Ecological Informatics 57 (2020) 101063

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

Animal species classification using deep neural networks with noise labels

Ahmed Ahmed[a], Hayder Yousif[a], Roland Kays[b,c], Zhihai He[a,*]

[a] Department of Electrical Engineering and Computer Science, University of Missouri, MO 65211, USA
[b] Department of Forestry and Environmental Resources, North Carolina State University, NC 27607, USA
[c] North Carolina Museum of Natural Sciences, Raleigh, NC 27601, USA

ARTICLE INFO

Keywords:
Noisy labels
Deep neural networks
Clean network
K-means clustering

ABSTRACT

In this paper, we developed a robust learning method for animal classification from camera-trap images collected in highly cluttered natural scenes and annotated with noisy labels. We proposed two different network structures with and without clean samples to handle noisy labels. We use k-means clustering to divide the training samples into groups with different characteristics, which are then used to train different networks. These networks with enhanced diversity are then used to jointly predict or correct sample labels using max voting. We evaluate the performance of the proposed method on two public available camera-trap image datasets: Snapshot Serengeti and Panama-Netherlands datasets. Our experimental results demonstrate that our method outperforms the state-of-the-art methods from the literature and achieved improved accuracy on animal species classification from camera-trap images with high levels of label noise.

1. Introduction

Deep neural networks trained with large-scale annotated datasets have achieved remarkable success on various image analysis tasks including image classification (Krizhevsky et al., 2012), attribute learning (Zhang et al., 2014), and scene classification (Zhou et al., 2014). These accomplishments rely on large collections of labeled images, such as the ImageNet dataset (Deng et al., 2009). It is very time-consuming and labor-intensive to collect large datasets with accurate labels. One possible solution is to use large existing datasets that might not be formally labeled but have associated data for automated annotation, for example, using tags from social networks, extracting keywords from search engines, or crowdsourcing to non-professional volunteers (Fergus et al., 2010; Niu et al., 2015). It should be noted that these labels may not be reliable due to incorrect labels. These noisy labels significantly degrade the network learning performance (Nettleton et al., 2010; Rolnick et al., 2017; Sukhbaatar et al., 2014). Therefore, it is important to develop robust learning method for image classification which is able to handle some level of noisy labels.

The problem of learning with noisy labels can be addressed by two major approaches: the first one learns from noisy labels directly which assumes the clean labels are not available. This approach models the label noise that is conditionally independent from the input image (Natarajan et al., 2013; Sukhbaatar et al., 2014). Alternatively, a label cleansing module is used to remove or correct the mislabeled samples (Brodley and Friedl, 1999). The second uses a small set of clean samples with correct labels to guide the network to learn from noisy labels. This approach provides a small subset of clean labels verified by human to improve the classification accuracy. For example, Yuan et al. (2018) used a set of 5000 clean samples to monitor and improve the learning performance of the deep neural network. Veit et al. (2017) used a label cleaning network which is supervised by a small set of clean labels to correct noisy input labels. Clean validation sets have also been used to assign weights to training data (Han et al., 2018; Jiang et al., 2017; Ren et al., 2018).

In this paper, we propose to explore a new approach for learning robust animal classifiers from camera-trap images collected from highly cluttered natural scenes with noisy annotation using an ensemble of convolutional neural networks to jointly clean noisy labels. Specifically, we proposed two different network structures with and without clean samples to handle noisy labels. We use k-means clustering to divide the training samples into groups with different characteristics, which are then used to train networks. These networks with enhanced diversity are then used to jointly predict or correct sample labels using max voting. We evaluate the performance of the proposed method on two publicly available camera-trap image datasets: Snapshot Serengeti and Panama-Netherlands datasets. Our experimental results demonstrate that our method achieved improved accuracy on animal species classification from camera-trap images with high levels of label noise.

Fig. 1 shows an example of training dataset with noisy labels. We

[*] Corresponding author.
E-mail address: hezhi@missouri.edu (Z. He).

https://doi.org/10.1016/j.ecoinf.2020.101063
Received 11 October 2019; Received in revised form 29 January 2020; Accepted 29 January 2020
Available online 31 January 2020

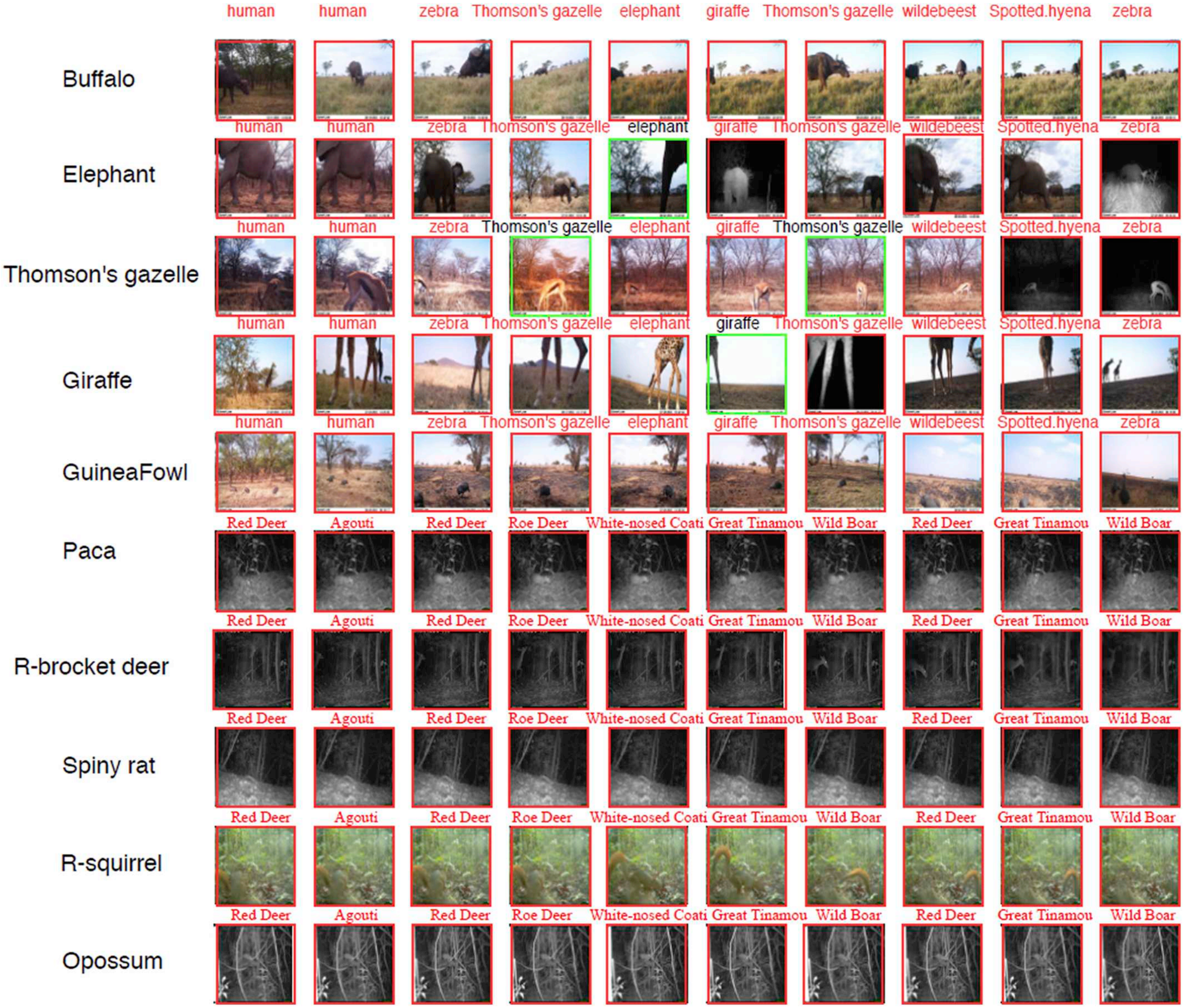


Fig. 1. Samples of camera-trap images training data sets diluted with noisy labels. The red box is the noisy label while the green box is the clean label.

can see that these camera-trap images are very challenging for animal classification with cluttered background, heavy occlusion, and incomplete view of the animals. To simulate noisy labels at different levels, for example, 30–70% of labels being incorrect, we randomly flip the correct label into other labels based on a uniform distribution. For example, in the Snapshot Serengeti dataset, a buffalo is incorrectly labeled as “human” or “zebra” or other species as shown in Fig. 1. Our goal is to learn a robust animal classifier that can be effectively trained with limited clean labels and large numbers of noisy labels. The main contributions of this work can be summarized as:

1) We explored a new approach to learn an accurate and robust network for animal species classification using an ensemble of networks with multi-stage learning for joint label prediction.
2) We proposed to use k-means clustering to split the data based on the similarity of extracted features.
3) We evaluate the performance of proposed method on two public camera-trap datasets and demonstrated improve performance in animal species classification with noisy labels.

The rest of paper is organized as follows. Section 2 reviews several related works to our method. Section 3 presents our proposed method to train deep neural networks with noisy labels with and without clean samples. Section 4 presents the experimental results of our method in two datasets of camera-trap images with three different noise levels. Further discussions and conclusions are provided in Section 5.

2. Related work

Noisy labels in the training data can adversely affect the accuracy of classifiers. To deal with label noise, noise-robust learning algorithms (Beigman and Klebanov, 2009; Manwani and Sastry, 2013; Teng, 2001) and label noise cleansing methods to remove or correct the mislabeled training samples (Barandela and Gasca, 2000; Brodley and Friedl, 1999; Miranda et al., 2009) have been developed. The major challenge lies in how to identify correct training samples from mislabeled samples. Yuan et al. (2018) trained two networks on different datasets to predict sample labels and used the cross prediction between these two networks to clean noisy labels. Sukhbaatar et al. (2014) introduced a noise layer into the base model to match the noise distribution and improve the performance. Jindal et al. (2016) augmented a deep network with a softmax layer that models the label noise distribution. Xiao et al. (2015) proposed a conditional noise model that predicts the type of noise affecting training samples and then attempted to remove them. They used two CNNs to predict the class label and noise type. A small set of clean samples is used to pre-train or fine-tune the model. Natarajan et al. (2013) studied noisy data in binary classification using two approaches to reduce the impact of label noise: unbiased estimation and a weighted surrogate loss.

Another approach to learn from noisy labeled data is to combine noisy labels with a small set of clean labels. A semi-supervised approach is developed to learn from both labeled and unlabeled data (Zhu and Goldberg, 2009). Kingma et al. (2014) developed deep generative models for semi-supervised learning to improve the performance. Veit et al. (2017) provided a small set of clean labels to train a label cleaning network. They proposed to learn the mapping between noisy and clean labels and then use the mapping for deep neural network training. The label propagation method was proposed by (Zhu and Ghahramani, 2002) to propagate labels from the labeled dataset to unlabeled data. A graph-based label propagation was proposed by Fergus et al. (2009).

It should be noted that deep networks are robust to noise labels to a certain degree. Szegedy et al. (2013) demonstrated the robustness of neural networks to adversarial samples. Rolnick et al. (2017) show the ability of deep neural network to learn from massive noisy labels. They have investigated the effect of batch size and learning rate on model performance. Van Horn et al. (2015) found that learning algorithms based on CNN features and part localization are robust to annotation errors and training data corruption if the error rate is not too high. In this work, we follow the basic idea of Yuan et al. (2018) using multiple networks to estimate the correct the label of the input sample. We recognize that, during this joint estimation process, it is very important to ensure the diversity between these networks so that different networks can provide different perspectives or information about the input sample. In this way, the performance of joint estimation can be improved.

3. Our proposed method

In this section, we present our method to train deep neural networks with noisy labels for robust animal classification.

3.1. Noisy labels

Suppose that we have a set of n images, i.e., $D = \{(x_1,y_1),(x_2,y_2), \ldots,(x_n,y_n)\}$,where x_i is the $i - th$ image and $y_i \epsilon \{1,2,\ldots,L\}$ is the correct label for x_i, and L is the total number of classes. To simulate noisy labels, we randomly flip the labels of a subset of the training set to wrong labels. The relative size of this subset of noisy labels, also called noise level, ranges from 30% to 70%. We denote the noisy labeled data set by $D^* = \{(x_1,{y_1}^*),(x_2,{y_2}^*),\ldots,(x_n,{y_n}^*)\}$, where ${y_i}^* \epsilon \{1,2,\ldots,L\}$ is the noisy label with $y_i \neq {y_i}^*$. Additionally, we have access to a small set of clean labeled data, for example, those being reviewed and confirmed by experts.

3.2. Method overview

Our goal is to train our network with different noise levels with and without a small subset of clean labels. The main framework of the proposed method is illustrated in Fig. 2. As shown in Fig. 2(a), we use the pretrained clean network to encode the input image into a feature vector. In the alternative approach (Fig. 2b), we assume that the clean data is not available. We use the pretrained base model with different levels of noise to transform the input training image into feature maps. In both structures, we apply k-means clustering (Everitt et al., 2011) to partition the training dataset for each class into 12 clusters, and then randomly choose 6 clusters to train independent convolutional neural networks.

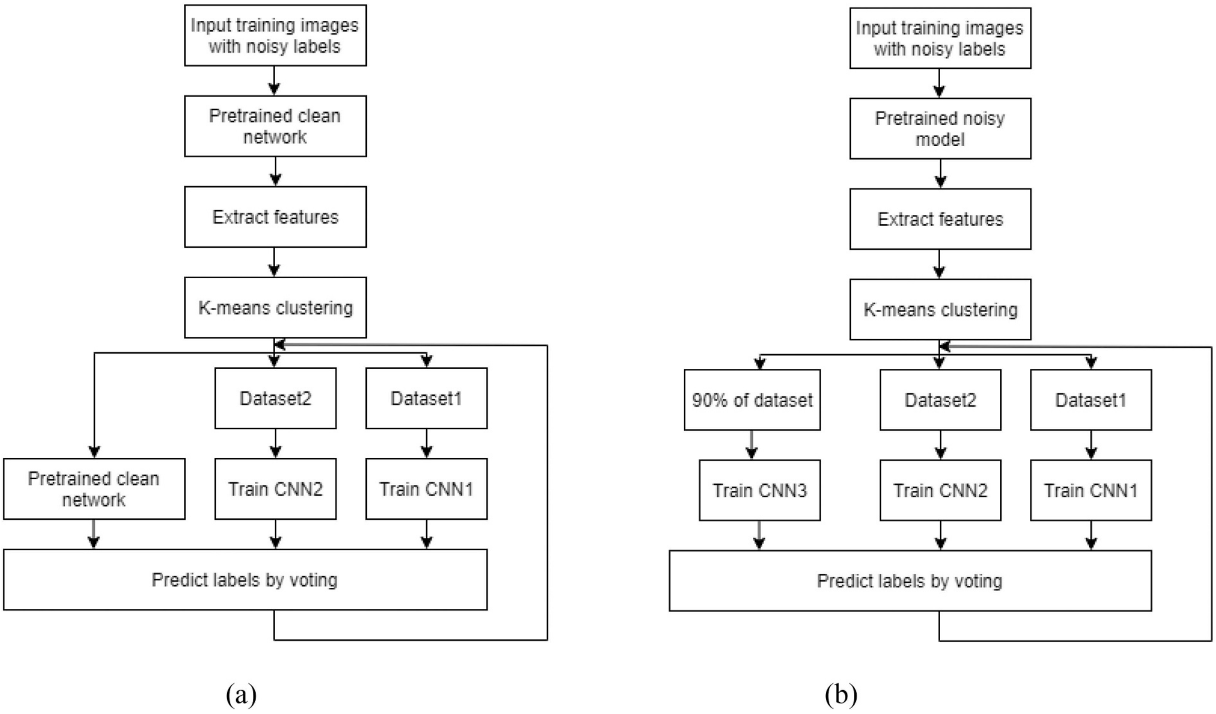


Fig. 2. Flow diagram of the proposed method. (a) With clean network. (b) Without clean network.

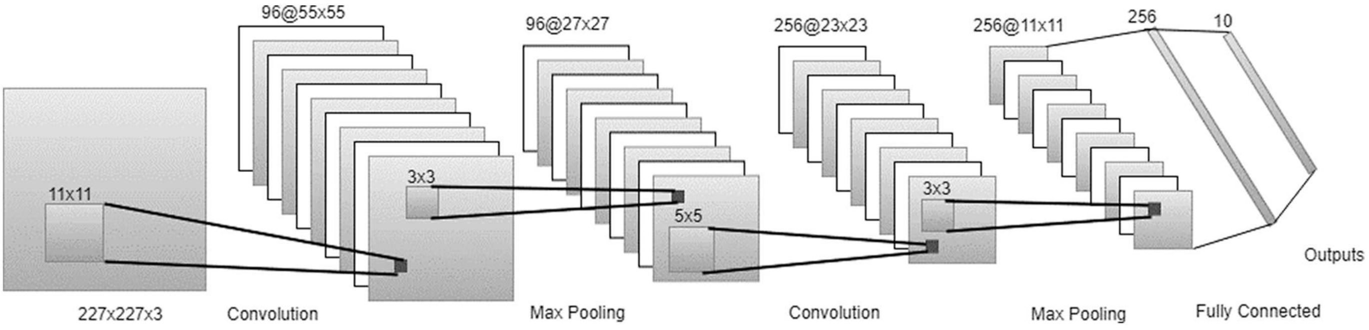


Fig. 3. The architecture of the convolutional neural network.

From Fig. 2(a), we can see the pretrained clean network, which is only trained once on the clean samples, can be used only to predict the label for each stage without training on the dataset resulted from k-means clustering and help the other two networks to clean their labels. In contrast, the CNN3 network in Fig. 2(b) can be trained on different percentages of training dataset. We randomly sample 90% of the whole training dataset to train the CNN3. This network, along with the other two independent networks, will be used to predict new labels. The predicted labels will be used to update the network in the next stage. This iterative process will be repeated multiple times. We set the total number of stages to be 5.

In both structures, we use each network individually to predict the label for each image of the training dataset. We use max voting among the predicted labels to update the original image label. Specifically, if two or more networks produce the same label, we then assign this label to the sample. Otherwise, we just use the original label, which could be noisy. Once the labels are updated, we then come back to refine those two networks. This label update and network refinement process are repeated for multiple stages. Detailed algorithms to implement these two approaches are outlined in Algorithm 1 and Algorithm 2. See Appendix A for details.

Fig. 3 shows the network structure for the base network, the clean network, and the independent networks we used for the Serengeti dataset. We use the same model architecture as in Yuan et al. (2018) for fair performance comparison. Each network has two convolutional layers with 96 feature maps of 11 × 11 filter size and 256 feature maps of 5 × 5 filter size respectively, and each convolutional layer is followed by rectified linear units (ReLUs). Local response normalization (LRN) layers are used to normalize the unbounded activations produced by ReLU, and 3 × 3 max pooling layer to find the largest response in the local neighborhood. We choose this model due to its low complexity for high-speed analysis of massive camera-trap images. We train the model using a stochastic gradian decent with momentum 0.9, learning rate 1×10^{-3} and batch size 64.

As discussed in the above, we use k-means clustering to partition the feature maps of the first fully connected layer with high-level feature values followed by rectified linear unit into k clusters using the CNN features. We choose $k = 12$. The feature maps represent the result of applying the filters to the training data where each layer produces different feature maps. Instead of using CNN classification results to predict the noisy label, we propose to cluster the fully connect layer features into clusters which shows a significant improvement over the supervised method. We recognize that this approach is more effective than randomly splitting the training set into two groups since these two groups obtained by the clustering approach will have much more different image characteristics than those two groups obtained by the random splitting approach.

Table 1
The partition of 10 species of the Snapshot Serengeti dataset.

Species	Without clean samples		With clean samples		
	Training set	Test set	Training set	Test set	Clean Set
Buffalo	966	242	1027	121	60
Elephant	458	115	487	57	29
Thomson's gazelle	1254	314	1333	157	78
Giraffe	764	191	812	95	48
Guineafowl	1046	262	1112	131	65
Hartebeest	606	152	644	76	38
Human	1520	380	1615	190	95
Spotted hyena	484	121	514	61	30
Wildebeest	1155	289	1227	145	72
Zebra	2068	517	2197	259	129
Total	10,321	2583	10,968	1292	644

Table 2
The partition of 20 species of the Panama-Netherlands dataset.

Species	Without clean samples		With clean samples		
	Training set	Test set	Training set	Test set	Clean set
Agouti	140	97	140	83	14
Collared peccary	385	119	385	100	19
Paca	471	106	471	87	19
Red brocket deer	447	140	447	115	25
White-nosed Coati	571	160	571	134	26
Spiny rat	230	85	230	69	16
Ocelot	289	45	289	34	11
Red squirrel	237	66	237	53	13
Common opossum	297	108	297	90	18
Birds	353	112	353	99	13
Great tinamou	767	83	767	64	19
White-tailed deer	964	533	964	492	41
Mouflon	718	519	718	470	49
Red deer	1126	885	1126	837	48
Roe deer	498	135	498	110	25
Wild boar	891	137	891	105	32
Red fox	242	77	242	60	17
European hare	288	98	288	83	15
Wood mouse	688	154	688	112	42
Coiban agouti	240	58	240	49	9
Total	9842	3717	9842	3246	471

4. Experimental results

4.1. Datasets

We use two datasets of camera-trap images for performance evaluations. The first one is the Snapshot Serengeti dataset with camera-trap images collected from the Serengeti National Park, Tanzania

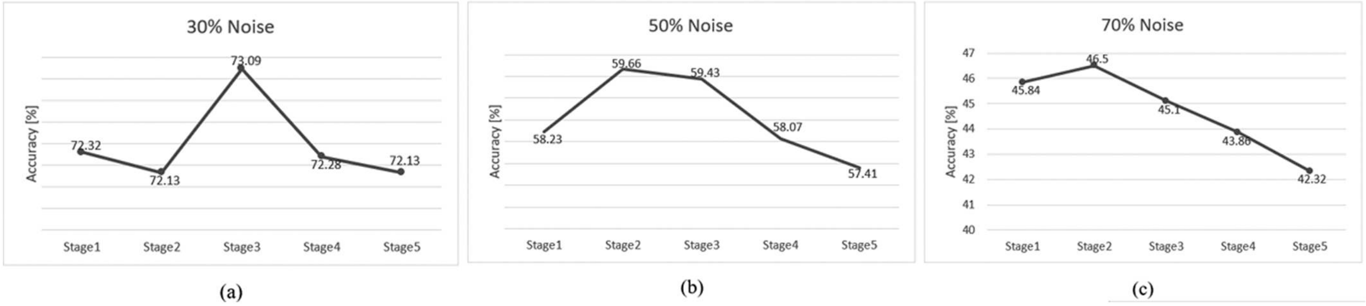


Fig. 4. Accuracy of five stages on 10 species of Serengeti dataset in different levels of noise without clean samples.

(Swanson et al., 2015). We consider image-level animal classification, assuming that one image only has one animal species. The dataset consists of 12,904 color images with 10 animal species. We randomly split our dataset into 80% samples for training and 20% for test. When the clean network is used, we use 85% samples for training, 10% for test, and the rest 5% samples are used as clean samples. Table 1 summarizes the number of images used in the training and testing sets for each animal species. The second one is the Panama-Netherlands dataset (Zhang et al., 2016) with image sequences of 20 animal species. We randomly sampled the training, testing and cleaning images as shown in

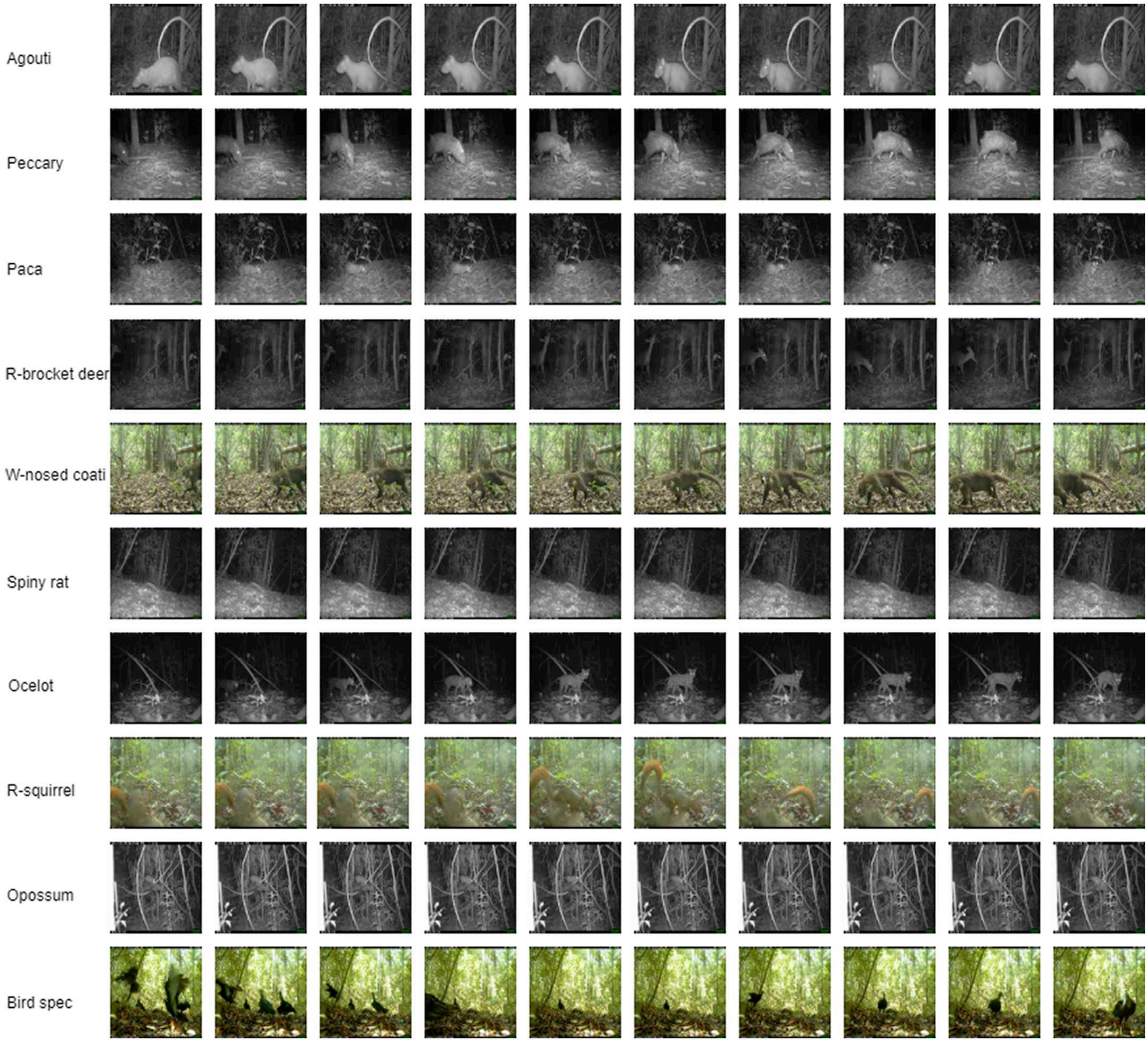


Fig. 5. Samples of 10 species from the Panama-Netherlands camera-trap dataset. Each sample corresponds to a sequence of images, often in the range 3–20 images.

Table 3
Performance (classification accuracy) comparison on 10 species of Snapshot Serengeti dataset at different levels of noise without clean samples.

Methods	Label noise levels		
	30%	50%	70%
Base CNN	69.26%	56.91%	44.48%
ICL (Yuan et al., 2018)	71.62%	58.46%	46.38%
Our method	73.09%	59.66%	46.61%

Table 4
Performance (classification accuracy) comparison on 20 species of Panama-Netherlands dataset at different levels of noise without clean network.

Method	Noise levels		
	30%	50%	70%
Base CNN	45.71	41.83	39.84
ICL (Yuan et al. (2018))	46.76	43.07	42.24
Our method	48.94	45.60	43.64

Table 2. During network training, data augmentation methods, such as random image flipping, translation, and center crop, are applied to improve the classification accuracy.

4.2. Results

In this section, we evaluate the performance of our method on the Snapshot Serengeti and Panama-Netherlands datasets with three different levels of noising labels: 30%, 50%, and 70%. We compare the result of our proposed method with the baseline CNN and Yuan et al. (2018).

4.2.1. Results on the Snapshot Serengeti and Panama-Netherlands datasets without clean samples

We first test our method without the clean network on 10 species of Snapshot Serengeti dataset using 10,321 images for training and 2583 for testing. Fig. 4 shows how the classification accuracy obtained at each stage for those three different label noise levels. For 30%, our method achieves accuracy of 73.09% at stage 3 and the accuracy drops after this stage. For noise levels of 50% and 70%, it achieves accuracy of 59.66% and 46.5% respectively at stage 2 and the accuracy drops after stage 2. We can see that multiple-stage of label update and network refinement can improve the accuracy. But, when the number of stages is too large, for example, larger than 3, the continuous label update becomes less and less accurate, which degrades the overall performance.

We also evaluate our method on the Panama-Netherlands dataset which has 20 animal species commonly seen in North America. Fig. 5 shows samples of 10 species from the Panama-Netherlands dataset. Each sample corresponds to a sequence of images. This is because, once triggered by animal motion, the camera-trap will take a sequence of images, often in the range of 3–20 images at a frame rate of one frame per second, depending how long the animal stays in the camera view. We use the pretrained AlexNet convolutional neural network to initialize our networks.

We compare the performance our method against the ICL method, a state-of-the-art method for removing label noise. We also include the baseline CNN method without any noise cleaning for comparison. From Tables 3 and 4, we can see our method outperforms the base CNN and ICL (Yuan et al., 2018) at all three levels of noise on the Snapshot Serengeti and Panama-Netherlands datasets.

We show the effect of sampling different percentages of whole data on the performance. It is clear that selecting 90% of whole data to train the network achieves high accuracy on the Snapshot Serengeti as shown In Table 5. In Table 6, we achieve high accuracy of sampling 70% of all Panama-Netherlands dataset to train the network.

Table 5
Classification accuracy with different percentages of training sample used for the base network on the Snapshot Serengeti dataset at different levels of noise without clean samples.

Percentage	Noise levels		
	30%	50%	70%
50%	72.36%	58.46%	45.03%
70%	71.78%	58.85%	45.99%
90%	73.09%	59.66%	46.61%
100%	72.09%	58.73%	46.38%

Table 6
Classification accuracy with different percentages of training sample used for the base network on the Panama-Netherlands dataset at different levels of noise without clean samples.

Percentage	Noise levels		
	30%	50%	70%
50%	48.16	44.04	43.05
70%	48.94	45.60	43.64
90%	47.54	43.91	42.45
100%	48.10	44.09	39.36

4.2.2. Results on the Snapshot Serengeti and Panama-Netherlands datasets with clean samples

In the following experiments, we assume 5% of clean samples are available for training the clean network. We use accuracy, precision, recall, and F1 score metrics to evaluate the effectiveness of the proposed method. The accuracy is defined as

$$Accuracy\ (ACC) = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

where TP represents the total number of true positive samples, TN represents the total number of true negative samples, FP represents the total number of false positive samples, and FN represents the total number of false negative samples. Precision is defined according to the following equation

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Conversely, recall or sensitivity can be computed as

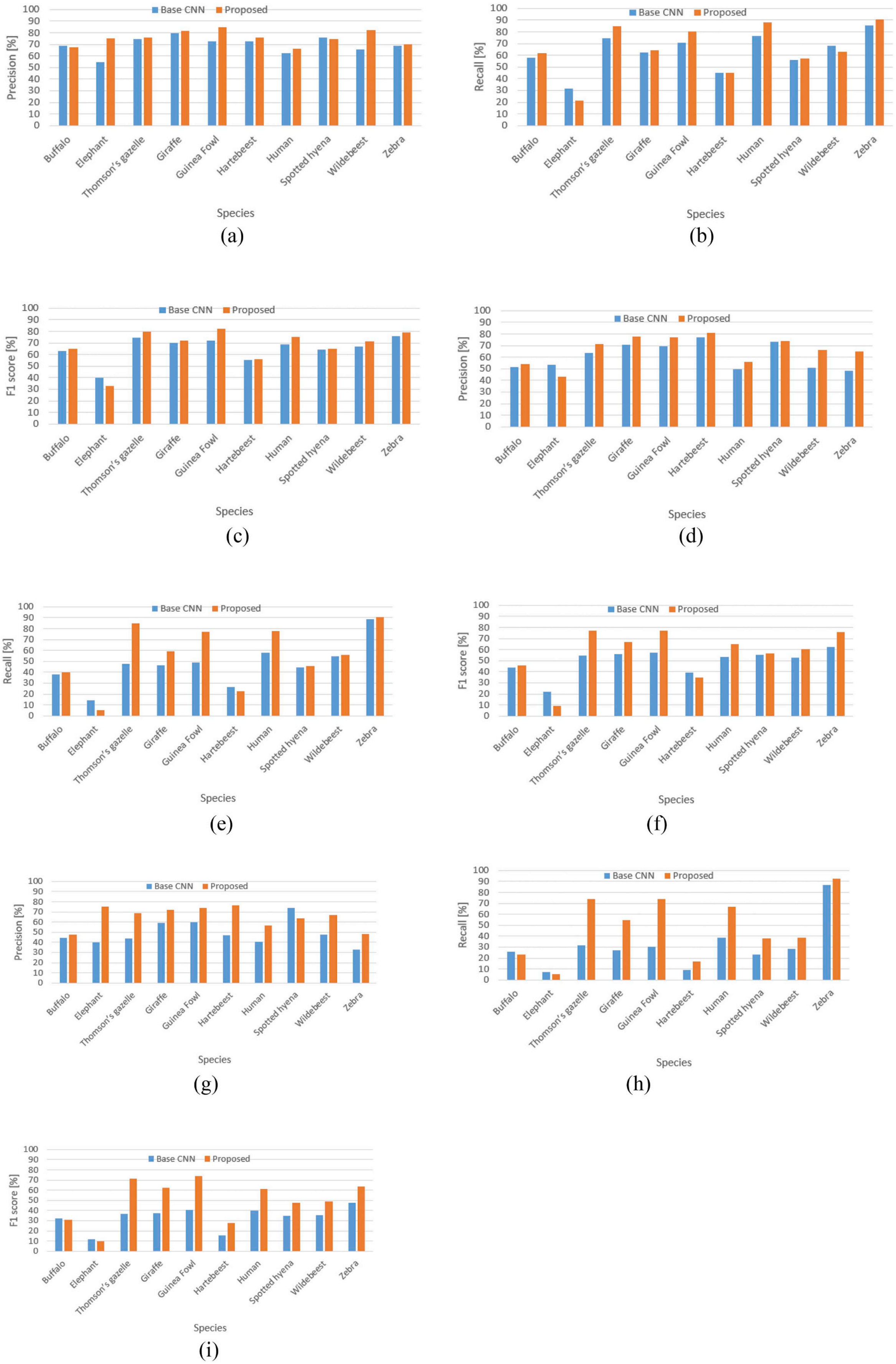
$$Recall = \frac{TP}{TP + FN} \tag{3}$$

To calculate the combination of precision and recall, we use F1 score or F measure which represents a harmonic mean of precision and recall. It is defined as

$$F1\ score = \frac{2 \times precision \times recall}{precision + recall} = \frac{2TP}{2TP + FP + FN} \tag{4}$$

For each animal species, we compare the classification precision, recall, and F1 score metrics of our proposed method with the base CNN as shown in Fig. 6. We can see that the proposed method on the Snapshot Serengeti is able to correct noisy labels for 30%, 50%, and 70% levels of noise. These results are summarized in Table 7. For Panama-Netherlands dataset, we also compare the average precision, average recall, and average F1 score metrics of our proposed method with the base CNN as shown in Table 8. Our method outperforms the base CNN at all three levels of noise.

In Table 9, we compare our method with the state-of-the-art method by Yuan et al., 2018. For label noise levels of 30%, 50%, and 70%, our method is able to improve the accuracy over the base CNN method by 4.41%, 11.38%, and 18.81%, respectively. Compared to Yuan's method, it has increased the classification accuracy by 3.33%, 7.09%, and 14.34%,



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Fig. 6. Precision (a), Recall (b), and F1 score (c) values for each species with 30% noise level. Precision (d), Recall (e), and F1 score (f) values for each species with 50% noise level. Precision (g), Recall (h), and F1 score (i) values for each species with 70% noise level on 10 of Snapshot Serengeti dataset.

Table 7
Average precision, average recall, and average F1 score values with different proportions of noise on 10 species of Snapshot Serengeti dataset.

Method	Average precision			Average recall			Average F1 score		
	30%	50%	70%	30%	50%	70%	30%	50%	70%
Base CNN	69.44	60.80	48.83	62.74	46.68	30.86	64.99	49.70	33.24
Our method	75.22	66.50	64.98	65.52	55.85	48.40	67.80	56.99	49.78

Table 8
Average precision, average recall, and average F1 score values with different proportions of noise on 20 species of Panama-Netherlands dataset.

Method	Average precision			Average recall			Average F1 score		
	30%	50%	70%	30%	50%	70%	30%	50%	70%
Base CNN	49.26	40.60	31.17	48.39	38.61	33.72	46.40	37.71	30.17
Our method	59.71	59.08	62.16	57.29	56.13	54.38	54.55	51.69	50.25

Table 9
Performance (classification accuracy) comparison on 10 species of Snapshot Serengeti dataset at different levels of noise with clean network.

Method	Noise levels		
	30%	50%	70%
Base CNN	68.89	54.41	39.63
ICL (Yuan et al. (2018))	69.97	58.70	44.10
Our method	73.30	65.79	58.44

Table 10
Performance (classification accuracy) comparison on 20 species of Panama-Netherlands dataset at different levels of noise with clean network.

Method	Noise levels		
	30%	50%	70%
Base CNN	48.64	41.99	40.33
ICL (Yuan et al. (2018))	52.19	44.37	42.89
Our method	58.93	55.14	54.47

Table 11
Performance (classification accuracy) evaluation with different numbers of groups on the Snapshot Serengeti dataset at different levels of noise with clean samples.

Number of groups	Clusters per group	Noise levels		
		30%	50%	70%
2	6	73.30	65.79	58.44
2	4	72.83	65.40	58.36
3	4	72.91	65.40	57.12
4	3	71.21	64.40	56.42
5	2	70.36	63.62	56.19

Table 12
Performance (classification accuracy) evaluation with different numbers of groups on the Panama-Netherlands dataset at different levels of noise with clean samples.

Number of groups	Clusters per group	Noise Levels		
		30%	50%	70%
2	6	58.10	55.14	53.64
2	4	58.07	53.73	51.73
3	4	57.86	54.68	54.13
4	3	58.93	54.65	54.47
5	2	58.44	54.04	52.03

respectively.

We also test our method on the Panama-Netherlands dataset. In Table 10, we can see that our method outperforms the ICL method by a large margin. For label noise levels of 30%, 50%, and 70%, we can see that our method outperforms the base CNN by 10.29%, 13.15%, and 14.14% respectively and by 6.74%, 10.77%, and 11.58% respectively compared to Yuan's method.

We note that the number of clusters and the number of networks play an important role in our algorithm. We use two groups with 6 clusters for each to train two networks. In Table 11, we evaluate the performance of our method on the Snapshot Serengeti with different number of groups. We can see that the best performance is achieved with two groups.

In Table 12, We show the classification accuracy of our method on Panama-Netherlands dataset with different number of groups. For label noise levels of 30% and 70%, We achieve high accuracy with 4 groups and 12 clusters, 3 clusters for each group. For 50% noise level, we achieve high accuracy with 2 groups and 12 clusters, 6 clusters per each group.

5. Conclusions and further discussions

We have studied the effect of noisy labels on animal classification. We have developed a new method to learn accurate an animal species classification network from these noisy samples. We considered the network training process with and without clean samples. The experimental results demonstrate the robustness of our method to label noise with and without clean samples. In this paper, we have recognized that the diversity of the networks is important for achieving improved performance of joint estimation of sample labels. We used k-means clustering with deep neural network features to create clusters with different characteristics. We then partition these clusters into groups. Each group is then used to train a separate network. In this way, we have made sure that each network is trained with different set of images. With max voting, this allows us to predict the true label of the noisy samples.

The proposed method for animal species classification from camera-trap images with noisy labels has important application in citizen science-based large-scale wildlife monitoring (Fegraus et al., 2019). The massive number of camera-trap images are collected, annotated, and uploaded by non-professional volunteers or citizen scientists. Their annotations will inevitably contain a significant amount of incorrect labels. The proposed method will allow us to learn efficient animal species classifiers from these datasets.

Acknowledgements

This work has been supported in part by National Science Foundation under grant 1539389.

Appendix A. Algorithm 1 and Algorithm 2 to implement two approaches with and without clean samples

Algorithm 1 Training Models with Clean Samples

```
Input: Input training data $D$ with noisy labels and Pretrained clean network
Output: Learned two independent networks
set $n = 1$ , $N = 5$, and initialize two independent networks parameters
Extract feature maps using pretrained clean network
Cluster training dataset into $G1$ and $G2$ groups by k-means clustering
While $n < N$
    C_net1= $G1$ trained on net1
    C_net2= $G2$ trained on net2
    Lp1=labels predicted for $D$ by C_net1
    Lp2=labels predicted for $D$ by C_net2
    Lp3=labels predicted for $D$ by C_clean
    for each $x_i$ in $D$ do
        if $x_i's$ label is equal in Lp1 and Lp2 or $x_i's$ label is equal in Lp1 and Lp3 then
            update $x_i's$ label to Lp1
        elsif $x_i's$ label is equal in Lp2 and Lp3 then
            update $x_i's$ label to Lp2
        else keep the same label without change
        end if
    end for
    set n=n+1
end while
return learned two independent networks
```

Algorithm 2 Training Models without Clean Samples

```
Input: Input training data $D$ with noisy labels and Pretrained noisy model
Output: Learned three independent networks
set $n = 1$ , $N = 5$, and initialize three networks parameters
Extract feature maps using pretrained noisy model
Cluster training dataset into $G1$ and $G2$ groups by k-means clustering and select 90% of G1G2
While $n < N$
    C_net1= $G1$ trained on net1
    C_net2= $G2$ trained on net2
    C_net3=90% of G1G2 trained on net3
    Lp1=labels predicted for $D$ by C_net1
    Lp2=labels predicted for $D$ by C_net2
    Lp3=labels predicted for $D$ by C_net3
    for each $x_i$ in $D$ do
        if $x_i's$ label is equal in Lp1 and Lp2 or $x_i's$ label is equal in Lp1 and Lp3 then
            update $x_i's$ label to Lp1
        elsif $x_i's$ label is equal in Lp2 and Lp3 then
            update $x_i's$ label to Lp2
        else keep the same label without change
        end if
    end for
    set n=n+1
end while
return learned three independent networks
```

References

Barandela, R., Gasca, E., 2000, August. Decontamination of training samples for supervised pattern recognition methods. In: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Springer, Berlin, Heidelberg, pp. 621–630.

Beigman, E., Klebanov, B.B., 2009. Learning with annotation noise. In: In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. Vol 1. Association for Computational Linguistics, pp. 280–287 August.

Brodley, C.E., Friedl, M.A., 1999. Identifying mislabeled training data. J. Artif. Intell. Res. 11, 131–167.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: In Computer Vision and Pattern Recognition, CVPR 2009. IEEE Conference on (pp. 248–255). Ieee.

Everitt, B.S., Landau, S., Leese, M., Stahl, D., 2011. An Introduction to Classification and Clustering, in Cluster Analysis, 5th edition. John Wiley & Sons, Ltd, Chichester, UK.

Fegraus, A.J.A., Birch, E.T., Flores, N., Kays, R., Brien, T.G.O., Palmer, J., Schuttler, S., Zhao, J.Y., Jetz, W., Kinnaird, M., Kulkarni, S., Lyet, A., Thau, D., 2019. Wildlife Insights : A Platform to Maximize the Potential of Camera Trap and Other Passive Sensor Wildlife Data for the Planet. Environ Conserv page 2014. https://doi.org/10.1017/S0376892919000298.

Fergus, R., Weiss, Y., Torralba, A., 2009. Semi-supervised learning in gigantic image collections. Adv. Neural Inf. Proces. Syst. 522–530.

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

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M., 2018. Co-teaching: robust training of deep neural networks with extremely noisy labels. Adv. Neural Inf. Proces. Syst. 8527–8537.

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

Jindal, I., Nokleby, M., Chen, X., 2016, December. Learning deep networks from noisy labels with dropout regularization. In Data Mining (ICDM). In: 2016 IEEE 16th International Conference on (pp. 967–972), ieee.

Kingma, D.P., Mohamed, S., Rezende, D.J., Welling, M., 2014. Semi-supervised learning with deep generative models. Adv. Neural Inf. Proces. Syst. 3581–3589.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Proces. Syst. 1097–1105.

Manwani, N., Sastry, P.S., 2013. Noise tolerance under risk minimization. IEEE Trans. Cybernetics 43 (3), 1146–1151.

Miranda, A.L., Garcia, L.P.F., Carvalho, A.C., Lorena, A.C., 2009, June. Use of classification algorithms in noise detection and elimination. In: International Conference on Hybrid Artificial Intelligence Systems. Springer, Berlin, Heidelberg, pp. 417–424.

Natarajan, N., Dhillon, I.S., Ravikumar, P.K., Tewari, A., 2013. Learning with noisy labels. Adv. Neural Inf. Proces. Syst. 1196–1204.

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

Niu, L., Li, W., Xu, D., 2015. Visual recognition by learning from web data: a weakly supervised domain generalization approach. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2774–2783.

Ren, M., Zeng, W., Yang, B., Urtasun, R., 2018. Learning to reweight examples for robust deep learning. arXiv preprint arXiv:1803.09050.

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

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

Swanson, A., Kosmala, M., Lintott, C., Simpson, R., Smith, A., Packer, C., 2015. Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. Sci. data 2.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R., 2013. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

Teng, C.M., 2001, May. A comparison of noise handling techniques. In: In FLAIRS Conference (pp. 269–273).

Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., Belongie, S., 2015. Building a bird recognition app and large scale dataset with citizen scientists: the fine print in fine-grained dataset collection. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 595–604.

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., Belongie, S.J., 2017, July. Learning from noisy large-scale datasets with minimal supervision. In: In CVPR, pp. 6575–6583.

Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X., 2015. Learning from massive noisy labeled data for image classification. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2691–2699.

Yuan, B., Chen, J., Zhang, W., Tai, H.S., McMains, S., 2018, March. Iterative cross learning on noisy labels. In: In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 757–765 IEEE.

Zhang, N., Paluri, M., Ranzato, M.A., Darrell, T., Bourdev, L., 2014. Panda: pose aligned networks for deep attribute modeling. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 1637–1644.

Zhang, Z., He, Z., Cao, G., Cao, W., 2016. Animal detection from highly cluttered natural scenes using spatiotemporal object region proposals and patch verification. IEEE Trans. Multimedia 18 (10), 2079–2092.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A., 2014. Learning deep features for scene recognition using places database. Adv. Neural Inf. Proces. Syst. 487–495.

Zhu, X., Ghahramani, Z., 2002. Learning from Labeled and Unlabeled Data with Label Propagation.

Zhu, X., Goldberg, A.B., 2009. Introduction to semi-supervised learning. Synth. Lect. Artif. Intell. Mach. Learn. 3 (1), 1–130.