

Short Papers

Learning From Large-Scale Noisy Web Data With Ubiquitous Reweighting for Image Classification

Jia Li¹, Senior Member, IEEE, Yafei Song, Jianfeng Zhu, Lele Cheng, Ying Su, Lin Ye, Pengcheng Yuan², and Shumin Han

Abstract—Many important advances of deep learning techniques have originated from the efforts of addressing the image classification task on large-scale datasets. However, the construction of clean datasets is costly and time-consuming since the Internet is overwhelmed by noisy images with inadequate and inaccurate tags. In this paper, we propose a Ubiquitous Reweighting Network (URNNet) that can learn an image classification model from noisy web data. By observing the web data, we find that there are five key challenges, i.e., imbalanced class sizes, high intra-classes diversity and inter-class similarity, imprecise instances, insufficient representative instances, and ambiguous class labels. With these challenges in mind, we assume every training instance has the potential to contribute positively by alleviating the data bias and noise via reweighting the influence of each instance according to different class sizes, large instance clusters, its confidence, small instance bags, and the labels. In this manner, the influence of bias and noise in the data can be gradually alleviated, leading to the steadily improving performance of URNNet. Experimental results in the WebVision 2018 challenge with 16 million noisy training images from 5000 classes show that our approach outperforms state-of-the-art models and ranks first place in the image classification task.

Index Terms—Image classification, noisy web data, CNNs, ubiquitous reweighting, deep learning

1 INTRODUCTION

In the past decade, the deep learning technique has achieved great advances under the assistance of large-scale visual datasets. Among these advances, many of highly influential innovations originate from the efforts in addressing the image classification task on ImageNet [1], such as the multi-layer convolution structure of AlexNet [2], very deep architecture of VGGNet [3], multi-scale inception of GoogLeNet [4], residual connections of ResNet [5], group convolution in ResNeXt [6] and squeeze-and-excitation module in SENet [7]. Based on these innovations in image classification, the deep learning technique has also obtained remarkable success in many other

applications such as autonomous driving [8], robotic control [9], medical imaging diagnostics [10] and game playing [11].

With the usage of these deep learning techniques, state-of-the-art models have achieved impressive high accuracy in the image classification task on ImageNet. He *et al.* [12] proposed a model whose performance has outperformed human. The improvement subsequently becomes marginal in recent years, which has led to the termination of this challenge from 2017. However, quite a few researchers still think that the image classification problem has not been completely solved, since ImageNet is a clean and well annotated dataset, while the data in the real world are usually biased and noisy. Human can learn exact knowledge from such data, however, state-of-the-art methods may lack this ability. To this end, researchers are now looking urgently at the datasets which are more similar to real-world data. To this end, Li *et al.* [13] have constructed and published the WebVision dataset, whose 2.0 version consists of 16 million images within five thousand classes. For each class, the images are retrieved from Google and Flickr using specific retrieval words which are also taken as the label of this class. By means of modern search engine, such large-scale image dataset can be constructed quickly as well as with low cost. However, the retrieved results are not always accurate, the algorithms as well as the corresponding models will face some key challenges to exploit such a dataset. To accelerate the research progress on these challenges, Li *et al.* [13] have organized the WebVision Image Classification Challenge since 2017.

To effectively exploit the large-scale noisy dataset, current works usually follow two types of methodologies. Some works focus on purifying the noisy dataset. For example, Veit *et al.* [14] resorted to a clean sub-set and the multi-task learning algorithm to jointly clean noisy annotations and accurately classify images. Wang *et al.* [15] proposed an iterative learning framework to detect the noisy samples with open-set labels so as to pull away noisy samples from clean samples. Tanaka *et al.* [16] proposed a joint optimization framework to update the parameters and estimate true labels. These methods aim to learn the accurate model as well as clean the dataset. However, it is a more difficult task to correct the label of each sample, which also may be not necessary.

The other works focus on decreasing the weight of noisy samples so as to alleviate their negative effects. For example, Fu *et al.* [17] assumed that the features of true positive samples are similar and the features of noisy samples are variant, which is also an important achievement for deep learning methods to tackle the selectivity-invariance dilemma [18]. Based on this assumption, they minimized the discrepancy between the affinity representation and its low rank approximation to decrease the influence of noisy samples. Zhuang *et al.* [19] randomly grouped multiple images into a single training instance and adopted attention strategy to suppress the noisy signal. Niu *et al.* [20] adopted more accessible category-level supervision. In particular, they built their deep probabilistic framework upon variational autoencoder (VAE), in which the classification network and VAE can jointly leverage category-level hybrid information. These methods focus on reweighting the noisy samples, however, both noise and bias are ubiquitous.

With previous works in mind, we carefully analyze the WebVision dataset 2.0, and find that there may be five primary difficulties during exploiting the large-scale noisy web data. First, the size of each class varies from about three hundred to fifty thousand, which leads to imbalanced class sizes. Second, as some classes are really similar to each other, these classes usually have high intra-classes diversity and inter-class similarity. Third, since the

- J. Li is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, and the Peng Cheng Laboratory, Shenzhen 518055, China. E-mail: jiali@buaa.edu.cn.
- Y. Song is with the National Engineering Laboratory for Video Technology, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China. E-mail: songyf@pku.edu.cn.
- J. Zhu, L. Cheng, Y. Su, L. Ye, and S. Han are with the Computer Vision Technology Department of Baidu, Beijing 100871, China. E-mail: {zhujianfeng03, chenglele, suyings02, yelin02, hanshumin}@baidu.com.
- P. Yuan is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the Computer Vision Technology Department of Baidu, Beijing 100871, China. E-mail: yuanpengcheng@buaa.edu.cn.

Manuscript received 29 Oct. 2018; revised 22 July 2019; accepted 15 Dec. 2019. Date of publication 24 Dec. 2019; date of current version 1 Apr. 2021.

(Corresponding authors: Shumin Han and Jia Li.)

Recommended for acceptance by A. Gupta.

Digital Object Identifier no. 10.1109/TPAMI.2019.2961910

instances are directly retrieved from Internet, quite a few instances are with imprecise or wrong labels. Fourth, the common instances may be overwhelmed by uncommon instances, therefore representative instances are insufficient. Last but not least, due to the ambiguous class labels, some instances may actually belong to several classes at the same time.

To address these challenges, we assume that every instance has the potential to contribute more by suppressing the noise via reweighting it according different classes, instance clusters, individual instance, small instance bags, and their labels. In this manner, the influence of noise in the web data can be gradually alleviated, leading to the steadily improving performances of the proposed URNet model. Experimental results in the WebVision 2018 challenge with 16 million noisy training images from 5,000 classes show that our approach outperforms state-of-the-art models and ranks the first place in the image classification task. The contributions of this paper mainly lie in three aspects:

- 1) We thoroughly analyze the challenges while exploiting the large-scale noisy web data and summarize five potential causes, which may inspire further studies.
- 2) We propose a ubiquitous reweighting network as well as five reweighting strategies which are adopted to alleviate the imbalanced class sizes, high intra-classes diversity and inter-class similarity, imprecise instance labels, insufficient representative instances, and ambiguous class labels.
- 3) With the proposed URNet, we can exploit the large-scale noisy web data effectively to perform image classification, and achieve the first place in the WebVision 2018 challenge.

The rest of this paper is organized as follows. In Section 2, we briefly review related works. In Section 3, we summarize five key challenges to exploit large-scale noisy data. Accordingly, our URNet along with five reweighting strategies are detailedly described in Section 4. Finally, we show the experimental results in Section 5 and conclude this paper in Section 6.

2 RELATED WORK

Benefitting from the carefully annotated big visual dataset and powerful convolutional neural networks, researchers have achieved great progresses recently on the image classification problem. In 2012, Alex *et al.* [2] first used GPU to accelerate the time-consuming training process of a deep CNN model and obtained a great improvement than previous shallow models. This work demonstrates the powerful representative ability of a deep model when processing images. After that, it is generally assumed that deeper and wider models would obtain better performance due to their more powerful representative ability. Some works then focus on increasing the depth as well as the width. For example, Simonyan *et al.* [3] proposed the VGGNet and successfully trained it, which extends the depth of a single model from several weight layers to 19. This work breaks previous depth limit and demonstrates that it is possible to effectively construct and train a very deep model. After that, He *et al.* [5] proposed the residual structure as a block module to construct the ResNet. Benefitting from this structure, gradient can directly flow from high layers to low layers. This technology makes it possible to effectively train an ultra-deep model with 152 weight layers. Moreover, the model doesn't suffer seriously difficulties even with over one thousand weight layers. He *et al.* [6] further proposed ResNeXt to improve the residual block. Besides the depth, some other works focus on the width. For example, Szegedy *et al.* [4] proposed multi-scale inception module and construct the GoogleNet, which can encode multi-scale features in one layer. Huang *et al.* [21] proposed the densely connected networks, in which each layer connects with all its previous layers. Based on these works, some improved networks [22], [23], [24] were also proposed to achieve higher accuracy or computational efficiency. These methods usually focus on the image classification problem on carefully

annotated big visual dataset and pay less attention on the noisy or bias dataset.

To effectively exploit the large-scale noisy dataset, current works usually follow two types of methodologies. Some works focus on purifying the noisy dataset. These methods may resort to a clean sub-set [14], detect the noisy samples [15] and try to correct the wrong labels [16]. Specifically, Veit *et al.* [14] resorted to a clean sub-set and multi-task learning to jointly learn to clean noisy annotations and to accurately classify images. This work demonstrates how to use the clean annotations to reduce the noise in the large-scale dataset before fine-tuning the network using both the clean set and the full set with reduced noise. Wang *et al.* [15] proposed an iterative learning framework to detect the noisy samples with open-set labels so as to pull away noisy samples from clean samples. To benefit from the noisy label detection, they designed a Siamese network to encourage clean labels and noisy labels to be dissimilar. A reweighting module is also applied to simultaneously emphasize the learning from clean labels and reduce the effect caused by noisy labels. Tanaka *et al.* [16] proposed a joint optimization framework of updating parameters and estimating true labels. They optimized the labels themselves instead of treating the noisy labels as fixed. The joint optimization of network parameters and the noisy labels can correct inaccurate labels as well as improve the performance of the classifier. This group of methods aims to learn the accurate model as well as correct the noisy labels. However, it is a more difficult task to correct the noisy labels simultaneously than to only learn an accurate model, which also may be not necessary.

The other works focus on decreasing the weight of noisy samples so as to alleviate their negative effects. These methods usually resort to the similarity between intra-classes features [17], multi-instance learning, attention network [19], or category-level supervision [20]. Specifically, Fu *et al.* [17] assumed that the features of true positive samples in a class are similar and the features of noisy samples are very different. They embedded the feature map of the last deep layer into a new affinity representation, and further minimized the discrepancy between the affinity representation and its low-rank approximation. Zhuang *et al.* [19] proposed two unified strategies, i.e., random grouping and attention, to reduce the negative impact of noisy annotations. Specifically, random grouping is to stack multiple images into a single training instance and increase the labeling accuracy at the instance level. Attention, on the other hand, is to suppress the noisy signals from both incorrectly labeled images and less discriminative image regions. This method can increase the labeling accuracy at the instance level and effectively reduce the negative impact of noisy annotations. Niu *et al.* [20] adopted a more accessible category-level supervision strategy. In particular, they built a deep probabilistic framework upon variational autoencoder (VAE), in which classification network and VAE can jointly leverage category-level hybrid information. Then, the method was extended for domain adaptation followed by a low-rank refinement strategy. This group of methods focuses on reweighting the noisy samples, however, besides noise, data bias are also ubiquitous in large-scale real-world data.

3 CHALLENGE ANALYSIS

To motivate the algorithm design, we analyze the large-scale noisy web data, i.e., the dataset used in the WebVision Challenge 2018 (denoted as WebVision 2.0), and summarize five primary difficulties. These difficulties are not specific only for WebVision but generally exist in the real-world data. Based on these analysis, we propose the ubiquitous reweighting strategy as well as the URNet. We also expect that our analysis could inspire the researchers to design more specific algorithms so as to exploit the large-scale noisy real-world data effectively.

First, as is well-known, the Pareto principle extensively exists and the distribution of real-world things has high imbalance. To be



Fig. 1. The data have high intra-classes diversity and inter-class similarity, e.g., in the class 973 and 3440, instances in one class are very different from each other, while some instances belonging to different classes are quite similar.

specific, the Internet users do not fairly share each class of images, and the number of images in each class varies in a really wide range. The number of images is large in a common class, while it is small in a uncommon class. Actually, the number of images in a class varies from about three hundred to more than ten thousand. If we assign the same weight for each instance, the model will pay more attention on the class with more instances. And the class with small size will be overwhelmed by the class with large size. To alleviate this problem, the straight-forward method is to random sample a subset from the large class. However, this method can only make use of partial data. In this paper, we make each class to have almost the same influence on the model via reweighting each instance. Then the model can pay equal attention on each class while exploit all of the instances.

Second, the instances usually have high intra-classes diversity and inter-class similarity. As shown in Fig. 1, due to the various forms, shapes, illuminations, appearance and views, the instances in the same class may be extremely different from each other. However, these instances have the same semantic concept. On the other hand, the instances in one class may be really similar to the instances in another class due to their similar semantic concept, e.g., the class 973 and 3,440 in Fig. 1. Due to this phenomenon, some instances are really difficult to tackle. Intuitively, these difficult instances should be paid more attention by the model. To this end, we propose to cluster all the instances according to their appearance similarity. Intuitively, the instances in the large cluster may be similar to more instances from other classes, which should be paid more attention. That is to say, each instance can be reweighted by its cluster size. In this manner, we can automatically detect the difficult instances and reweight them.

Third, the instances in the dataset are directly retrieved from the image websites, i.e., google and flickr, using query words. The query words are taken as the semantic label. Although these modern image search engines have made great progresses recently, there are still considerable amount of inaccurate instances, which are noise for the dataset and would confuse the model. To suppress their negative influence, we can detect them or may further correct them. However, this process is more difficult and may also introduce additional noise. In this paper, we adopt a more flexible method and estimate the confidence of each predicted instance, which is then used to reweight this instance. More confidence the instance is, more weight it has.

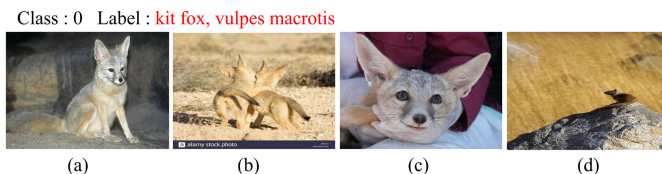


Fig. 2. Insufficient representative instances, e.g., image (a) is an representative instance and has more information than others, but may be overwhelmed by the uncommon instances like (b), (c), and (d).

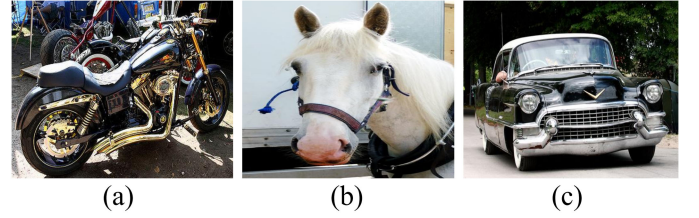


Fig. 3. One instance may belong to several classes simultaneously due to the ambiguous class labels, e.g., image (a) may belong to class 2334: motorcycle policeman, speed cop, motorcycle cop (id: label), and class 3369: trail bike, dirt bike, motorcycle bike, image (b) may belong to class 0331: seed, conker seed, and class 1044: Equus caballus, female horse, horse, image (c) may belong to class 0268: jeep, landrover, and class 0563: paddle wheel, paddlewheel.

Fourth, in one specific class, some instances have more information while some others have less, e.g., in Fig. 2, the first image may contain more information than the others. We refer the instances like Fig. 2a as representative instances. However, the instances in one class usually have a very high diversity of appearance, pose, lightness, and so on. This leads to insufficient representative instances, which may be overwhelmed by uncommon instances. To alleviate this problem, we resort to the bag-specific instance saliency technology to discover the representative instances and highlight them.

Last but not least, due to the semantic confusion and ambiguous class labels, all the classes are not orthogonal to each other. This means that some instances may actually belong to several classes at the same time, e.g., the images in Fig. 3. The model will be confused to tackle these instances. To alleviate this problem, we resort to a soft label as the optimization target rather than a hard label. With the soft label, one instances can be regarded as belonging to several classes at the same time.

4 UBIQUITOUS REWEIGHTING

Based on our analysis, we propose a ubiquitous reweighting network to alleviate the five challenges. Specifically, to handle imbalanced classes, a class-weighted loss is defined, which is to restrain the effect of classes with large size, while enhance the effect of the classes with small size. To tackle the high intra-classes diversity and inter-class similarity, we designed a cluster-weighted strategy. With this strategy, the instances in confusable clusters are further emphasized. To alleviate the imprecise instances, a model is trained to predict the correlation score of each instance relative to its query words. Each instance is reweighted according to its correlation score. The higher correlation score, the higher weight. In consideration of insufficient representative instances, the instances are organized into small bags. The salient instance in a bag usually contain more information, which is reweighted via an attention network. To alleviate the ambiguous class labels, not only the ground truth label but also the label predicted by the baseline model are both used to formulate a joint loss function.

4.1 Reweighting Class

To perform the image classification task, we construct our URNet based on ResNeXt-101 [6], which is among state-of-the-art image classification models. The detailed structure of the network can be found in [6]. The only difference lies in the output layer, which is modified to $C = 5,000$ dimensions, which are corresponding to the 5,000 classes in the WebVision 2.0 dataset. We first train the model via minimizing the standard cross entropy loss

$$\mathcal{L}_{sce} = - \sum_{i=1}^N \sum_{j=1}^C y_{ij} \log(p_{ij}), \quad (1)$$

where N is the total number of instances in the dataset, $y_{ij} \in \{0, 1\}$ and $y_{ij} = 1$ only if the instance \mathcal{I}_i is belong to the j th class \mathcal{C}_j , p_{ij} is

the probability of $\mathcal{I}_i \in \mathcal{C}_j$ which is predicted by the model. This trained model is taken as the baseline model.

From the standard cross entropy loss (1), we can see that each instance contributes equally to the loss function. However, if the imbalanced class size is taken into consideration, the classes with a few instances will be overwhelmed by the classes with a lot of instances. To alleviate this problem, a straight-forward solution is to use balanced cross entropy loss [25], where the standard cross entropy loss is then modified to the class weighted loss as

$$\mathcal{L}_{wce} = - \sum_{i=1}^N \sum_{j=1}^C w_j^c y_{ij} \log(p_{ij}), \quad (2)$$

where w_j^c is the weight of each class, which can be simply defined as inversely proportional to the class size

$$w_j^c = \frac{r_j}{\sum_{j=1}^C r_j}, \text{ where } r_j = \frac{1}{N_j}. \quad (3)$$

In this way, each class has the same weight to the loss function. Furthermore, to be more flexible, a parameter $\alpha \in [0, 1]$ is introduced to adjust from equal instance weight to equal class weight, then the class weight

$$w_j^c = (1 - \alpha) + \alpha \frac{r_j}{\sum_{j=1}^C r_j}. \quad (4)$$

When $\alpha = 0$, each instance contributes equally to the loss, while when $\alpha = 1$, each class contributes equally to the loss. In our experiments, we set $\alpha = 0.5$ empirically.

4.2 Reweighting Cluster

The large-scale noisy data have high intra-classes diversity and inter-class similarity, which makes it more difficult to correctly classify the instances, especially for them which are similar to the instances from other classes. Inspired by the unsupervised algorithms, e.g., [26], [27], which aims to maximize inter-class similarity and minimize intra-class similarity, we design a reweighting strategy based on unsupervised clustering to increase the weight of such instances.

We first evaluate the inter-class similarity and construct a confusion matrix \mathcal{M} for all classes, where an element \mathcal{M}_{jk} records the number of instances which are belong to the j th class but classified as the k th class by the baseline model. For a class \mathcal{C}_j , its four most confused classes $\mathcal{C}_{jk_1}, \mathcal{C}_{jk_2}, \mathcal{C}_{jk_3}, \mathcal{C}_{jk_4}$ are selected according to the confusion matrix. Combined with the class \mathcal{C}_j , all instances in these five classes are unsupervisedly clustered into five groups $\mathcal{G}_{j1}, \mathcal{G}_{j2}, \mathcal{G}_{j3}, \mathcal{G}_{j4}, \mathcal{G}_{j5}$ using k-means algorithm [26]. Specifically, we take the output of the layer before the classification layer of the baseline model as the image features.

Due to the high intra-classes diversity and inter-class similarity, the instances from different classes may be assigned into a group only because of their similar appearance, and vice versa. Moreover, the group sizes usually vary in a wide range. It is obvious that one instance in a big group has similar appearance to many other instances. Such an instance is usually difficult for classification and should be paid more attention by the model. In other words, each instance should be reweighted according to its group size. Then the cluster weight is defined as

$$w_{ji}^g = \frac{N_{\mathcal{G}_{ji}}}{\sum_{l=1}^5 N_{\mathcal{G}_{jl}}}, l \in \{1, 2, 3, 4, 5\}, \quad (5)$$

where $N_{\mathcal{G}_{ji}}$ is the number of instances in group \mathcal{G}_{ji} . The cluster weight of each instance can be further defined as

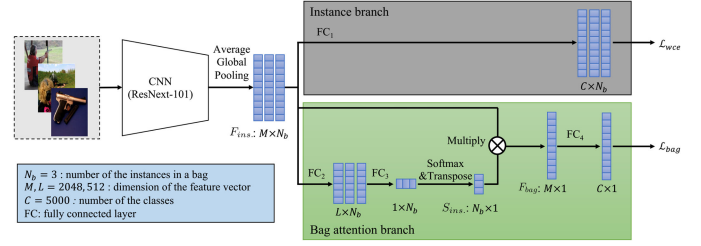


Fig. 4. We add the bag attention branch to reweight bag-specific instance saliency. With this branch, the model can pay more attention on the representative instances.

$$w_{ij}^g = \begin{cases} w_{ji}^g, & \text{if } \mathcal{I}_i \in \mathcal{C}_j \cap \mathcal{G}_{ji} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

Taking cluster weight into consideration, the weighted cross entropy loss (2) can be modified as

$$\mathcal{L}_{wce} = - \sum_{i=1}^N \sum_{j=1}^C w_{ij}^g w_j^c y_{ij} \log(p_{ij}). \quad (7)$$

Benefiting from the cluster weight, the model can pay more attention to the instances which are easily confused with other classes, and the performance is significantly improved. Moreover, as the clustering algorithm is unsupervised, there is no more manual work to be done.

4.3 Reweighting Instance

There are many imprecise instances in the large-scale noisy data. To alleviate this problem, we predict the confidence of each instance and then reweight it accordingly. The more confident the instance is, the higher weight it should have. A straight-forward solution is to train a model which aims to predict the confidence. However, the influence of imprecise instances is ubiquitous. It is difficult to construct a clean training set. By observing the data, we find that the top ranking images of each class which are retrieved by Google are usually high confident. Therefore, we select the top 30 ranking images along with their labels as the positive training samples. At the same time, for each image, we random select a label except its ground-truth label to construct the negative training samples.

Instead of constructing an regression or classifier model, we resort to construct the relation between image and text features [28], [29]. As shown in Fig. 5, our confidence model takes the correlation between the image features $\mathcal{V}_{\mathcal{I}_i}$ and the text features $\mathcal{V}_{\mathcal{T}_i}$ as the confidence score, where the text features are extracted from the label text corresponding to the image class, the image features are extracted using the baseline model. The label text are actually the

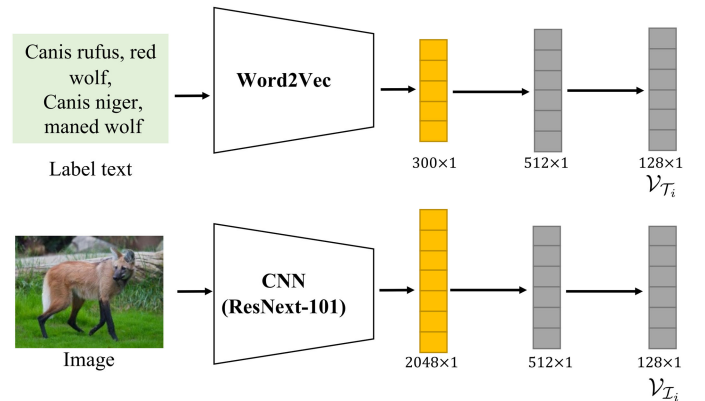


Fig. 5. We take the correlation between the image features $\mathcal{V}_{\mathcal{I}_i}$ and the text features $\mathcal{V}_{\mathcal{T}_i}$ as the confidence score.



Fig. 6. Some examples of clustering result on class 218 and its top-4 confusing classes.

search terms during searching the images with this label which are provided by the WebVision dataset 2.0. With this design, we can embed the image features into the text features manifold which can reflect the relationships between all the classes. To extract features from the words, we adopt the fastText-300 [30] model, whose last layer is replaced by two fully connected layers and output text features \mathcal{V}_{T_i} . The last layer of the baseline model is also replaced by two fully connected layer, which output image features \mathcal{V}_{I_i} . Note that, the dimensions of feature \mathcal{V}_{I_i} and \mathcal{V}_{T_i} are the same. Then the model can be trained via minimizing the correlation loss

$$\mathcal{L}_{tic} = - \sum_{i=1}^{N'} (\cos(\langle \mathcal{V}_{I_i}, \mathcal{V}_{T_i} \rangle) - \cos(\langle \mathcal{V}_{I_i}, \mathcal{V}_{\tilde{T}_i} \rangle)), \quad (8)$$

where $N' = 30 \times C$, \tilde{T}_i is the text of a random other class which is different from T_i . By minimizing this loss, we can maximize the correlation between the image features and its ground-truth text features, while minimize the correlation between the image features and other text features. As soon as the model is trained, the correlation $w_i^T = \cos(\langle \mathcal{V}_{I_i}, \mathcal{V}_{T_i} \rangle)$ can be used to evaluate the confidence of each instance in the dataset. Then the weighted cross entropy loss (7) can be modified as

$$\mathcal{L}_{wce} = - \sum_{i=1}^N \sum_{j=1}^C w_i^T w_{ij}^G w_j^C y_{ij} \log(p_{ij}). \quad (9)$$

4.4 Reweighting Bag-Specific Instance Saliency

The noisy large-scale dataset have high intra-classes diversity, which leads to insufficient representative instances. To highlight such instances, we reweight each instance according to its saliency, as the saliency can evaluate the representativeness. The straightforward method is to adopt attention network [31], [32], [33], which is also similar to multi-instance learning [34], [35], [36]. To this end, we take N_b instances from one class as a bag, where $N_b = 3$. Each bag is assumed to have at least one representative instance. Even though only 30 percent instances are representative, this assumption still will be satisfied in more than 97 percent cases.

As shown in Fig. 4, besides the standard instance branch, we add a new branch named bag attention branch. Its input are the N_b feature vectors extracted from the images in a bag respectively, which is denoted as $F_{ins.}$. In this branch, there are 2 fully connected layers, i.e., FC_2 and FC_3 , which can predict the saliency of all instances along with softmax normalization operation, which is denoted as $S_{ins.}$. Then we can take the weighted summation $F_{ins.} \times S_{ins.}$ as the features of the bag, which is denoted as F_{bag} . Another fully connected layer FC_4 can further estimate the probability when the image is belonging to each class. We use the cross entropy loss function to train this branch as well as the backbone network. The bag attention branch can highlight the saliency instance and its features so as to enable the model to pay more attention on the representative instances.

4.5 Reweighting Label

For the dataset, only one class is assigned for each instance. However, an instance may actually belong to several classes at the same time. These instances may confuse the model. To this end, we further reweight the loss (9) with soft label as

$$\mathcal{L}_{wce} = - \sum_{i=1}^N \sum_{j=1}^C w_i^T w_{ij}^G w_j^C (\beta y_{ij} + (1 - \beta) \hat{y}_{ij}) \log(p_{ij}), \quad (10)$$

where $\hat{y}_{ij} \in \{0, 1\}$ and $\hat{y}_{ij} = 1$ only if \mathcal{I}_i is supposed to belong to class \mathcal{C}_j , the parameter β is to balance the ground-truth label and the supposed label. In our experiments, we set $\beta = 0.8$ empirically.

To obtain the supposed label, we also resort to the baseline model. Actually, its predicted result is directly taken as the supposed label. Note that, when the predicted result is same with the ground-truth, the loss is unchanged. When the predicted result is different with the ground-truth, instead of trusting any one, we think that the instance may have some uncertainties and this loss is a trade-off.

5 EXPERIMENTS

Benefiting from the ubiquitous reweighting network along with some other experimental tricks, our model ranks the first place at the WebVision image classification challenge 2018. In this section,

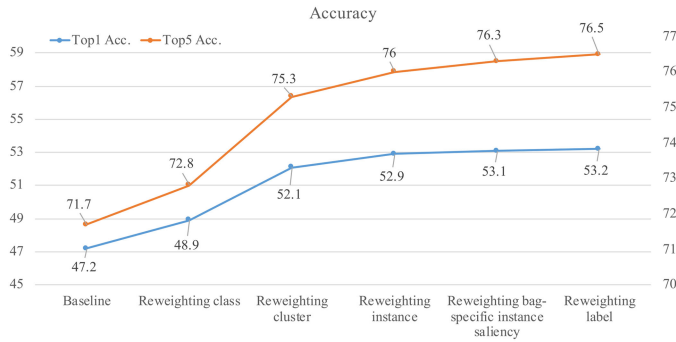


Fig. 7. The effectiveness of each reweighting strategy. From the figure, we can see that the performance is improved steadily with our reweighting strategies.

we introduce the experimental details and verify the effectiveness of each module of our URNet.

In the experiments, we train our model using the training set in WebVision dataset 2.0, and evaluate it using the validation set. To train the baseline model ResNeXt-101, we take the previous model architecture designed for ImageNet and modify the output layer so as to make it corresponding to the 5,000 classes defined by WebVision dataset 2.0. Then the model is optimized by minimizing the cross entropy loss on the training set. The baseline model is then adopted to further construct the URNet step by step. And it is also taken to construct the auxiliary data needed by URNet, such as to construct the confusion matrix, predict the confidence of each instance.

5.1 Effectiveness of Each Step

As shown in Fig. 7, the baseline model achieves top-5 accuracy 71.7 percent. Based on the baseline model, we add each reweighting strategy one by one, i.e., reweighting class, reweighting cluster, reweighting instance, reweighting bag-specific instance saliency, reweighting label. Then the performance is improved steadily to 72.8, 75.3, 76, 76.3, and 76.5 percent respectively.

We can see that reweighting cluster is the most helpful strategy, which can increase the top-5 accuracy by 2.5 percent. As shown in Fig. 6, we demonstrate the clustering result of class 218 and its confusing four classes. We can see that, benefiting from this strategy, the model can pay more attention to the instances which are confusing with the instances from other classes. The large improvement also verifies the effectiveness of our reweighting cluster strategy.

The second important strategy is reweighting class. Class bias not only exists in the WebVision dataset 2.0 but also ubiquitously exists in most real-world data. The experimental results show that our reweighting class strategy can alleviate this problem and increase the top-5 accuracy by 1.1 percent.

Besides, with reweighting bag-specific instance saliency strategy, the model can pay more attention on the representative instances, which can increase the top-5 accuracy by 0.3 percent. The above three strategies focus on alleviate the data bias. To be specific, reweighting class strategy can alleviate the bias arising from unbalanced class sizes. Reweighting cluster strategy can alleviate the bias arising from unbalanced instance difficulty. Reweighting bag-specific instance saliency strategy can alleviate the bias arising from unbalance instance representativeness.

On the other hand, the other two strategies focus on the instances with noisy labels. With reweighting instance strategy, the model can

TABLE 2
Some Experimental Tricks and Their Effects, Which are Used in the WebVision Challenge

Tricks	Top-5 accuracy	Improvement
Remove noise out of top 15	77.0%	0.5%
Model ensemble	78.3%	1.3%
Multi-crop testing	79.3%	1.0%
Multi-scale testing	79.8%	0.5%

TABLE 3
The Results of WebVision Image Classification Challenge 2018

Rank	Team name	Top-5 Accuracy
1	Vibranium (Our team)	79.25%
2	Overfit	75.30%
3	ACRV_ANU	69.56%
4	EBD_birds	69.44%
5	INFIMIND	68.74%
6	CMIC	61.14%

pay more attention on the more confident instances, and the influence of less confident instances can be suppressed. With reweighting label strategy, we can alleviate the issue when an instance belongs to two classes at the same time. Benefiting from these two strategies, our model can be further improved by 0.9 percent on top-5 accuracy.

To exploit the influence of each individual strategy, we conduct an ablation experiment. We add each strategy to the baseline model or exclude each strategy from the final model. The top-1 and top-5 accuracy of each model is presented in Table 1, from which we can see that each reweighting strategy is effective.

5.2 Final Results in the WebVision 2018 Challenge

Besides our reweighting strategies, in the WebVision 2018 challenge, we also use some useful tricks to further improve the performance. The tricks and the corresponding effectiveness are show in Table 2. The first trick is that we found that the instances from Flickr usually are unreliable. So, we only use the top 15 instances from Flickr for each class. This trick can improve the performance of our final model by 0.5 percent on top-5 accuracy.

The others are three common tricks, i.e., model ensemble, multi-crop testing, multi-scale testing. For model ensemble, we train 29 models and ensemble their results. This trick can improve the top-5 accuracy by 1.3 percent. For multi-crop and multi-scale tricks, we resize and crop the input images multiple times and fuse the prediction results. These tricks can improve the top-5 accuracy by 1.0 and 0.5 percent respectively.

Benefiting from our URNet, we achieve the first place on the WebVision Image Classification Challenge 2018. As shown in Table 3, our method achieves 79.25 percent top-5 accuracy on final test data, which outperforms the second method by about 4 percent. The results verify the effectiveness of our URNet.

To exploit the difference between the models which are trained on clean or noisy datasets, we fine-tune each model to perform object detection task on PASCAL VOC 2007 dataset. To this end,

TABLE 1
The Influence of Each Strategy When it is Added (Denoted With +) to the Baseline Model or Excluded (Denoted With -) From the Final Model

	Baseline	+Class	+Cluster	+Instance	+Saliency	+Label	-Class	-Cluster	-Instance	-Saliency	-Label	Final
Top-1	47.2%	48.9%	51.2%	52.1%	50.9%	47.7%	52.5%	52.8%	52.2%	51.3%	53.1%	53.2%
Top-5	71.7%	72.8%	74.6%	75.5%	75.5%	72.2%	74.8%	75.1%	74.7%	74.1%	76.3%	76.5%

TABLE 4
The Object Detection Performance of Different Fine-Tuned Models Which are Pre-Trained Using Different Datasets and Strategies

Pre-train dataset	Model	mAP
ImageNet	Baseline model	73.05%
WebVision 2.0	Baseline model	45.24%
WebVision 2.0	URNet	51.35%

we embed the pre-trained model into the Faster-RCNN framework [37] using its PyTorch implementation.¹ The results are evaluated using mean average precision (mAP) and presented in Table 4 which demonstrates that the baseline model suffers from a dramatically decrease on detection performance due to the noise data. However, with the proposed ubiquitous reweighting strategy, the performance can be increased by about 6.1 percent. This experiment also verifies the effectiveness of the proposed ubiquitous reweighting strategy.

6 CONCLUSION

In this paper, we demonstrate that every instance has the potential to contribute positively by alleviating the noise and bias via reweighting the influence of different classes as well as their labels, large instance clusters, small instance bags and their confidence. In this manner, the influence of the noise and bias in the web data can be gradually alleviated, which leads to the steadily improving performances of URNet. Experimental results in the WebVision 2018 challenge with 16 million noisy training images from 5,000 classes show that our approach outperforms several state-of-the-art models and ranks the first place in the image classification task.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grants 61922006, 61672072, and 61532003, and also supported by the Beijing Nova Program (Z181100006218063).

REFERENCES

- [1] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2014.
- [4] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
- [5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [6] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5987–5995.
- [7] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
- [8] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning affordance for direct perception in autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2722–2730.
- [9] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, no. 4/5, pp. 421–436, 2018.
- [10] G. Litjens et al., "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, 2017.
- [11] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, Feb. 2015, Art. no. 529.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.

- [13] W. Li, L. Wang, W. Li, E. Agustsson, and L. V. Gool, "WebVision Database: Visual learning and understanding from web data," 2017, *arXiv:1708.02862*.
- [14] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. J. Belongie, "Learning from noisy large-scale datasets with minimal supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6575–6583.
- [15] Y. Wang et al., "Iterative learning with open-set noisy labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8688–8696.
- [16] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5552–5560.
- [17] J. Fu, Y. Wu, T. Mei, J. Wang, H. Lu, and Y. Rui, "Relaxing from vocabulary: Robust weakly-supervised deep learning for vocabulary-free image tagging," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1985–1993.
- [18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015, Art. no. 436.
- [19] B. Zhuang, L. Liu, Y. Li, C. Shen, and I. D. Reid, "Attend in groups: A weakly-supervised deep learning framework for learning from web data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, Art. no. 7.
- [20] L. Niu, Q. Tang, A. Veeraraghavan, and A. Sabharwal, "Learning from noisy web data with category-level supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7689–7698.
- [21] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2261–2269.
- [22] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
- [23] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5987–5995.
- [24] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4467–4475.
- [25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2999–3007.
- [26] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *J. Roy. Statist. Soc. Ser. C, Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.
- [27] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2002, pp. 849–856.
- [28] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5005–5013.
- [29] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," 2014, *arXiv:1411.2539*.
- [30] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, 2017.
- [31] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "ABCNN: Attention-based convolutional neural network for modeling sentence pairs," 2015, *arXiv:1512.05193*.
- [32] E. W. Teh, M. Ročan, and Y. Wang, "Attention networks for weakly supervised object localization," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 52.1–52.11.
- [33] F. Wang et al., "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6450–6458.
- [34] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3460–3469.
- [35] E. Alpaydm, V. Cheplygina, M. Loog, and D. M. Tax, "Single-vs. multiple-instance classification," *Pattern Recognit.*, vol. 48, no. 9, pp. 2831–2838, 2015.
- [36] W. Li and N. Vasconcelos, "Multiple instance learning for soft bags via top instances," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4277–4285.
- [37] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.

1. <https://github.com/jwyang/faster-rcnn.pytorch>