Image classification with deep learning in the presence of noisy labels: A survey

Görkem Algan [a,*,1,2], Ilkay Ulusoy [b,2]

[a] *ASELSAN, Balikhisar mah. Cankiri bulvari 7.km No:89, 06750, Ankara, Turkey*
[b] *METU, Universiteler mah. Dumlupinar mah.Electrical-Electronics Engineering, Department A-410, 06800, Ankara, Turkey*

ARTICLE INFO

ABSTRACT

Image classification systems recently made a giant leap with the advancement of deep neural networks. However, these systems require an excessive amount of labeled data to be adequately trained. Gathering a correctly annotated dataset is not always feasible due to several factors, such as the expensiveness of the labeling process or difficulty of correctly classifying data, even for the experts. Because of these practical challenges, label noise is a common problem in real-world datasets, and numerous methods to train deep neural networks with label noise are proposed in the literature. Although deep neural networks are known to be relatively robust to label noise, their tendency to overfit data makes them vulnerable to memorizing even random noise. Therefore, it is crucial to consider the existence of label noise and develop counter algorithms to fade away its adverse effects to train deep neural networks efficiently. Even though an extensive survey of machine learning techniques under label noise exists, the 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 noise structure and use this information to avoid the adverse effects of noisy labels. Differently, methods in the second group try to come up with inherently noise robust algorithms by using approaches like robust losses, regularizers or other learning paradigms.

1. 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 for representation learning [9,10], it is shown that these powerful models can overfit to even complete random noise [11]. Various works are devoted to explain this phenomenon [12,13], yet regularizing deep neural networks (DNNs) while avoiding overfitting 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 train deep neural networks effectively in the presence of noise.

There are two kinds of noise in the literature: feature noise and label noise [14]. Feature noise corresponds to the corruption in observed data features, while label noise means the change of label from its actual class. Even though both noise types may cause a significant decrease in the 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 [14,18,19]. This is due to several factors; the 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]. 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 significant 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 via search engines or social media. Similarly, the labeling process can be crowdsourced with the help of systems like Amazon Mechanical Turk,[3] Crowdflower,[4] which decrease the cost of labeling

---

* Corresponding author.
E-mail addresses: e162565@metu.edu.tr (G. Algan), ilkay@metu.edu.tr (I. Ulusoy).
[1] ASELSAN, Ankara.
[2] Middle East Technical University, Electrical-Electronics Engineering, Ankara.

[3] http://www.mturk.com
[4] http://crowdflower.com

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 adversarial poisoning purposes [20,21]. Being a natural outcome of dataset collection and labeling process 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 [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 in the presence of 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 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 with a clear overview of the literature.

This paper is organized as follows. Section 2 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 Sections 3–4. In Section 5 widely used experimental setups are presented, and leaderboard on a benchmarking dataset is provided. Finally, Section 6 concludes the paper.

2. Preliminaries

This section introduces necessary concepts for a better understanding of the paper. Firstly, the problem statement for supervised learning in the presence of noisy labels is given. Secondly, types of label noises are presented. Finally, sources of label noise are discussed.

2.1. Problem statement

Classical supervised learning consists of an input dataset $S = \{(x_1, y_1), \ldots, (x_N, y_N)\} \in (X, Y)^N$ drawn according to an unknown distribution $\mathcal{D}$, over (X, Y). The learning objective 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 an example $(x_i, y_i) \in (X, Y)$, $l(f_\theta(x_i), y_i)$ evaluates how good is the classifier prediction. Then, for any classifier f, expected risk is defined as follow, where E denotes the expectation over distribution $\mathcal{D}$.

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

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

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

As a result, obtained parameters by minimizing over $\mathcal{D}_n$ are different from 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, the 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 the literature, and they will be further evaluated in the upcoming sections.

2.2. 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 depends 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 image features but depends on its class; $p_e = p(e|y)$. That means data from a particular class 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. Generating xy-dependent synthetic noise is harder than the previous two models; therefore, some works tried to provide a generic framework by either checking the complexity of data [23] or their position in feature space [24]. All these types of noises are illustrated in Fig. 1

The case of multi-labeled data, in which each instance has multiple labels given by different annotators, is not considered here. In that scenario, works show that modeling each labeler's characteristics and using this information during training significantly boosts the performance [25]. 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 [26]; therefore, they can be modeled as random noise. On the other hand, labelers with better accuracies than random selection can be modeled by y-dependent or xy-dependent noise. As a result, the labeler's characteristic is not introduced as an extra ingredient in these definitions.

2.3. 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 [25,27,28], semantic segmentation [29–31], crowd-sourcing [32], social network tagging [33,34], financial analysis [35] and many more. This work focuses on various solutions
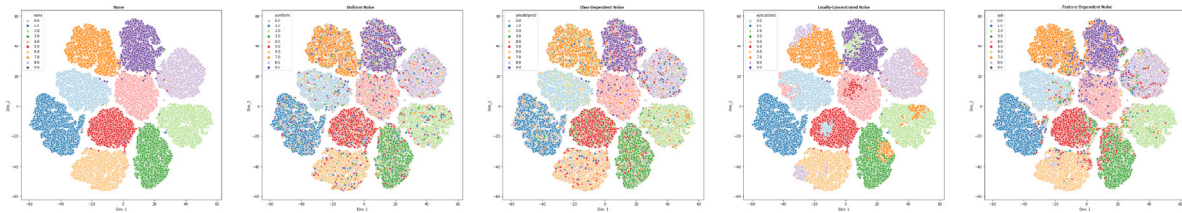
Fig. 1. T-SNE plot of data distribution of MNIST dataset in feature space for 25% noise ratio. (a) clean data (b) random noise (c) y-dependent noise which is still randomly distributed in feature domain (d) xy-dependent noise in locally concentrated form (e) xy-dependent noise that is concentrated on decision boundaries.

to such problems, but it may be helpful to investigate the causes of label noise 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 [36–41]. Nevertheless, 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 [26]. 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. 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 [26].

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 [42,43]. This complexity can be due to the subjectiveness of the task for human experts or the lack of annotator experience. Considering the fields where the accurate diagnosis is of crucial importance, overcoming this noise is of great interest.

Lastly, label noise can intentionally be injected in purpose of regularizing [44] or data poisoning [20,21].

2.4. 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 within 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 develop inherently noise robust strategies without explicit modeling of the noise structure. These approaches assume that the classifier is not too sensitive to the noise, and performance degradation results from 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 Sections 3 and 4. Table 1 presents all the 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 algorithms, and they may belong to more than one category. However, for the sake of integrity, they are placed in the subclass of most resemblance.

3. Noise model based methods

In the presence of noisy labels, the learning objective is to find the best estimator for hidden distribution $\mathcal{D}$, while iterating over distribution $\mathcal{D}_n$. If the mapping function $M : \mathcal{D} \rightarrow \mathcal{D}_n$ 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.

3.1. Noisy channel

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

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

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* [45].

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 [49], or predefined nodes [157]. These restrictions are determined by human experts, which allows additional noise information to be inserted into the training procedure.

Table 1
Existing methods to deal with label noise in the literature.

Noise Model Based Methods	**1.** Noisy Channel *a.* Explicit calculation: predictions on noisy data [45], predictions on clean data [46] easy data [47] *b.* Iterative calculation: EM [27,48,49], fully connected layer [50], anchor point estimate [51] Dirichlet-distribution [52] *c.* Complex Noisy Channel: noise type estimation [53], relevance estimation [54]
	2. Label Noise Cleaning *a.* Using data with clean labels: train on clean set [55], ensemble [56], graph-based [57] *b.* Using data with both clean and noisy labels: iteratively correct [58], correct for fine-tune [59] *c.* Using data with just noisy labels: calculate posterior [60], posterior with compatibility [61] consistency with model [62–64], ensemble [65], prototypes [66], quality embedding [67] partial labels [68]
	3. Dataset Pruning *a. Data pruning* network prediction based [69], ensemble of filters [70,71], according to noise rate [72] transfer learning [73], cyclic state [74], K-means [75] *b. Label pruning* semi-supervised learning [76–79], relabeling [80–82]
	4. Sample Choosing a. curriculum learning: Screening loss [83], teacher–student [84] selecting uncertain samples [85], curriculum loss [86], data complexity [87] consistency with model [88] b. multiple classifiers: Consistency of networks [89], co-teaching [90–93]
	5. Sample Importance Weighting Meta task [94–96], siamese network [97], pLOF [28], abstention [98] estimate noise rate [99,100], similarity loss [101], transfer learning [102], θ-distribution [103]
	6. Labeler Quality Assessment EM [26,104,105], trace regularizer [106], crowd-layer [107], image difficulty estimate [108] consistency with network [109], omitting probability variable [110] softmax layer per labeler [25]
Noise Model Free Methods	**1.** Robust Losses Non-convex loss functions [111–113], 0-1 loss surrogate [114], MAE [115], IMEA [116] Generalized cross-entropy [117], symmetric loss [118], unbiased estimator [119], modified cross-entropy for omission [120], information theoretic loss [121] linear-odd losses [122], classification calibrated losses [123], SGD with robust losses [124]
	2. Meta Learning Choosing best methods [125], pumpout [126], noise tolerant parameter initialization [127], knowledge distillation [128,129], gradient magnitude adjustment [130,131], meta soft labels [132]
	3. Regularizers Dropout [133], adversarial training [134], mixup [135], label smoothing [136,137] pre-training [138], dropout on final layer [139], checking dimensionality [140], auxiliary image regularizer [141]
	4. Ensemble Methods LogitBoost&BrownBoost [142], noise detection based AdaBoost [143], rBoost [144] RBoost1&RBoost2 [145], robust multi-class AdaBoost [146]
	5. Others Complementary labels [147,148], autoencoder reconstruction error [149] minimum covariance determinant [150], less noisy data [151], data quality [152], prototype learning [153,154], multiple instance learning [155,156]

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

3.1.1. Explicit calculation
Noise transition matrix is calculated explicitly, and then the 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 $p(y = \tilde{y}_i|x_i) = 1$, [45] constructs T just based on noisy class probability estimates of a pre-trained model, so-called *confusion matrix*. A similar approach is followed in [46]; however, the noise transition matrix is calculated from the network's confusion matrix on the clean subset of data. Two datasets are gathered in [47], 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. Another method proposed in [50] calculates the confusion matrix on both noisy data and clean data. Then, the difference between these two confusion matrices gives T.

3.1.2. Iterative calculation
Noise transition matrix is estimated incrementally during the training of the base classifier. In [48,49] expectation–maximization (EM) [158] 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 [27]. [50] adds a linear fully connected layer as a last layer of the base classifier, which is trained to adapt noise behavior. 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. [51] 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
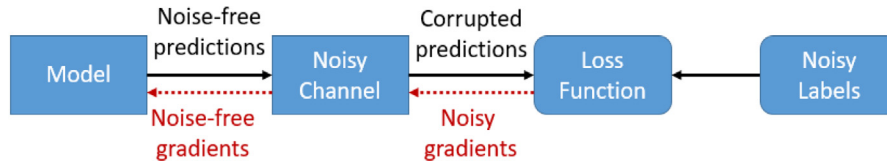
Fig. 2. Noise can be modeled as a noisy channel on top of base classifier. Noisy channel adapts the characteristic of the noise so that base classifier is fed with noise-free gradients during training.

iteratively to minimize loss during training. Instead of using soft-max probabilities, [52] models noise transition matrix in Bayesian form by projecting it into a Dirichlet-distributed space.

3.1.3. 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 [53], 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 [54] 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 does not get penalized much for it.

3.2. Label noise cleaning

An obvious solution to noisy labels is to identify and correct suspicious labels to their corresponding true classes. Cleaning the whole dataset manually can be costly; therefore, some works propose to pick only suspicious samples to be sent to a human annotator to reduce the cost [41]. However, this is still not a scalable approach. As a result, various algorithms are proposed in the literature. Including the label correction algorithm, the empirical risk takes the following form

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

where $G(\tilde{y}_i, x_i) = p(y_i|\tilde{y}_i, x_i)$ represents the label cleaning algorithm. Label cleaning algorithms rely on a feature extractor to map data to the feature domain to investigate noisiness. While some works use a pre-trained network as the feature extractor, others use the base classifier as it gets more and more accurate during training. This approach results in an iterative framework: as the classifier gets better, the label cleaning is more accurate, and as the label quality gets better, the classifier gets better. From this point of view, label cleaning can be viewed as a dynamically evolving component of the system instead of preprocessing of the 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 cleaning can be separated according to their need for clean data or not.

3.2.1. 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, [55] uses alpha blending of given noisy labels and predicted labels. An ensemble of networks trained with different subsets of the dataset is used in [56]. If they all agree on the label, it is changed to the predicted label; otherwise, it is set to a random label. Instead of keeping the noisy label, setting it randomly helps break the noise structure and makes noise more uniformly distributed in label space. In [57] a graph-based approach is used, where a conditional random field extracts relation among noisy labels and clean labels.

3.2.2. 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 [58], 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 are trained jointly so that label cleaning 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 [59] in teacher–student manner. First, the student is trained on noisy data. Then features are extracted from the 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.

3.2.3. Using data with just noisy labels

Noise-free data is not always available, so the primary approach in this situation is to estimate cleaner posterior label distribution incrementally. 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 delusional top training accuracy. Therefore, additional regularizers are commonly used to make label posterior distribution even. A joint optimization framework for both training base classifier and propagating noisy labels to cleaner labels is presented in [60]. Using expectation–maximization, both classifier parameters and label posterior distribution is estimated to minimize the loss. A similar approach is used in [61] with additional compatibility loss conditioned on label posterior. Considering noisy labels are in the minority, this term assures posterior label distribution does not diverge too much from the given noisy label distribution so that majority of the clean label contribution is not lost. [62,63] deploy a confidence policy where labels are determined by either network output or given noisy labels, depending on the confidence of the model's prediction. Arguing that, in the case of noisy labels, the model first learns correctly labeled data and then overfits to noisy data, [64] aims to extract the probability of a sample being noisy or not from its loss value. To achieve this, the loss of each instance is fitted by a beta mixture model, which models the label noise in an unsupervised manner. [65] 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 is trained on clean data to predict true labels of noisy data. Afterward, these two subsets of data are merged to create the final

enhanced dataset. [66] constructs prototypes that can represent deep feature distribution of the corresponding class for each class. Then corrected labels are found by checking similarity among data samples and prototypes. [67] introduces a new parameter, namely *quality embedding*, which represents the trustworthiness of data. Depending on two latent variables, true class probability and quality embedding, an additional network tries to extract each instance's true class. In a 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, [68] uses one network to find and estimate easy missing labels and another network to be trained on this corrected data. [159] 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.

3.3. 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 the harmful 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.

There are two alternative ways for data pruning. The first option is to remove noisy samples completely and train the classifier on the pruned dataset. The second option is to remove just labels of noisy data and transform the dataset into two subsets as; labeled and unlabeled data. Then semi-supervised learning algorithms can be employed on the resultant dataset.

3.3.1. Removing data

The most straightforward approach is to remove instances misclassified by the base network [69]. [70] uses an ensemble of filtering methods, where each of them assigns a noisiness level for each sample. Then, these predictions are combined, and data with the highest noisiness level predictions are removed. [71] extends this work with label correction. If the majority of noise filters predict the same label for the noisy instance, its label is corrected to the predicted label. Otherwise, it is removed from the dataset. In [72], with the help of a probabilistic classifier, training data is divided into two subsets: confidently clean and noisy. Noise rates are estimated according to the sizes of these subsets. Finally, relying on the base network's output confidence in data instances, the number of most unconfident samples is removed according to the estimated noise rate. In [73], 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 [74], the learning rate is adjusted cyclicly to change network status between underfitting and overfitting. Since, while underfitted, noisy samples cause high loss, samples with large noise during this cyclic process are removed. [75] first train network on noisy data and extract feature vectors by using this model. Afterward, data is clustered with the K-means algorithm running on extracted features, and outliers are removed. [160] provides a comparison of performances of various noise-filtering methods for crowd-sourced datasets.

3.3.2. Removing labels

The simplest option is to employ straightforward semi-supervised training on labeled and unlabeled data [76]. Alternatively, label removing can be done iteratively in each epoch to update the dataset for better utilization of semi-supervised learning dynamically. [77] uses consistency among the given label and moving average of model predictions to evaluate if the provided label is noisy or not. Then the model is trained on clean samples on the next iteration. This procedure continues until convergence to the best estimator. The same approach is used in [78] with a little tweak. Instead of comparing with given labels, the moving average of predictions is compared with predicted labels in the current epoch. To avoid the data selection biased caused by one model, [79] uses two models to select an unlabeled set for each other. Afterward, each network is trained in a semi-supervised learning manner on the dataset chosen by its peer network. Another approach in this class is to train a network on labeled and unlabeled data and then use it to relabel noisy data [80,82]. Assuming that correctly labeled data account for the majority, [81] proposes splitting datasets into labeled and unlabeled subgroups randomly. Then, labels are propagated to unlabeled data using a similarity index among instances. This procedure is repeated to produce multiple labels per instance, and then the final label is set with majority voting.

3.4. 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 the classifier finding its way easier in the presence of noisy labels. It can be formulated as follows

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

where $V(x_i, y_i) \in \{0, 1\}$ is a binary operator that decides to whether use the given data (x_i, y_i) or not. If $V(x_i, y_i) = 1$ for all data, then it turns out to be classical risk minimization (Eq. (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 Section 3.3. 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 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 subsections.

3.4.1. Curriculum learning

Curriculum learning (CL) [161], inspired from human cognition, proposes to start from easy samples and go through harder samples to guide training. This learning strategy is also called *self-paced learning* [162,163] when prior to sample hardness is not known and inferred from the loss of the current model on that sample. In the noisy label framework, clean labeled data can be accepted as an easy task, while noisily labeled data is the harder task. Therefore, the idea of CL can be transferred to label noise setup as starting from confidently clean instances and go through noisier samples as the classifier gets better. Various screening loss functions are proposed in [83] to sort instances according to their noisiness level. The teacher–student approach is implemented in [84], where the teacher's task is to choose confidently clean samples for the student. Instead of using a predefined curriculum, the teacher constantly updates its curriculum depending on the student's outputs. Arguing that CL slows down the learning speed,

since it focuses on easy samples, [85] suggests choosing uncertain samples that are mispredicted sometimes and correctly on others during training. These samples are assumed to be probably not noisy since noisy samples should be mispredicted 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 [86]. In [87], data is split into subgroups according to their complexities. Since less complex data groups are 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 [88], 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 a 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.

3.4.2. Multiple classifiers

Some works use multiple classifiers to help each other to choose the next batch of data to train on. This is different from the teacher–student approach since none of the networks supervise the other but rather help each other out. Multiple models 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 essential. They are most likely to be initialized with a different subset of the data. If they have the same weight initializations, there is no update since they agree to disagree with labels. In [89] 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 [90], 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 the 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 the increasing number of epochs [91,92]. Another work using co-teaching first trains two networks on a selected subset for a given number of epochs and then moves to the full dataset [93].

3.5. 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 a 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 Section 3.4. Differently, here β is not binary and has a different value for each instance. 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. As a result, 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 [50]. However, this utilizes information poorly; moreover, clean data is not always available. Works of [94] and [95], uses meta-learning paradigm to determine the weighting factor. In each iteration, gradient descent step on the given mini-batch for weighting factor is performed so that it minimizes the loss on clean validation data. A similar method is adopted in [96], 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 [97]. 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 the whole dataset. [28] also iteratively separates noisy samples and clean samples. On top of that, not to miss valuable information from clean hard samples, noisy data are weighted according to their noisiness level, estimated by pLOF [164]. [98] 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 [99], 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 [100]. In [101], the weighting factor is determined by checking instance similarity to its representative class prototype in the feature domain. [102] formulates the problem as transfer learning where the source domain is noisy data, and the target domain is a clean subset of data. Then weighting in the source domain is arranged in a way to minimize target domain loss. [103] 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.

3.6. Labeler quality assessment

As explained in Section 2.3, there can be several reasons for the dataset to be labeled by multiple annotators. Each labeler may have a different level of expertise, and their labels may occasionally contradict each other. This is a typical case in crowd-sourced data [165–167] or datasets that require a high level of expertise such as medical imaging [19]. Therefore, modeling and using labeler characteristics can significantly increase performance [25].

In this setup, there are two unknowns; noisy labeler characteristics and ground truth labels. One can estimate both with expectation–maximization [26,104,105]. If noise is assumed to be y-dependent, the labeler characteristic can be modeled with a noise transition matrix, just like in Section 3.1. [106] adds a regularizer to the loss function, which is the sum of traces of annotator confusion matrices, to force sparsity on each labeler's estimated confusion matrix. A similar approach is implemented in [107], where a crowd-layer is added to the end of the network. In [108], xy-dependent noise is also considered by taking image complexities into account. Human annotators and computer vision systems are used mutually in [109], where consistency among predictions of these two components is used to evaluate labelers' reliability. [110] deals with the noise when the labeler omits a tag in the image. Therefore, instead of the noise transition matrix for labelers, the omitting probability variable is used, which is
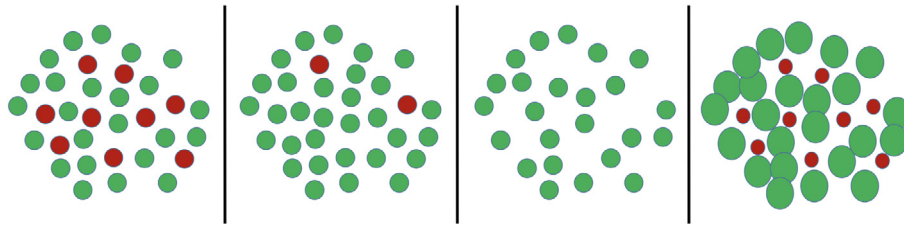
Fig. 3. Illustration of different types of algorithms. Starting from left; (1) representation of samples from a single class in the 2d-space. Green samples represent the clean samples, and red ones represent noisy samples. (2) label noise cleaning algorithms aim to correct the labels of noisy data (3) dataset pruning methods aim to eliminate noisy data (or just their labels) (4) sample importance weighting algorithms aim to up-weight clean samples and down-weight noisy samples (which is illustrated by size). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

estimated together with the true class using the expectation–maximization algorithm. Separate softmax layers are trained for each annotator in [25] 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.

3.7. Discussion

A visual illustration of some of the methods is presented in Fig. 3. Noise model based methods are heavily dependent on the accurate estimate of the noise structure. This brings a dilemma. For a 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 estimation and classification. However, it is essential to prevent the system diverging from reality. Therefore regularizing noise estimates and not letting it getting delusional is crucial. To achieve this, the literature commonly makes 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 be inserted into the system for a 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.

4. 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 an anomaly; therefore, these methods are similar to overfit avoidance. They commonly rely on the classifier's internal noise tolerance and aim to boost performance by regularizing undesired memorization of noisy data. Various methodologies are presented in the following subsections.

4.1. Robust losses

A loss function is said to be noise robust if the classifier learned with noisy and noise-free data achieves the same classification accuracy [111]. Algorithms under this section aim to design loss function so that the existence of 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 [111], 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 [112,113] 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 [114], 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 [115], where it is shown empirically that mean absolute value of error is more noise tolerant. [116] 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 [116], where gradients are scaled with a hyper-parameter to adjusts weighting variance of MAE. [117] also argues that MAE provides a much lower 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, the loss function characteristic 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, [118] 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, [119] provides two surrogate loss functions using the prior information about label noise, namely, an unbiased and a weighted estimator of the loss function. [120] 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 the network less for producing wrong but confident predictions since these labels are more likely to be noisy. Instead of using distance-based loss, [121] proposes to use information-theoretic loss, in which determinant based mutual information [168] between given labels and predictions are evaluated for loss calculation. Weakly supervised learning with noisy labels are considered in [122], and necessary conditions for loss to be noise tolerant are drawn. [123] shows that classification-calibrated loss functions are asymptotically robust to symmetric label noise. Stochastic gradient descent with robust losses is analyzed in general [124] and shown to be more robust to label noise than its counterparts.

4.2. 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 can learn complex functions on their own, there remain 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 [169,170]. Algorithms belonging to this category usually implement an additional learning loop for the meta objective, optimizing the base learning procedure. 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 the conventional training process.

Designing a task beyond classical supervised learning in a 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 the family of methods, for a given noisy dataset in [125]. *Pumpout* [126] 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) [170] seeks optimal weight initialization that can easily be fine-tuned for the desired objective. A similar mentality is used in [127] for noisy labels, which aims to find noise-tolerant model parameters that are less prone to noise under teacher–student training framework [171,172]. Multiple student networks are fed with data corrupted by synthetic noise. A meta objective is defined to maximize consistency with teacher outputs 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 [128] is to train a teacher network in a clean dataset and transfer its knowledge to the student network to guide the training process in the presence of mislabeled data. They used *distillation* technique proposed in [173] for controlled transfer of knowledge from teacher to student. A similar methodology of using *distillation* and label correction in the human pose estimation task is implemented in [129]. In [130,131], the target network is trained on excessive noisy data, and the confidence network is trained on a clean subset. Inspiring from [169], 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. [132] uses clean data to produce soft labels for noisy data, for which the classifier trained on it would give the best performance on the clean data. As a result, it seeks optimal label distribution to provide the most noise robust learning for the base classifier.

4.3. 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 [133], adversarial training [134], mixup [135], label smoothing [136,137]. [138] shows that pre-training has a regularization effect in the presence of noisy labels. In [139] 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 [133]. [140] proposes a complexity measure to understand if the network starts to overfit. It is shown that learning consists of two steps: (1) dimensionality compression, which models low-dimensional subspaces that closely match the underlying data distribution, (2) dimensionality expansion, which steadily increases subspace dimensionality to overfit the data. The key is to stop before the second step. *Local intrinsic dimensionality* [174] is used to measure the complexity of the trained model and stop before it starts to overfit. [141] 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.

4.4. Ensemble methods

It is well known that bagging is more robust to label noise than boosting [175]. 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 [142]. Therefore, noise-robust alternatives of AdaBoost is proposed in literature, such as noise detection based AdaBoost [143], rBoost [144], RBoost1&RBoost2 [145] and robust multi-class AdaBoost [146].

4.5. Others

Complementary labels define classes that observations do not belong to. For example, in the case of ten classes, there is one true class and nine complimentary classes for each instance. Since annotators are less likely to mislabel, some works propose to work in complementary label space [147,148]. [149] uses reconstruction error of autoencoder to discriminate noisy data from clean data, arguing that noisy data tend to have bigger reconstruction error. In [150], the base model is trained with noisy data. An additional generative classifier is trained on top of the 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 [151], a special setup is considered where dataset consists of noisy and *less-noisy* data for binary classification task. [152] aims to extract the quality of data instances. Assuming that the training dataset is generated from a mixture of the 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 [153,154] 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 made, the existence of noisy labels along with true labels in a bag has less impact on learning. In [155], authors propose to effectively choose training samples from each bag by minimizing the total bag level loss. Extra model is trained in [156] as an attention model, which determines parts of the images to be focused on. The aim is to focus on a few regions on correctly labeled images and not focus on any region for mislabeled images.

4.6. Discussion

Overall, methods belonging to this category 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 the 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.

5. Experiments

This section discusses how the proposed algorithms from the literature conduct experiments to test their robustness against label noise. In general, for quick implementation and testing, most of the works start by testing on toy datasets (MNIST [176], MNIST-Fashion [177], CIFAR10&CIFAR100[178]) with synthetic label noise. However, as explained in Section 2.2, there are various types of artificial noises. Moreover, each work experiment with a different model architecture and hyper-parameter set. Therefore, results on these datasets are not appropriate for a fair comparison of the algorithms. They are instead used as a proof of concept for the proposed algorithm.

Some works from the literature use two alternative datasets. The first one is the Food101N dataset [101] containing 310k images of food recipes belonging to 101 different classes. However, its noise ratio is pretty small (around 20%), making it inadequate to evaluate noise robust algorithms' performance. The second option is the WebVision dataset [179] containing 2.4 million images crawled from Flickr website and Google Images search. This is a big dataset, which requires a lot of computational power to run algorithms. Some works conduct tests on this dataset by using data only from the first 50 classes, aiming to make it computationally feasible. But still, WebVision fails to provide a benchmarking dataset for the evaluation of noise robust algorithms.

To fill the absence of a benchmarking dataset, [53] collects a large amount of images from the web with labels interpreted from the surrounding user tags. As a result, it has real-world noisy labels with an estimated noise ratio of around 40%. Dataset consists of one million images belonging to 14 different classes. 50K, 14K and 10K additional images with verified clean labels for train, validation and test purposes. We observed a high majority of the methods do not use additional 50K clean data for training. Furthermore, the literature seems to have a consensus on the experimental setup. All methods use the same model architecture of ResNet50 [2] with pre-trained parameters on Imagenet [180] and stochastic gradient descent optimizer. Considering the identical experimental setup and real-world noisiness of the dataset, the Clothing1M dataset is widely accepted as a benchmarking dataset.

We listed (to the best of our knowledge) all of the results presented on this dataset in Table 2. We collected results only from the works trained on 1M noisy training data without additional 50K clean data for a fair evaluation. We sorted algorithms according to their test accuracy. Nevertheless, it should be noted that each method has its pros and cons, such as computational cost, memory requirements, etc.

Table 2
Leaderboard for algorithms tested on the Clothing1M dataset. All results are taken from the corresponding paper. For fair evaluation only the works which did not used additional 50k clean training data are presented.

Method	Category	Accuracy
[132]	Meta Learning	76.02
[79]	Dataset Pruning	74.76
[101]	Sample Importance Weighting	74.69
[66]	Label Noise Cleansing	74.45
[51]	Noisy Channel	74.18
[75]	Dataset Pruning	73.77
[96]	Sample Importance Weighting	73.72
[61]	Label Noise Cleansing	73.49
[127]	Meta Learning	73.47
[116]	Robust Losses	73.20
[52]	Noisy Channel	73.07
[153]	Others	72.50
[121]	Robust Losses	72.46
[60]	Label Noise Cleansing	72.23
[63]	Label Noise Cleansing	71.74
[157]	Noisy Channel	71.10
[118]	Robust Losses	71.02
[64]	Label Noise Cleansing	71.00

6. 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. Despite its importance for supervised learning in practical applications, it is also an important step to collect datasets from the web [181,182], design networks that can learn from unlimited web data with no human supervision [37–40]. Furthermore, beside image classification, there are more fields where dealing with mislabeled instances is important, such as generative networks [183,184], semantic segmentation [29–31], sound classification [185] 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-suited method according to need. For example, if noise can be represented as a 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. Unlike noise model-based algorithms, noise model free methods do not depend on any prior information about the noise structure. 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 in the presence of 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 impact of label noise on deep networks can be a fruitful future research topic. [186] shows that each layer of CNN learns to extract different features from the data. Moreover, learned representations form a hierarchical pattern, where each layer learns more complex features from the previous layer. A fully connected layer uses features from the last layer to interpret the corresponding label on the final layer. Understanding which parts of the network is highly affected by label noise may help analyze the adverse effect of the label noise on neural networks. For example, if initial layers are affected, one can conclude that learned primitive features are corrupted, so the rest of the network cannot be adequately trained. On the other hand, if final convolution layers are affected, it can be said that the network cannot form the hierarchical feature pattern throughout the convolutional layers. Alternatively, if the convolutional layers are not affected but the fully connected layer is the cause of the problem, it can be concluded that feature representation learning is not corrupted, but the network cannot correctly interpret meanings from the extracted features. Moreover, it can be interesting to investigate the cause of the problem for different types of label noise models presented in Section 2.2. If for different noise models, different parts of the network are affected, one can analyze the true nature of the label noise in the dataset by checking corruption in the neural network layers.

Alternatively, the question of how to train in the existence of both attribute and label noise is an understudied problem with significant potential for practical applications [55]. [118] shows noisy labels degrades the learning, especially for challenging samples. So, instead of overfitting to noisy samples, underfitting to challenging samples may be the reason for the 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 [81]. 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 [176–178,187,188]. However, this prevents fair comparison and evaluation of algorithms since each work adds its own noise type. Some large datasets with noisy labels are proposed in literature [9,101,179,189]. 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 cannot 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 has both noisy and verified labels; thus, the noise rate can be adjusted as desired.

Minimal 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 [55], due to its cost or privacy. Effectively learning from a small amount of noisy data with no ground truth can significantly improve autonomous medical diagnosis systems. Even though some pioneer researches are available [25,27,28], there is still much more to be explored.

The ability to effectively learn from noisily labeled data brings up big opportunities for practical applications of machine learning algorithms. The bottleneck of data collection can easily be resolved with the massive amount of data collected from the web. Labels for this data can be assigned with simple algorithms

(such as interpreting from the surrounding text [53]). By effectively dealing with noisy labels, deep learning algorithms can be fed with massive datasets. Moreover, there are research opportunities for alternative usage of semi-supervised learning algorithms along with noisily labeled data. Common usage of semi-supervised learning methods for noisily labeled data is to remove the labels of noisy data and then train with conventional semi-supervised learning methods. Alternatively, algorithms can be developed to use three types of data; cleanly labeled data, noisily labeled data, and unlabeled data. With the help of these algorithms, a massive amount of unlabeled and noisy data can be effectively used under the supervision of a small cleanly annotated data.

Besides classification, knowledge of learning from noisily labeled data can be used in alternative fields by transforming the task. For example, in a multi-labeled dataset, where each instance belongs to multiple classes, not all classes are equally relevant. One can assume labels with a small resemblance to the data sample as noisy and employ algorithms designed for learning from noisy labels [190]. Similarly, [191] divides an untrimmed video into smaller parts and aims to find the video snippet most relevant to the video tag. Irrelevant video parts are assumed to be noisily labeled. Even though the original dataset does not have noisy labels, it can be transformed to use algorithms from the field. As a result, learning from noisy labels has many potentials in various areas besides straight image classification.

CRediT authorship contribution statement

Görkem Algan: Conceptualization, Validation, Investigation, Writing - original draft. **Ilkay Ulusoy:** Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Krizhevsky, I. Sutskever, 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, 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, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.

[4] R. Girshick, J. Donahue, T. Darrell, 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, 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, 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, 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, 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, N. Shavit, Deep learning is robust to massive label noise, 2017, arXiv preprint arXiv:1705.10694.

[10] A. Drory, O. Ratzon, S. Avidan, R. Giryes, The resistance to label noise in K-NN and CNN depends on its concentration, 2018, arXiv preprint arXiv:1803.11410.

[11] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning requires rethinking generalization, in: International Conference on Learning Representations, 2017.

[12] D. Krueger, N. Ballas, S. Jastrzebski, D. Arpit, M.S. Kanwal, T. Maharaj, E. Bengio, A. Fischer, 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, Vol. 70, JMLR. org, 2017, pp. 233–242.

[14] X. Zhu, X. Wu, Class noise vs. attribute noise: A quantitative study, Artif. Intell. Rev. 22 (3) (2004) 177–210.

[15] B. Frénay, M. Verleysen, Classification in the presence of label noise: a survey, IEEE Trans. Neural Netw. Learn. Syst. 25 (5) (2014) 845–869.

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

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

[18] D.F. Nettleton, A. Orriols-Puig, A. Fornells, A study of the effect of different types of noise on the precision of supervised learning techniques, Artif. Intell. Rev. 33 (4) (2010) 275–306.

[19] M. Pechenizkiy, A. Tsymbal, S. Puuronen, 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] B. Li, Y. Wang, A. Singh, 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, P. Liang, Certified defenses for data poisoning attacks, in: Advances in Neural Information Processing Systems, 2017-Decem, 2017, pp. 3518–3530.

[22] D. Angluin, P. Laird, Learning from noisy examples, Mach. Learn. 2 (4) (1988) 343–370.

[23] L.P. Garcia, J. Lehmann, A.C. de Carvalho, A.C. Lorena, New label noise injection methods for the evaluation of noise filters, Knowl.-Based Syst. 163 (2019) 693–704.

[24] G. Algan, İ. Ulusoy, Label noise types and their effects on deep learning, 2020, arXiv preprint arXiv:2003.10471.

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

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

[27] Y. Dgani, H. Greenspan, 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.

[28] C. Xue, Q. Dou, X. Shi, H. Chen, P.-A. Heng, Robust learning at noisy labeled medical images: Applied to skin lesion classification, in: 2019 IEEE 16th International Symposium on Biomedical Imaging, ISBI 2019, IEEE, 2019, pp. 1280–1283.

[29] Z. Lu, Z. Fu, T. Xiang, P. Han, L. Wang, X. Gao, Learning from weak and noisy labels for semantic segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (3) (2016) 486–500.

[30] Y. Zhu, K. Sapra, F.A. Reda, K.J. Shih, S. Newsam, A. Tao, B. Catanzaro, Improving semantic segmentation via video propagation and label relaxation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8856–8865.

[31] D. Acuna, A. Kar, 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. 11075–11083.

[32] P. Welinder, S. Branson, P. Perona, S.J. Belongie, The multidimensional wisdom of crowds, in: Advances in Neural Information Processing Systems, 2010, pp. 2424–2432.

[33] Y. Cha, 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.

[34] Y. Wang, Y. Rao, X. Zhan, H. Chen, M. Luo, J. Yin, Sentiment and emotion classification over noisy labels, Knowl.-Based Syst. 111 (2016) 207–216.

[35] Y. Aït-Sahalia, J. Fan, D. Xiu, High-frequency covariance estimates with noisy and asynchronous financial data, J. Amer. Statist. Assoc. 105 (492) (2010) 1504–1517.

[36] F. Schroff, A. Criminisi, A. Zisserman, Harvesting image databases from the web, IEEE Trans. Pattern Anal. Mach. Intell. 33 (4) (2010) 754–766.

[37] R. Fergus, L. Fei-Fei, P. Perona, A. Zisserman, Learning object categories from internet image searches, Proc. IEEE 98 (8) (2010) 1453–1466.

[38] X. Chen, A. Shrivastava, A. Gupta, NEIL: Extracting visual knowledge from web data, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1409–1416, http://dx.doi.org/10.1109/ICCV.2013.178.

[39] S.K. Divvala, A. Farhadi, 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, http://dx.doi.org/10.1109/CVPR.2014.412.

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

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

[42] J. De Fauw, J.R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O'Donoghue, D. Visentin, et al., Clinically applicable deep learning for diagnosis and referral in retinal disease, Nat. Med. 24 (9) (2018) 1342.

[43] 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 316 (22) (2016) 2402–2410.

[44] L. Xie, J. Wang, Z. Wei, M. Wang, 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.

[45] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, 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.

[46] D. Hendrycks, M. Mazeika, D. Wilson, K. Gimpel, Using trusted data to train deep networks on labels corrupted by severe noise, in: Advances in Neural Information Processing Systems, 2018, pp. 10456–10465.

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

[48] A.J. Bekker, 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, E. Ben-Reuven, Training deep neural-networks using a noise adaptation layer, 2016.

[50] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, R. Fergus, Training convolutional networks with noisy labels, in: International Conference on Learning Representations, 2015.

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

[52] J. Yao, H. Wu, Y. Zhang, I.W. Tsang, J. Sun, Safeguarded dynamic label regression for noisy supervision, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, Association for the Advancement of Artificial Intelligence (AAAI), 2019, pp. 9103–9110, http://dx.doi.org/10.1609/aaai.v33i01.33019103, arXiv:1903.02152.

[53] T. Xiao, T. Xia, Y. Yang, C. Huang, 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.

[54] I. Misra, C. Lawrence Zitnick, M. Mitchell, 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.

[55] L. Jaehwan, Y. Donggeun, K. Hyo-Eun, Photometric transformer networks and label adjustment for breast density prediction, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2019.

[56] B. Yuan, J. Chen, W. Zhang, H.S. Tai, S. McMains, Iterative cross learning on noisy labels, in: Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, Vol. 2018, WACV 2018, 2018, pp. 757–765, http://dx.doi.org/10.1109/WACV.2018.00088.

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

[58] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, 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.

[59] M. Dehghani, A. Mehrjou, S. Gouws, J. Kamps, B. Schölkopf, Fidelity-weighted learning, in: International Conference on Learning Representations, 2018.

[60] D. Tanaka, D. Ikami, T. Yamasaki, 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.

[61] K. Yi, 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.

[62] X. Liu, S. Li, M. Kan, S. Shan, X. Chen, Self-error-correcting convolutional neural network for learning with noisy labels, in: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2017, IEEE, 2017, pp. 111–117.

[63] S. Zheng, P. Wu, A. Goswami, M. Goswami, D. Metaxas, C. Chen, Error-bounded correction of noisy labels, in: International Conference on Machine Learning, 2020.

[64] E. Arazo, D. Ortego, P. Albert, N.E. O'Connor, K. McGuinness, Unsupervised label noise modeling and loss correction, in: International Conference on Machine Learning, 2019.

[65] J. Zhang, V.S. Sheng, T. Li, X. Wu, Improving crowdsourced label quality using noise correction, IEEE Trans. Neural Netw. Learn. Syst. 29 (5) (2017) 1675–1688.

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

[67] J. Yao, J. Wang, I.W. Tsang, Y. Zhang, J. Sun, C. Zhang, R. Zhang, Deep learning from noisy image labels with quality embedding, IEEE Trans. Image Process. 28 (4) (2018) 1909–1922.

[68] T. Durand, N. Mehrasa, G. Mori, Learning a deep convnet for multi-label classification with partial labels, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 647–657.

[69] S.J. Delany, N. Segata, B. Mac Namee, Profiling instances in noise reduction, Knowl.-Based Syst. 31 (2012) 28–40.

[70] L.P. Garcia, J.A. Sáez, J. Luengo, A.C. Lorena, A.C. de Carvalho, F. Herrera, Using the one-vs-one decomposition to improve the performance of class noise filters via an aggregation strategy in multi-class classification problems, Knowl.-Based Syst. 90 (2015) 153–164.

[71] J. Luengo, S.-O. Shim, S. Alshomrani, A. Altalhi, F. Herrera, CNC-NOS: Class noise cleaning by ensemble filtering and noise scoring, Knowl.-Based Syst. 140 (2018) 27–49.

[72] C.G. Northcutt, T. Wu, 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, 2017, arXiv:1705.01936.

[73] X. Wu, R. He, Z. Sun, T. Tan, A light CNN for deep face representation with noisy labels, IEEE Trans. Inf. Forensics Secur. 13 (11) (2018) 2884–2896, http://dx.doi.org/10.1109/TIFS.2018.2833032, arXiv:1511.02683.

[74] J. Huang, L. Qu, R. Jia, 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, 2019, pp. 3326–3334.

[75] K. Sharma, P. Donmez, E. Luo, Y. Liu, I.Z. Yalniz, NoiseRank: Unsupervised label noise reduction with dependence models, in: Eeuropean Conferance on Computer Vision, 2020, 2020.

[76] Y. Ding, L. Wang, D. Fan, 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.

[77] D.T. Nguyen, T.-P.-N. Ngo, Z. Lou, M. Klar, L. Beggel, T. Brox, Robust learning under label noise with iterative noise-filtering, 2019, arXiv:1906.00216.

[78] D.T. Nguyen, C.K. Mummadi, T.P.N. Ngo, T.H.P. Nguyen, L. Beggel, T. Brox, SELF: Learning to filter noisy labels with self-ensembling, 2019, arXiv:1910.01842.

[79] J. Li, R. Socher, S.C. Hoi, Dividemix: Learning with noisy labels as semi-supervised learning, in: International Conference on Learning Representations, 2020.

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

[81] J. Jiang, J. Ma, Z. Wang, C. Chen, X. Liu, Hyperspectral image classification in the presence of noisy labels, IEEE Trans. Geosci. Remote Sens. 57 (2) (2019) 851–865, http://dx.doi.org/10.1109/TGRS.2018.2861992, arXiv:1809.04212.

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

[83] B. Han, I.W. Tsang, L. Chen, P.Y. Celina, S.-F. Fung, Progressive stochastic learning for noisy labels, IEEE Trans. Neural Netw. Learn. Syst. (99) (2018) 1–13.

[84] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, L. Fei-Fei, Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels, in: International Conference on Machine Learning, 2018, pp. 2304–2313.

[85] H.-S. Chang, E. Learned-Miller, 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.

[86] Y. Lyu, I.W. Tsang, Curriculum loss: Robust learning and generalization against label corruption, 2019, arXiv:1905.10045.

[87] S. Guo, W. Huang, H. Zhang, C. Zhuang, D. Dong, M.R. Scott, 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.

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

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

[90] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, 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.

[91] X. Yu, B. Han, J. Yao, G. Niu, I.W. Tsang, M. Sugiyama, How does disagreement help generalization against label corruption? in: International Conference on Machine Learning, 2019.

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

[93] P. Chen, B.B. Liao, G. Chen, S. Zhang, Understanding and utilizing deep neural networks trained with noisy labels, in: International Conference on Machine Learning, 2019, pp. 1062–1070.

[94] M. Ren, W. Zeng, B. Yang, R. Urtasun, Learning to reweight examples for robust deep learning, in: International Conference on Machine Learning, 2018.

[95] S. Jenni, P. Favaro, Deep bilevel learning, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 618–633.

[96] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, D. Meng, Advances in Neural Information Processing Systems, 2019, pp. 1917–1928,

[97] Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song, 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.

[98] S. Thulasidasan, T. Bhattacharya, J. Bilmes, G. Chennupati, J. Mohd-Yusof, Combating label noise in deep learning using abstention, in: International Conference on Machine Learning, 2019.

[99] T. Liu, D. Tao, Classification with noisy labels by importance reweighting, IEEE Trans. Pattern Anal. Mach. Intell. 38 (3) (2015) 447–461.

[100] R. Wang, T. Liu, D. Tao, Multiclass learning with partially corrupted labels, IEEE Trans. Neural Netw. Learn. Syst. 29 (6) (2018) 2568–2580.

[101] K.-H. Lee, X. He, L. Zhang, 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.

[102] O. Litany, D. Freedman, Soseleto: A unified approach to transfer learning and training with noisy labels, in: International Conference on Learning Representations Workshop on Learning from Limited Labeled Data, 2018.

[103] W. Hu, Y. Huang, F. Zhang, R. Li, Noise-tolerant paradigm for training face recognition CNNs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11887–11896.

[104] V.C. Raykar, S. Yu, L.H. Zhao, A. Jerebko, C. Florin, G.H. Valadez, L. Bogoni, 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.

[105] Y. Yan, R. Rosales, G. Fung, R. Subramanian, J. Dy, Learning from multiple annotators with varying expertise, Mach. Learn. 95 (3) (2014) 291–327, http://dx.doi.org/10.1007/s10994-013-5412-1.

[106] R. Tanno, A. Saeedi, S. Sankaranarayanan, D.C. Alexander, N. Silberman, Learning from noisy labels by regularized estimation of annotator confusion, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11244–11253.

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

[108] J. Whitehill, T.-f. Wu, J. Bergsma, J.R. Movellan, 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.

[109] S. Branson, G. Van Horn, P. Perona, Lean crowdsourcing: Combining humans and machines in an online system, in: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2017, CVPR 2017, 2017, pp. 6109–6118, http://dx.doi.org/10.1109/CVPR.2017.647.

[110] H. Izadinia, B.C. Russell, A. Farhadi, M.D. Hoffman, 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.

[111] N. Manwani, P. Sastry, Noise tolerance under risk minimization, IEEE Trans. Cybern. 43 (3) (2013) 1146–1151.

[112] A. Ghosh, N. Manwani, P. Sastry, Making risk minimization tolerant to label noise, Neurocomputing 160 (2015) 93–107.

[113] N. Charoenphakdee, J. Lee, M. Sugiyama, On symmetric losses for learning from corrupted labels, in: International Conference on Machine Learning, 2019.

[114] P.L. Bartlett, M.I. Jordan, J.D. McAuliffe, Convexity, classification, and risk bounds, J. Amer. Statist. Assoc. 101 (473) (2006) 138–156.

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

[116] X. Wang, E. Kodirov, Y. Hua, N.M. Robertson, Improved Mean Absolute Error for Learning Meaningful Patterns from Abnormal Training Data, Tech. Rep., 2019, arXiv:1903.12141v5.

[117] Z. Zhang, 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.

[118] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, J. Bailey, Symmetric cross entropy for robust learning with noisy labels, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 322–330.

[119] N. Natarajan, I.S. Dhillon, P.K. Ravikumar, A. Tewari, Learning with noisy labels, in: Advances in Neural Information Processing Systems, 2013, pp. 1196–1204.

[120] V. Mnih, 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.

[121] Y. Xu, P. Cao, Y. Kong, 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.

[122] G. Patrini, F. Nielsen, R. Nock, M. Carioni, Loss factorization, weakly supervised learning and label noise robustness, in: International Conference on Machine Learning, 2016, pp. 708–717.

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

[124] B. Han, I.W. Tsang, 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.

[125] L.P. Garcia, A.C. de Carvalho, A.C. Lorena, Noise detection in the meta-learning level, Neurocomputing 176 (2016) 14–25.

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

[127] J. Li, Y. Wong, Q. Zhao, M.S. Kankanhalli, Learning to learn from noisy labeled data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5051–5059.

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

[129] N. Kato, T. Li, K. Nishino, Y. Uchida, Improving multi-person pose estimation using label correction, 2018, arXiv:1811.03331.

[130] M. Dehghani, A. Severyn, S. Rothe, J. Kamps, Learning to learn from weak supervision by full supervision, 2017, arXiv:1711.11383, arxiv.org.

[131] M. Dehghani, A. Severyn, S. Rothe, J. Kamps, Avoiding your teacher's mistakes: Training neural networks with controlled weak supervision, 2017, arXiv preprint arXiv:1711.00313.

[132] G. Algan, I. Ulusoy, Meta Soft Label Generation for Noisy Labels, in: Proceedings of the 25th International Conference on Pattern Recognition, ICPR, 2020, pp. 7142–7148.

[133] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

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

[135] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, 2017, arXiv:1710.09412.

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

[137] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, 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.

[138] D. Hendrycks, K. Lee, M. Mazeika, Using pre-training can improve model robustness and uncertainty, 2019, arXiv:1901.09960.

[139] I. Jindal, M. Nokleby, 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.

[140] X. Ma, Y. Wang, M.E. Houle, S. Zhou, S.M. Erfani, S.-T. Xia, S. Wijewickrema, J. Bailey, Dimensionality-driven learning with noisy labels, in: International Conference on Learning Representations, 2018.

[141] S. Azadi, J. Feng, S. Jegelka, T. Darrell, Auxiliary image regularization for deep cnns with noisy labels, in: International Conference on Learning Representations, 2016.

[142] X. Sun, H. Zhou, An empirical comparison of two boosting algorithms on real data sets with artificial class noise, in: Communications in Computer and Information Science, 201 CCIS, (PART 1) 2011, pp. 23–30, http://dx.doi.org/10.1007/978-3-642-22418-8_4.

[143] J. Cao, S. Kwong, R. Wang, A noise-detection based adaboost algorithm for mislabeled data, Pattern Recognit. 45 (12) (2012) 4451–4465.

[144] J. Bootkrajang, A. Kabán, Boosting in the presence of label noise, in: Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference, UAI 2013, 2013, pp. 82–91, arXiv:1309.6818.

[145] Q. Miao, Y. Cao, G. Xia, M. Gong, J. Liu, J. Song, Rboost: label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners, IEEE Trans. Neural Netw. Learn. Syst. 27 (11) (2015) 2216–2228.

[146] B. Sun, S. Chen, J. Wang, H. Chen, A robust multi-class adaboost algorithm for mislabeled noisy data, Knowl.-Based Syst. 102 (2016) 87–102.

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

[148] Y. Kim, J. Yim, J. Yun, J. Kim, Nlnl: Negative learning for noisy labels, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 101–110.

[149] Y. Xia, X. Cao, F. Wen, G. Hua, J. Sun, Learning discriminative reconstructions for unsupervised outlier removal, in: Proceedings of the IEEE International Conference on Computer Vision, 2015 Inter, 2015, pp. 1511–1519, http://dx.doi.org/10.1109/ICCV.2015.177.

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

[151] Y. Duan, O. Wu, Learning with auxiliary less-noisy labels, IEEE Trans. Neural Netw. Learn. Syst. 28 (7) (2017) 1716–1721, http://dx.doi.org/10.1109/TNNLS.2016.2546956.

[152] S. Choi, S. Hong, S. Lim, Choicenet: Robust learning by revealing output correlations, 2018, arXiv:1805.06431.

[153] W. Zhang, Y. Wang, 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.

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

[155] L. Niu, W. Li, 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.

[156] B. Zhuang, L. Liu, Y. Li, C. Shen, 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.

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

[158] A.P. Dawid, A.M. Skene, Maximum likelihood estimation of observer error-rates using the EM algorithm, J. R. Stat. Soc. Ser. C. Appl. Stat. 28 (1) (1979) 20–28.

[159] J.-X. Zhong, N. Li, W. Kong, S. Liu, T.H. Li, G. Li, Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1237–1246.

[160] C. Li, V.S. Sheng, L. Jiang, H. Li, Noise filtering to improve data and model quality for crowdsourcing, Knowl.-Based Syst. 107 (2016) 96–103.

[161] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 41–48.

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

[163] L. Jiang, D. Meng, S.-I. Yu, Z. Lan, S. Shan, A. Hauptmann, Self-paced learning with diversity, in: Advances in Neural Information Processing Systems, 2014, pp. 2078–2086.

[164] H.-P. Kriegel, P. Kröger, E. Schubert, A. Zimek, Loop: local outlier probabilities, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management, ACM, 2009, pp. 1649–1652.

[165] J. Vuurens, A.P. de Vries, 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, CIR'11, 2011, pp. 21–26.

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

[167] P.G. Ipeirotis, F. Provost, J. Wang, Quality management on amazon mechanical turk, in: Proceedings of the ACM SIGKDD Workshop on Human Computation, ACM, 2010, pp. 64–67.

[168] 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.

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

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

[171] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, T. Raiko, Semi-supervised learning with ladder networks, in: Advances in Neural Information Processing Systems, 2015, pp. 3546–3554.

[172] A. Tarvainen, 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.

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

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

[175] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, Mach. Learn. 40 (2) (2000) 139–157.

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

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

[178] A. Torralba, R. Fergus, W.T. Freeman, 80 million tiny images: A large data set for nonparametric object and scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 30 (11) (2008) 1958–1970.

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

[180] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, 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.

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

[182] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (6) (2017) 1452–1464.

[183] T. Kaneko, Y. Ushiku, T. Harada, Label-noise robust generative adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2467–2476.

[184] K.K. Thekumparampil, A. Khetan, Z. Lin, S. Oh, Robustness of conditional GANs to noisy labels, in: Advances in Neural Information Processing Systems, 2018-Decem, 2018, pp. 10271–10282, arXiv:1811.03205.

[185] E. Fonseca, M. Plakal, D.P. Ellis, F. Font, X. Favory, X. Serra, Learning sound event classifiers from web audio with noisy labels, in: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2019-May, 2019, pp. 21–25, http://dx.doi.org/10.1109/ICASSP.2019.8683158, arXiv:1901.01189.

[186] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European Conference on Computer Vision, Springer, 2014, pp. 818–833.

[187] A. Krizhevsky, G. Hinton, et al., Learning Multiple Layers of Features from Tiny Images, Tech. Rep., Citeseer, 2009.

[188] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Ng, Reading Digits in Natural Images with Unsupervised Feature Learning, in: NIPS, 2011.

[189] B. Thomee, D.A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, L.-J. Li, The New Data and New Challenges in Multimedia Research, Vol. 1, (8) 2015, arXiv preprint arXiv:1503.01817.

[190] Y. Lin, Y. Li, C. Wang, J. Chen, Attribute reduction for multi-label learning with fuzzy rough set, Knowl.-Based Syst. 152 (2018) 51–61.

[191] H. Fan, X. Chang, D. Cheng, Y. Yang, D. Xu, A.G. Hauptmann, Complex event detection by identifying reliable shots from untrimmed videos, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 736–744.