. Houle, "Dimensionality, discriminability, density and distance distributions," in *2013 IEEE 13th International Conference on Data Mining Workshops*. IEEE, 2013, pp. 468–473.

[166] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine learning*, vol. 40, no. 2, pp. 139–157, 2000.

[167] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[168] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE transactions*

on pattern analysis and machine intelligence, vol. 40, no. 6, pp. 1452–1464, 2017.

[169] T. Kaneko, Y. Ushiku, and T. Harada, “Label-Noise Robust Generative Adversarial Networks,” 2018.

[170] K. K. Thekumparampil, A. Khetan, Z. Lin, and S. Oh, “Robustness of conditional GANs to noisy labels,” in *Advances in Neural Information Processing Systems*, vol. 2018-Decem, 2018, pp. 10 271–10 282.

[171] E. Fonseca, M. Plakal, D. P. Ellis, F. Font, X. Favory, and X. Serra, “Learning Sound Event Classifiers from Web Audio with Noisy Labels,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2019-May, 2019, pp. 21–25.

[172] Y. LeCun, “The mnist database of handwritten digits,” *http://yann.lecun. com/exdb/mnist/*, 1998.

[173] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv preprint arXiv:1708.07747*, 2017.

[174] A. Torralba, R. Fergus, and W. T. Freeman, “80 million tiny images: A large data set for nonparametric object and scene recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.

[175] A. Krizhevsky, G. Hinton *et al.*, “Learning multiple layers of features from tiny images,” Citeseer, Tech. Rep., 2009.

[176] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng, “Reading digits in natural images with unsupervised feature learning,” *NIPS*, 01 2011.

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

[178] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, “The new data and new challenges in multimedia research,” *arXiv preprint arXiv:1503.01817*, vol. 1, no. 8, 2015.