# 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 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 literature. Deep networks are known to be relatively robust to label noise, however 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 for training deep neural networks efficiently. Even though an extensive survey of machine learning techniques under label noise exists, literature lacks a comprehensive survey of methodologies specifically centered around deep learning in the presence of noisy labels. This paper aims to present these algorithms while categorizing them according to their similarity in proposed methodology.**

***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 are able to memorize complete random noise [11]. Various works are devoted to better explain this phenomenon [12], [13], yet regularizing deep neural networks (DNNs), 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 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 significant decrease in performance [15], [16], label noise is considered to be more harmful [14] and shown to deteriorate the performance of classification systems in a broad range of problems [14], [17]–[19]. This is due to several factors; label is unique for each data while features are multiple and importance of each feature varies while label has always big impact [15], [19]. This work focuses on label noise, therefore noise and label noise is used synonymous throughout the article.

Necessity of 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 are emerged. For example, almost unlimited amount of data can be collected from web with the help of search engines or from social media. Similarly, 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. Beside these methods, label noise can occur even in the case of expert annotators. Labelers may lack 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 adversarially for data poisoning purposes [20], [21]. Being a natural outcome of dataset collection and labeling, makes label noise robust algorithms an important topic for development of efficient computer vision systems.

Supervised learning with label noise is an old phenomenon with three decades of history [22]. 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 specifically focuses to fill 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 learning over its alternatives, it is a topic of interest and many works are conducted in literature. Throughout the paper these methods will be briefly explained and grouped to provide reader a clear overview of the literature.

This paper is organized as follows. Section II explains several concepts that will be used throughout the paper. Proposed

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

solutions in literature are categorized into two major groups and these methods are discussed in section III - section IV. Section V discusses commonly used datasets and how to add synthetic label noise. Finally, section VI concludes the paper.

## II. Preliminaries

In this section, first the problem statement for supervised learning with noisy labels is given. Secondly, types and sources of label noise are be discussed.

### A. Problem Statement

Classical supervised learning consists of a input dataset $D = \{(x_1, y_1), ..., (x_n, y_n)\}$ where $x_i$ denotes the $i^{th}$ sample and $y_i$ is the corresponding true class which are coming from set $\{X, Y\}$. Task is to find the mapping function of neural network parameterized by $\theta$ so that $f(x_i, \theta) = y_i$ for $\forall i$. When there exists noisy labels in dataset it turns into $\tilde{D} = \{(x_1, \tilde{y_1}), ..., (x_n, \tilde{y_n})\}$, where $\tilde{y}$ represents the observable noisy label. Task remains to find out the function that can predict the true classes even without observing them. Throughout this paper $\{D, y\}$ will be used to represent the noise-free dataset and label respectively and $\{\tilde{D}, \tilde{y}\}$ for their noisy counterparts. Final dataset is represented with $D_f$, which can be one of two: datasets with small amount of reference set $D_f = \{D_{ref}, \tilde{D}\}$ or datasets without reference set $D_f = \{\tilde{D}\}$. In case of existence of reference set, which is noise-free subset of dataset, performance of algorithms can be tested on this subset or it can be further used to utilize learning. If such a subset is not available, possible methodologies to obtain test set is discussed in detail in subsection V-A

Label vector $y$ can either be binary variable for binary classification task or one-hot vector for multi-class classification problem. In case of multi-labeled data, there can be more than one $\tilde{y_i}$ for instance $x_i$, while there is only one $y_i$. If instances contain multiple objects, $y_i$ can represent more than one class. In this setup annotator may skip to label some classes in an instance which would result in *partial labels*, where corresponding labels for a particular instance are not changed but omitted.

While most of the works assumes $y, \tilde{y} \in Y$, it is not always necessarily the case that $y$ is from set $Y$. That means instances associated with noisy labels may have true class which is not contained in $Y$, which is named as *open-set noisy labels* problem [23].

### B. Label Noise Models

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

*Random noise* is totally random and does not depend on neither 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 $X$ but depends on $Y$. That means data from a particular class is more likely to be mislabeled. For example, in hand written digit recognition task, ”3” and ”8” much more likely to be confused with each other rather then ”3” and ”5”. *XY-dependent noise* depends on both $X$ and $Y$. As in the y-dependent case, objects from particular class may be more likely to be mislabeled. Moreover, 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.

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 characteristic of each labeler and using this information during training, significantly boosts the performance [24]. Characteristic of labeler can be independent from both $X$ and $Y$. For example in a crowd-sourced dataset, some labelers can be total spammers who label with a random selection [25]. The behavior of each labeler can be explained by one of the groups above. For example, a spammer can be categorized as random noise while a better labeler can be modeled by *y-dependent* or *xy-dependent noise*. Therefore characteristic of 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 be seen 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 the solutions to such problems, but it may be helpful to investigate the causes of label noise to better understand the phenomenon.

Firstly, with the availability of gigantic amount of data on 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 dataset are well known to result in noisy labels.

Secondly, dataset can be labeled by multiple experts resulting in multi-labeled dataset. Each labeler has 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 others predictions for difficult 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 differentiating rate of error. Some labelers even can be a total spammer whose labels are random noise [25].

Thirdly, data can be too complex 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 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 can be of great interest.

Lastly, adversarial label noise can be inserted for the purpose of data poisoning to harm the learning procedure [20], [21].

### D. Methodologies

There are many possible ways to group methods proposed to increase the classification performance in the presence of noisy labels. For example, one possible way to distinguish algorithms is according to their need of reference set, that is cleanly annotated subset of dataset or not. Alternatively, they can be divided according to noise type they are dealing with or dataset is being multi-labeled or not. However, these are not handy to understand the main approaches behind the proposed algorithms, therefore a different sectioning is proposed as: noise model based and noise model free methods.

Noise model based methods aim to model the noise and uses this information to provide noise robust learning. Learned noise can be used to provide clean training to base classifier by either bridging between model and noisy data or by manipulating input stream to the model.

Noise model free algorithms don't model the noise explicitly but tries to overcome its negative effects with more generic methods.

Both of the mentioned approaches are discussed and further categorized in section III and section IV. 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

One way to overcome noisy labels is to explicitly or implicitly model the noise structure. Easiest option is to take noise as random and symmetric, but this is not usually the case. Noise is mostly dependent on various factors (e.g. *y-dependent*, *xy-dependent*) and is asymmetric. Therefore taking noise behavior into account can improve the performance significantly [15]. Methods in this section aim to model the noise and use this information during training for better performance. Following sections describe different approaches under this category.

### A. Noisy Channel

General setup for noisy channel is illustrated in Figure 1. By modeling noise transition matrix, true class probabilities of data samples can be extracted and fed into the classifier for noise robust training. Noisy channel works as mapping between model predictions and noisy labels. In test time, noisy channel can easily be removed to obtain noise free predictions of base classifier. In these kind of approaches, a common problem is scalability, since increasing number of classes requires increasing number of parameters to be estimated in noise transition matrix. This can be avoided by allowing connections only among most probable nodes [49] or predefined nodes [46]. There exists three main approaches for utilizing noisy channel in learning, as discussed below.

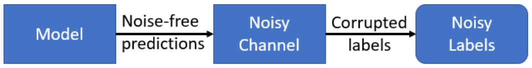

Fig. 1: Noise can be modeled as a noisy channel on top of base classifier.

*1) Extra layer:* Noisy channel can be modeled as an extra layer on top of the classifier. In the backpropagation phase, noise free gradients are passed through this layer for clean training. In test phase, this layer is removed to get noise-free predictions. [42][43] add linear fully connected layer as a last layer of the base classifier. To make this layer adapt the behavior of noise, trace of the noise transition matrix is added to the loss function as a regularizer. In [44] 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 [118].

*2) Separate Network:* In this approach, noisy channel is modeled with another network. Tree types of label noise is defined in [45] namely: no noise, random noise, structured noise. Two convolutional neural networks (CNNs) are used to predict the true label and noise type of instances respectively. Then expectation-maximization algorithm (EM) [136] is used to iteratively calculate posterior of random variables and network parameters. However, they need a small set of clean data to pre-train networks to give a good starting point for EM to converge. [46] uses masking which allows certain class transitions to happen while others not. With this extra constraint, one only needs to estimate unmasked noise transitions. Prior on the noise structure can be obtained from human experts which allows additional label information to be inserted to training procedure. Afterwards both the classifier and the noise transition matrix are approximated with neural networks. [47] 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, additional network tries to extract the true class of each instance.

*3) Explicit Calculation:* Noise transition matrix can be calculated explicitly and applied to map model predictions to noisy labels. In [48], [49] EM is used to iteratively train network to match given distribution and estimate noise transition matrix given the estimate of true labels. EM is known to depend on initial point, so it is suggested to first train network on noisy data to provide an initial point or train it with a predefined noise level. Same approach is used on medical data with noisy labels in [26]. [51] decouples these two phases by first estimating noise transition matrix and then learning with loss correction. Two types of loss corrections are proposed: forward and backward. Forward loss multiplies model predictions with noise transition matrix to match them with noisy labels while backward loss multiplies

| Noise Model Based Methods |
|---|
| 1.***Noisy Channel***<br>a.*Extra layer*: Linear fully connected layer [42][43], softmax layer [44]<br>b.*Separate Network*: Estimating noise type [45], masking [46], quality embedding [47]<br>c.*Explicit Calculation*: EM [26], [48], [49], conditional independence [50], forward&backward loss [51], unsupervised generative model [52], bayesian form [53] |
| 2.***Label Noise Cleansing***<br>a.*Using Reference Set*: train cleaner on reference set [54], [55], clean based on extracted features [56], teacher cleans for student [57], ensemble of networks [58]<br>b.*Not Using Reference Set*: Moving average of network predictions [59], consistency loss [60], ensemble of network [61], prototypes [62], random split [63], confidence policy [64] |
| 3.***Sample Choosing***<br>a.*Self Consistency*: Consistency with model [65], consistency with moving average of model [66], graph-based [67], dividing to two subset [68]<br>b.*Curriculum Learning*: Screening loss [69], teacher-student [70], selecting uncertain samples [71], extra layer with similarity matrix [72], curriculum loss [73], data complexity [74], partial labels [75]<br>c.*Multiple Classifiers*: Consistency of networks [76], co-teaching [77]–[79]<br>d.*Active Learning*: Relabel hard samples for classifier [39] |
| 4.***Sample Importance Weighting***<br>Estimate noise rate [**?**], [80], [81], visual relevance [82], similarity loss [83], feature relevance [84], meta task [85], siamese network [23], pLOF [27], abstention [86], $\theta$-distribution [87], estimating target distribution [88] |
| 5.***Labeler Quality Assessment***<br>EM [25], [89], [90], trace regularizer [91], image difficulty estimate [92], consistency with network prediction [93], omitting probability variable [94] , softmax layer per labeler [24], crowd-layer [95] |

| Noise Model Free Methods |
|---|
| 1.***Robust Losses***<br>Risk minimization [96], [97], 0-1 loss surrogate [98], MAE [99], IMEA [100], Generalized cross-entropy [101], symmetric cross entropy [102], linear-odd losses [103], classification calibrated losses [104], unbiased estimator [105], modified cross-entropy for omission [106] |
| 2.***Meta Learning***<br>Choosing best methods [107], pumpout [108], noise tolerant parameter initialization [109], knowledge distillation [110], [111], gradient magnitude adjustment [112], [113] |
| 3.***Multiple-Instance Learning***<br>Bag loss [114], attention model [115] |
| 4.***Semi-Supervised Learning***<br>Remove noisy labels before training [116], remove noisy labels iteratively with moving average [117] |
| 5.***Regularizers***<br>Dropout [118], adversarial training [119], mixup [120], label smoothing [121], [122], pre-training [123], weighting mini-batches [124], checking dimensionality [125] |
| 6.***Ensemble Methods***<br>LogitBoost&BrownBoost [126], noise detection based AdaBoost [127], rBoost [128], RBoost1&RBoost2 [129], robust multi-class AdaBoost [130] |
| 7.***Others***<br>Minimum covariance determinant [131], complementary labels [132], [133], dataset with noisy and less noisy labels [134], autoencoder reconstruction error [135] |

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

the calculated loss with inverse of noise transition matrix to obtain unbiased estimator of loss function for clean data. To calculate noise transition matrix, model is first trained on noisy data and then just based on noisy class probability estimates of the model, noise transition matrix is estimated. Conditional independence between $\tilde{y}$ and $y$ is assumed in [50]. Network is first trained on noisy data and then makes predictions on the *trusted data*. Softmax probabilities are summed over each predictions of trusted data to construct noise transition matrix. In [52] arguing that, in case of noisy labels, model first learns correctly labeled data and then overfits to noisy data, probability of sample being noisy or not is estimated by unsupervised generative model of instance loss values. In order to achieve this, cross entropy loss of each sample is fitted by beta mixture model, which in turn models the label noise in an unsupervised manner. [53] models noise transition matrix in Bayesian form by projecting it into a Dirichlet-distributed space. During training, posteriors for true labels are also iteratively calculated to boost performance.

### B. Label Noise Cleansing

A simple methodology to deal with noisy labels is to remove samples with suspicious labels or correct their noisy label to corresponding true class. This can be done in preprocessing stage of the training data, however such methods usually tackle the difficulty of distinguishing informative hard samples from those with noisy labels [15]. With superior classification abilities of deep neural networks, different methodologies to train label cleaning networks are proposed in literature. Unlike preprocessing of data, label cleaning models evaluated in this section are trained mutually with base classifier and their abilities keep improving during training. Methods under this category can be subdivided into two according to their need of reference set, label noise free subset of data, or not.

*1) Using Reference Set:* If there exists a clean subset of data, which has both noisy label and verified true label, network trained on this dataset can be used to relabel noisy labels until convergence [54]. In [55], network trained on a clean dataset is fine tuned on noisy dataset for relabeling. Afterwards, network is again trained on relabeled data to re-sample dataset. In [56] label cleaning network gets two inputs: extracted features of instances by the base classifier and corresponding noisy label. It is trained to predict true class depending on these inputs and therefore learn the noise structure. For noisy data, base classifier is supervised by the predictions of label cleaning network. [57] uses teacher to clean labels for student to fine tune its training. First, student is trained on noisy data. Then features are extracted from clean data via the student model and teacher learns the structure of noise depending on these extracted features. Afterwards, teacher predicts soft labels for noisy data and student is again trained on these soft labels for fine tuning. Ensemble of networks which are trained with different subset of dataset are used in [58]. If they all agree on the label it is changed to predicted label, otherwise label is set to a random label. Authors argue that setting it to random would make noise more uniformly random and less structured which results in less bias in the learned network.

*2) Not Using Reference Set:* Joint optimization framework for both training classifier and propagating noisy labels to cleaner labels is presented in [59]. Their work consists of two progressive steps 1) Fix labels and update classifier weights with stochastic gradient descent (SGD), 2) Fix network parameters and update labels as moving average of network predictions. Additional regularization terms are added to cost label cleaning network for preventing it attaining one class posterior to all classes and base network predicting it. The same framework is used in [60] with probabilistic end-to-end fashion, so that label posteriors are approximated with the help of classification and compatibility loss instead of moving average of network predictions. [61] proposes a two level approach. In the first stage, with any chosen inference algorithm, ground truth of the labels are determined and data is divided into two subset as noisy and clean. In the second stage, 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. [62] constructs prototypes, that are able to represent deep feature distribution of the corresponding class, for each class. Then corrected label is found by checking similarity among the data sample and prototypes. Using corrected label and noisy label, the final classifier network is trained. Assuming that correctly labeled data account for majority, [63] 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. [64] deploys a confidence policy where labels are determined by either network output or given noisy labels. With increasing number of epochs, more confidence is given to the network output defending that it gets better at predicting labels during training. [137] 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. Sample Choosing*

A widely used approach to overcome label noise is to manipulate the input stream to the classifier. Guiding the network with choosing right instances to be trained on, can help classifier to find its way easier in the presence of noisy labels. Since these methods operate outside of the existing system, they are easier to attach to the existing system as an add-on. Four major approaches under this group are discussed in the following paragraphs.

*1) Self Consistency:* Next samples to be trained on can be chosen by checking the consistency of the label with the network prediction. In [65], if both label and model prediction of the given sample are consistent, it is used in the training set. Otherwise, model has a right to disagree. Iteratively this provides better training data and better model. However, there is a risk of model being too skeptical and choosing labels in a delusional way, therefore consistency balance should be established. [66] uses consistency among label prediction in the current epoch 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. In [67] graph-based approach is used, where relation among noisy labels and clean labels are extracted by a conditional random field (CRF). In [68], 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 and less confident examples are removed from the clean subset. Finally, classifier is trained on pruned dataset.

*2) Curriculum Learning:* Curriculum learning (CL) [138], 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* [139], when prior of 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 easy task while noisily labeled data is 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 [69] to sort instances according to their noisiness level. Teacher-student approach is implemented in [70], where the task of the teacher is to choose most probably to be clean samples for the student. Novelty in their approach is the non-predefined curriculum, so that teacher gets the outputs from student as an input to update its curriculum during training. Arguing that CL slows down the learning speed, since it focuses on easy samples, [71] suggests to choose *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. Two datasets is gathered in [72] namely: easy data and hard data. Classifier is first trained on easy data to extract similarity relationships among classes, then calculated similarity matrix is added as an extra layer on top of the model and hard samples are fed for fine tuning. 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 [73]. In [74], data is split into subgroups according to their complexities, which are extracted by network that is pre-trained on 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 network gets better. In case of partial labels, [75] uses one network to find and estimate easy missing samples and other network to be trained on this corrected data. As network gets better, it can find harder labels, therefore providing a curriculum learning strategy.

*3) Multiple Classifiers:* Some works use two classifiers to help each other to choose next batch of data to train on. This is different than teacher student approach, since none of the networks is supervising the other on but they rather help each other out. This can provide robustness since networks can correct each other's noisy behavior due to their difference in feature domain. For this setup to work, initialization of the classifiers are important. They are most likely to be initialized with 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 [76] 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. Paradigm of *co-teaching* is introduced in [77], where two networks select next batch of data for each other. 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 filters noise error more successfully. Idea of co-teaching is further improved by iterating over data where two networks disagree [78], to prevent two network converging each other with increasing number of epochs. 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 [79].

*4) Active Learning:* Getting whole dataset cleanly annotated can be expensive, so active learning methods can be used to label only the critical instances. [39] suggests to train network on a large noisy dataset, which is easily obtainable from the web. After the classifier is trained, instances that has too much inconsistency with the model predictions are sent to annotator for correction. This procedure is repeated until desired classification accuracy is obtained.

### *D. Sample Importance Weighting*

Similar to subsection III-C, 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. Simplest approach would be, in case of availability of both clean and noisy data, weighting clean data more [43]. However, better performance can be achieved with more complex algorithms and clean data is not always reachable.

Weighting factor for instances proposed in [80], depends on conditional distribution of noisy samples $P_D(\tilde{Y}|X)$ and noise rate. While $P_D(\tilde{Y}|X)$ can directly be estimated by trained classifier on data, estimating noise rate is challenging. Assuming that there exists clean samples in dataset, noise rate is upper-bounded by $P_D(\tilde{Y}|X)$ and it is estimated by minimizing $P_D(\tilde{Y}|X)$ over $X \in X_1, ..., X_n$. Same methodology is extended to multi-class case in [81] with additional ways to estimate noise rate. [82] trains two separate networks to tackle with *labeler bias*: visual presence and relevance classifiers. While visual presence classifier tries to learn if objects exist in the data sample, relevance classifier attains the relevance of the class for the given instance. If relevance is low, in case of noise, classifier can still make predictions of true class and will not be penalized much for it. In [83], feature extractor network is used to extract features from both reference set and queried sample. Similarity loss is defined to measure relevance of the image to its noisy label, which is then used to weight importance of the particular sample to learning. [84] 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 mode is forced to choose relative features and therefore weights more the reliable images. In [85], meta learning paradigm is used 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 noise-free data. Weighting factor is validated in each iteration, which is argued to have regularizer effect on the model. *Open-set noisy labels* are considered in [23]. 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. In each iteration weighting factor is recalculated and classifier is updated. [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 is weighted according to their noisiness level which is estimated by pLOF [140]. [86] introduces *abstention*, which gives option to abstain samples, depending on their cross-entropy error, with an abstention penalty. Therefore network learns to abstain confusing samples during learning which helps to learn underlying structure of noise. [124] proposes to weight mini-batches according to similarity between gradients of training batch and validation batch, in order to improve generalization and prevent negative effects due to mislabeled instances. [87] uses $\theta$ values of samples in $\theta$-distribution to calculate their probability of being clean and use this information to weight clean samples more in training. [88] estimates the quality of data samples by checking consistency between generated and target distributions. This information is then used for loss correction for noisy labeled data.

### *E. Labeler Quality Assessment*

As explained in subsection II-C, there can be several reasons for multi-labeled datasets to exist. Each labeler may have different level of expertise and their labels may commonly contradict with each other. This is a common case in crowd-sourced data [141]–[143] or datasets which requires high level of expertise such as medical imaging [18]. Therefore, modeling and using labeler characteristic can significantly increase performance [24].

In this setup there are two unknowns namely; labeler noise transition matrix and ground truth labels. One can estimate both with EM algorithm [25], [89], [90]. [91] adds a regularizer to loss function which is the sum of traces of annotator confusion matrices, arguing that this additional term helps approximating true confusion matrices under some mild conditions. In [92], also difficulty of images are estimated for better evaluation of labeler accuracy. Human annotators and computer vision system are used mutually in [93], where consistency among predictions of these two components are used to evaluate the reliability of labelers. [94] deals with the noise when labeler omits a tag in the image. Therefore, instead of confusion matrix of labelers, omitting probability variable is used, which is estimated together with true class using EM algorithm. Softmax layer is trained for each annotator in [24] and a final classifier 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. Similar approach is implemented in [95], where crowd-layer is added to the end of network. Crowd-layer learns annotator specific mapping from output layer to noisy labels and adjusts gradients for each labeler in backpropagation.

## IV. Noise Model Free Methods

These methods aims to achieve label noise robustness without explicitly modeling it, but rather designing robustness in the proposed algorithm. 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 [96]. 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].

In [96], 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 [97] derives sufficient conditions for a loss function to be noise tolerant under the risk minimization for uniform and non-uniform noises. Their work shows that if a loss function is *symmetric*, meaning that sum of its components are equal to a constant and noise level is below a threshold, under uniform noise it is noise robust. In this content, they empirically show that none of the standard convex loss functions has noise robustness while 0-1 loss has, under given circumstances. However, 0-1 loss is non-convex and non-differentiable, therefore surrogate loss of 0-1 loss is proposed in [98], which is still noise sensitive. Multi-class classification is considered in [99] and it is shown that mean absolute value of error (MAE) is inherently robust to label noise under same circumstances. IMAE, which is an improved version of MAE, is proposed in [100], where gradients are scaled with an hyper-parameter to adjusts MAE's weighting variance. [101] argues that MAE provides much smaller learning rate than categorical cross entropy (CCE), therefore a new loss function is suggested which combines robustness of MAE and implicit weighting of CCE. With a tuning parameter, it can be made more similar to MAE or CCE. Symmetric cross entropy is used in [102], arguing that symmetry provides noise robust boosting in learning hard samples. Given that mean operator is not significantly affected by the noise, work of [103] showed that class of linear odd losses (LOLs) are robust to the label noise at some degree, such that difference between the risks of minimizers with noisy data and clean data are smaller than a predefined threshold. [104] shows that classification-calibrated loss functions are asymptotically robust to symmetric label noise. SGD method in general is analyzed under label noise in [144] and shown to be more robust than its counterparts.

Given that noise prior is known [105] provides two surrogate loss functions namely unbiased and weighted estimator of the loss function using the prior information about label noise. [106] considers asymmetric omission noise for binary classification case, where the task is to find road pixels from a satellite map image. Omission noise makes 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.

### B. *Meta Learning*

With the recent advancements of deep neural networks, necessity of hand designed features for computer vision are mostly eliminated. Instead, these features are learned autonomously via machine learning techniques. Even though these systems are able to learn complex functions on their own, there still remains many hand designed parameters such as network architecture, hyper-parameters, optimization 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 [145], [146].

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 most suitable method, among family of methods, for a given noisy dataset in [107]. *Pumpout* [108] presents a meta objective as recovering the damage done by noisy samples by erasing their effect on model via *scaled gradient ascent*. Model-agnostic meta-learning (MAML) [146] seeks for weight initialization which can easily be fine-tuned. Similar approach is used in [109] for noisy labels, which aims to find noise-tolerant model parameters that are less prone to noise under teacher-student training framework [147], [148]. Training consists of two steps: 1) in meta-train step synthetic noisy data is fed to students to update their parameters 2) in meta-test step consistency loss between teacher and all students are applied. Finally parameters are updated such that consistency loss is minimized and teacher network parameters are set as the exponential moving average of the student networks.

In case of available clean data, a meta objective can be defined to utilize these information. Approach used in [110] is to train a teacher network in a clean dataset and transfer its knowledge to student network for the purpose of guiding train process in the presence of mislabeled data. They used *distillation* technique proposed in [149] for controlled transfer of knowledge from teacher to student. Similar methodology of using *distillation* together with label correction in human pose estimation task is implemented in [111]. In [112], [113] target network is trained on excessive noisy data and confidence network is trained on reference set. Inspiring from [145], 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. Multiple-Instance Learning

In multiple-instance learning (MIL), data is grouped in clusters, called as bags, and each bag is labeled as positive if there is at least one positive instance in it and negative otherwise. Network is fed with group of data and produces a single prediction for each bag by learning inner discriminative representation of data. Since group of images are used and one prediction is produced, existence of noisy labels along with true labels in a bag has less impact on learning. In [114] authors propose to effectively choose training samples from each bag by minimizing the total bag level loss to train noise robust classifiers. Extra model is trained in [115] 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.

### D. Semi-Supervised Learning

Since noise is only in labels and not in features, only labels of noisy instances can be removed and network can be trained in semi-supervised fashion [150]. In [116], first most probably noisy labels are detected and removed, then the network is trained with labeled and unlabeled data in semi-supervised fashion. Label noise filtering is done iteratively in [117] as the moving average of model prediction consistency with given labels. [151] trains weak classifiers on labeled data and predicts labels of unlabeled data. These pseudo labels are used to infer the true labels. Empirically authors shows the robustness of the proposed system to label noise, compared to other semi-supervised learning techniques.

### E. Regularizers

Regularizers are well known to prevent DNNs from overfitting noisy labels. Some widely used techniques are weight decay, dropout [118], adversarial training [119], mixup [120], label smoothing [121], [122]. In [123], it is shown that pre-training contributes to label noise robustness of model. [125] proposes a complexity measure to understand if 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. Key is to stop before second step. *Local instrinsic dimensionality* (LID) [152] is used to measure complexity of trained model and stop before it starts to overfit.

### F. Ensemble Methods

It is well known that bagging is more robust to label noise than boosting [153]. 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 is more robust than AdaBoost [126]. Therefore noise robust alternatives of AdaBoost is proposed in literature, such as noise detection based AdaBoost [127], rBoost [128], RBoost1&RBoost2 [129] and robust multi-class AdaBoost [130].

### G. Others

Some works [132], [133] use *complementary labels*, which specify classes that observations do not belong to, to tackle noisy label since annotator is less likely to mislabel. In [134] dataset has two kinds of label; noisy label and less-noisy label. They provide EM based training framework to iteratively infer ground truth and train classifier. [135] uses reconstruction error of autoencoder to discriminate noisy data from clean data, arguing that noisy data tend to have bigger reconstruction error. In [131], first base model is trained with noisy data. Additional generative classifier is trained on top of feature space generated by the base model. By estimating its parameters with *minimum covariance determinant* (MCD), noise robust decision boundaries are aimed to be found.

## V. Datasets and Noise Generation

This section describes commonly used experimental setups for the evaluation of algorithms in the presence of label noise. Firstly, widely used datasets and methods to generate test sets are discussed. Afterwards, methodologies for adding synthetic noise are be presented.

### A. Datasets and Test Set

Benchmark datasets used to test performance of algorithms in the presence of label noise can be divided into two subcategories: datasets with noisy labels and datasets with clean labels. Datasets in the first category, commonly have reference set of cleanly annotated samples in order to be used as a validation set. In case of datasets belonging to second category, synthetic label noise is added manually to the training set while keeping validation set clean in order to evaluate performance of the network. Methods for adding the synthetic label noise are discussed in subsection V-B.

Datasets, that are commonly used in the literature of deep learning with label noise, are listed in Table II. Beside these presented datasets, if the given dataset is noisy and there is no reference set, there are several methodologies to split dataset into noisy training set and clean validation set. One option is to get small portion of data labeled by reliable experts with an additional cost. Another option, in case of multi-labeled data, is to extract data on which all experts agreed on as validation set. Differently, one can first train a network with noisy data and the data instances which predicted with big confidence can be accepted as clean.

| Name | Image count | Class count |
|---|---|---|
| ***Datasets with Clean Annotation*** | | |
| MNIST [154] | 70K | 10 |
| Fashion-MNIST [155] | 70K | 10 |
| Cifar10 [156] | 60K | 10 |
| Cifar100 [157] | 60K | 100 |
| SVHN [158] | 630K | 10 |
| ImageNet [159] | 1.2M | 1000 |
| MS-COCO [160] | 320K | 80 |
| OpenImages [161] | 9M | 6012 |
| ***Datasets with Noisy Annotation*** | | |
| Clothing1M [9] | 1M | 14 |
| Food101N [83] | 310K | 101 |
| WebVision [162] | 2.4M | 1000 |
| YFCC100M [163] | 100M | 5400 |

TABLE II: Publicly available datasets, which are commonly used in the literature for studying label noise.

### *B. Adding Synthetic Label Noise*

There are several widely used methods to add synthetic label noise to the given dataset. Y-dependent and XY-dependent noise is considered in this section, since random noise can easily be added by changing labels randomly. All mentioned methods are for the case of single labeler. If one requires to create synthetic noise for multiple annotators, any of the given methods can be used multiple times with different parameters to mimic different labelers.

*1) **Y-Dependent Noise**:* Y-dependent noise can be represented with confusion-matrix, where the entry $p_{ij}$ represents the probability of flipping label of instance from class i to class j. Then samples from each class can be flipped to other classes with the given probabilities. Since confusion matrix consists of probabilities, it should be satisfied that $p_{ij} > 0$ for $\forall i, j$ and $\sum_{j=0}^{j=n} p_{ij}$ for $\forall i$ where n is the number of classes.

***Uniform Noise:*** Confusion matrix is defined where probability of choosing any other class, rather than true class, is equally distributed. A predefined probability of mislabeling $p$ can be defined and samples from each class is changed from its true class to any other classes with equal probability.

$$T_U = \begin{bmatrix} 1-p & \frac{p}{n-1} & \cdots & \cdots & \frac{p}{n-1} \\ \frac{p}{n-1} & 1-p & \frac{p}{n-1} & \cdots & \frac{p}{n-1} \\ \vdots & & \ddots & & \vdots \\ \frac{p}{n-1} & \cdots & \cdots & \frac{p}{n-1} & 1-p \end{bmatrix}$$

***Random Noise:*** Confusion matrix is defined where probability of choosing any other class, rather than true class, is randomly distributed. A stochastic probability of mislabeling $p_{ij}$ can be defined for each mislabeling probability from class i to j.

$$T_R = \begin{bmatrix} 1-p_0 & p_{01} & \cdots & \cdots & p_{0n} \\ p_{10} & 1-p_1 & p_{12} & \cdots & p2n \\ \vdots & & \ddots & & \vdots \\ p_{n0} & \cdots & \cdots & p_{nn-1} & 1-p_n \end{bmatrix}$$

Where $p_i = p_{i0} + ... + p_{in}$

***Paired Noise:*** Any instance from a class can only be flipped to one other class with probability $p$. In this type of noise, sufficient condition for learning is $p < \%50$, because otherwise pairwise mislabeled data takes the majority.

$$T_P = \begin{bmatrix} 1-p & p & 0 & \cdots & 0 \\ 0 & 1-p & p & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ p & 0 & \cdots & \cdots & 1-p \end{bmatrix}$$

***Class Dependent Noise:*** Flip probabilities from class i to j is assigned according to similarity of classes. For example, in handwritten digit recognition task $p_{17}$, which is probability of mislabeling '1' as '7', can be higher while $p_{15}$ is lower. These probabilities can be assigned by an human annotator or by the prediction probabilities of pre-trained network.

$$T_C = \begin{bmatrix} 1-(p_{12}+p_{13}) & p_{12} & p_{13} \\ p_{21} & 1-(p_{21}+p_{23}) & p_{23} \\ p_{31} & p_{32} & 1-(p_{31}+p_{32}) \end{bmatrix}$$

*2) **XY-Dependent Noise**:* Artificially creating this type of noise is more complicated than previous ones, since one needs to consider the feature vectors of each sample. Most obvious method is to use human annotator to choose samples, but this is both expensive and not scalable. One particular approach is to use pre-trained network, which is either directly training on given data [10], [42] or dataset from similar but different domain [59], to extract decision boundaries . Differently, in case of the presence of surrounding texts for image, user defined texts can be used to extract mostly probable noisy label [25].

## VI. 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 dataset from web [164], [165], design networks that can learn from unlimited web data with no human supervision [35]–[38]. Furthermore, beside image classification classification, there are more fields where dealing with mislabeled instances is important, such as generative networks [166], [167], semantic segmentation [28]–[30], sound classification [168] and more. All these factors make dealing with label noise an important step through self sustained learning systems.

Methods in literature can be subdivided into two major groups: noise model based and noise model free methods. Methods in the first group aims to model the noise somehow and use this information to decrease the negative effect of noisy labels on learning. On the other hand, methods in the second group purposes to design more generic algorithms that would fade away the impact of noise. Commonly used techniques in past for machine learning, such as data cleansing in preprocessing stage seems to lose its charm in the era of deep learning due to its inability to differentiate hard informative samples from noisy samples. Instead, algorithms that iteratively clean data while training are emerged, which can be included in the family of noise model based algorithms. Also instead of cleaning label noise, breaking its structure is shown to be an effective method as well [63]. Practitioner may choose appropriate algorithm depending on constraints, such as label noise type, having reference set or not, size of the dataset, computation power required and more.

It is seen that adding synthetic label noise is an important stage for testing algorithms with existing benchmark datasets without noisy labels. Mostly used method is to add random noise or y-dependent noise with predefined noise rates, however these approaches may not be realistic most of the time. Therefore, more generic framework to dilute given dataset can be a topic of interest. Moreover, works indicate that mislabeled instances may have an effect of regularizer which improves generalization and prevents overfitting [74], [169]. Therefore polluting labels in a controlled way to improve robustness can be an interesting topic to be further researched.

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 [19], 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 network is highly effected from label noise may help to achieve transfer learning effectively. Alternatively, question of how to train in the existence of both attribute and label noise is an understudied problem with big potential on practical applications [54]. [102] 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 future.

Very small attention is given to the learning from noisy labeled dataset when there is a small amount of data. This can be a fruitful research direction considering its potential on fields where harvesting dataset is costly. For example, in medical imaging, collecting a cleanly annotated large dataset is not feasible most of the time [54], due to its cost or privacy of the data. Effectively learning from small amount of noisy data with no ground truth can have significant effect on autonomous medical diagnosis systems. Even though some pioneer researches are available [24], [26], [27], there are 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] 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.

[18] 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.

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

[20] 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.

[21] 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.

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

[23] 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.

[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," 2019.

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

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

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

[50] 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.

[51] 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.

[52] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Unsupervised Label Noise Modeling and Loss Correction," 2019.

[53] 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.

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

[55] 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.

[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] 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.

[59] 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.

[60] K. Yi and J. Wu, "Probabilistic End-to-end Noise Correction for Learning with Noisy Labels," mar 2019.

[61] 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.

[62] J. Han, P. Luo, and X. Wang, "Deep Self-Learning From Noisy Labels," aug 2019.

[63] 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.

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

[65] 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.

[66] 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.

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

[68] 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.

[69] 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.

[70] 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.

[71] 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.

[72] 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.

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

[74] 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.

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

[76] 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.

[77] 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.

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

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

[80] 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.

[81] 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.

[82] 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.

[83] 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.

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

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

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

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

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

[89] 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.

[90] 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.

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

[92] 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.

[93] 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.

[94] 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.

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

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

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

[98] 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.

[99] 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.

[100] "Improved Mean Absolute Error for Learning Meaningful Patterns from Abnormal Training Data," Tech. Rep.

[101] 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.

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

[103] 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.

[104] 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.

[105] 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.

[106] 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.

[107] 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.

[108] 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.

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

[110] 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.

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

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

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

[114] 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.

[115] 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.

[116] 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.

[117] 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.

[118] 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.

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

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

[121] 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.

[122] 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.

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

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

[125] 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.

[126] 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.

[127] 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.

[128] 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.

[129] 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.

[130] 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.

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

[132] 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.

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

[134] 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.

[135] 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.

[136] 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.

[137] 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.

[138] 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.

[139] 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.

[140] 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.

[141] 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.

[142] 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.

[143] 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.

[144] 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.

[145] 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.

[146] 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.

[147] 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.

[148] 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.

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

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

[151] 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.

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

[153] 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.

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

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

[156] 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.

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

[158] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[159] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[160] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[161] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit *et al.*, "Openimages: A public dataset for large-scale multi-label and multi-class image classification," *Dataset available from https://github. com/openimages*, vol. 2, p. 3, 2017.

[162] 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.

[163] 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.

[164] 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.

[165] 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.

[166] T. Kaneko, Y. Ushiku, and T. Harada, "Label-Noise Robust Generative Adversarial Networks," 2018.

[167] 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.

[168] 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.

[169] 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.