Training-ValueNet: Data Driven Label Noise Cleaning on Weakly-Supervised Web Images

Luka Smyth
Department of Computer Science
University of Exeter
Exeter, UK
ls479@exeter.ac.uk

Dmitry Kangin
Department of Computer Science
University of Exeter
Exeter, UK
d.kangin@exeter.ac.uk

Nicolas Pugeault
Department of Computer Science
University of Exeter
Exeter, UK
n.pugeault@exeter.ac.uk

***Abstract*—Manually labelling new datasets for image classification remains expensive and time-consuming. A promising alternative is to utilize the abundance of images on the web for which search queries or surrounding text offers a natural source of weak supervision. Unfortunately the label noise in these datasets has limited their use in practice. Several methods have been proposed for performing unsupervised label noise cleaning, the majority of which use outlier detection to identify and remove mislabeled images. In this paper, we argue that outlier detection is an inherently unsuitable approach for this task due to major flaws in the assumptions it makes about the distribution of mislabeled images. We propose an alternative approach which makes no such assumptions. Rather than looking for outliers, we observe that mislabeled images can be identified by the detrimental impact they have on the performance of an image classifier. *We introduce training-value* as an objective measure of the contribution each training example makes to the validation loss. We then present the training-value approximation network (Training-ValueNet) which learns a mapping between each image and its training-value. We demonstrate that by simply discarding images with a negative training-value, Training-ValueNet is able to significantly improve classification performance on a held-out test set, outperforming the state of the art in outlier detection by a large margin.**

I. Introduction

Large scale datasets such as ImageNet [1] have played a vital role in recent advances in image classification. Despite this progress, the manual labelling of large new datasets often remains prohibitively expensive and time consuming. A promising alternative is to utilise the abundance of freely available images on the web for which search engine queries or surrounding web page text can act as a natural source of weak supervision (e.g. [2] [3]). Unfortunately, many studies have shown that the label noise present in these 'webly-supervised' datasets can significantly degrade classification performance [4] [5] [6]. It is therefore desirable to develop methods capable of label noise cleaning in absence of full human supervision.

Existing methods can be categorized according to the level of human supervision they require. Semi-supervised approaches (e.g. [7] [8]) require a subset of training examples to manually labeled in order to learn a model capable of identify and removing mislabeled images from the dataset. Whilst these methods relieve some of the burden of manual labelling, their performance remains intrinsically linked to number of manual labels that the user is able to provide. Alternatively, many outlier detection methods have been proposed [9] [10] [11] with the clear advantage of requiring no human supervision. Whilst these methods are shown to be effective in cases where label noise is artificially generated via some random process, recent works have raised concerns about their effectiveness on real-world weakly-supervised datasets [6]. In this setting, label noise is more typically caused by some systematic error and as a result, the assumed equivalence between mislabeled images and outliers is rarely valid.

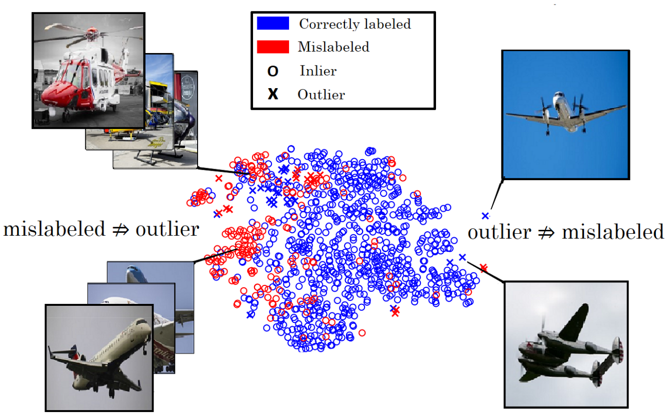

Fig. 1: A t-SNE [12] visualization of images from the Aircraft-7 dataset whose noisy class labels are all 'propeller plane'. Each image was manually annotated as either correctly labeled (blue) or mislabeled (red) and the 50 most outlying images are marked with an 'x'. Mislabeled images (examples left) are generally not outlying whilst true outliers (examples right) are rarely mislabeled. This demonstrates why outlier detection is often an unsuitable approach for identifying mislabeled images.

This phenomena is illustrated in Fig. 1 which depicts a t-SNE [12] visualisation of 1000 images from the Aircraft-7 dataset whose noisy class labels are all 'propeller plane'. These images were scraped from Flicker by searching for the class name 'propeller plane'. We manually annotate each image as being either correctly labeled (blue in Fig. 1) or mislabeled (red). Furthermore, we use the mean proximity of each image to its $K{=}10$ nearest-neighbours in this embedding as a rudimentary metric for outlier detection and mark the fifty most outlying images with an 'x' in Fig. 1.

Outlier detection methods for label noise cleaning all rely on the same fundamental assumption that mislabeled images are outliers, sparsely distributed and visually dissimilar from both correctly labeled images and other outliers. In practice however, we have found that this assumption rarely holds true in the case of weakly-supervised image datasets from the web. In Fig. 1 for example, we see that mislabeled images form several dense clusters, which in this case all correspond to different types of aircraft that were mistakenly included in the search results for 'propeller plane'. In this example, outlier detection methods will fail to identify a significant proportion of mislabeled images because they are simply not outlying.

The second flawed assumption of outlier detection is the converse of the first. Not only are many mislabeled images not outlying, true outliers are often not mislabeled. In practice we have found that many outliers are in fact correctly labeled yet visually unusual corner cases. In our propeller plane example, these corner cases account for 35 of the 50 most outlying examples. Outlier detection methods are unable to distinguish these valuable corner cases from truly mislabeled examples and mistakenly removing these images can lead to a damaging lack of diversity in the training set.

In this paper, we present a new approach for performing label noise cleaning on weakly-supervised datasets which serves as a conscious move away from outlier detection. Rather than searching for outliers, we exploit what we believe is a more reliable characteristic of mislabeled images. We observe that training on mislabeled images reliably detriments the performance of an image classifier whereas correctly labeled images almost always offer some positive value. As a result, we are able to drastically reduce the proportion of mislabeled examples in a dataset by simply identifying and removing any example which we expect to cause a net detriment to the final performance of the classifier. We adopt the term 'training-value' (first introduced in [13]) to refer to our measure of expected impact an individual training example has on the validation loss during training. We formalise our definition of training-value in section 3A of the paper. In order to efficiently estimate the training-values of large numbers of images, we obtain direct Monte Carlo estimations of training value for a small subset of training examples and then generalize this mapping to unseen images via the use of a training-value approximation network (Training-ValueNet) which we present in section 3B.

Experimental results on two challenging webly-supervised datasets demonstrate that by simply discarding all images with a negative predicted training-value, we are able to significantly reduce label noise and improve the final classification performance on a held-out test set. Notably, we demonstrate that Training-ValueNet achieves a substantial 14.8% (percentage points) lower label noise detection error on the Clothing 1M dataset [3] compared with with DRAE [11], the current state of the art in unsupervised outlier detection for label noise cleaning.

II. Related Work

Unsupervised outlier detection. Outlier detection is the most widely studied approach for label noise cleaning (see [14] for a comprehensive survey). Popular neighbour-based methods such as local outlier factor [15] and probabilistic methods such as robust kernel density estimation (extension of the classical *Parzen-Rosenblatt* KDE) have been implemented in recent works [11] [10] but are shown to perform poorly when the proportion of outliers approaches or exceeds 50%. In response to this, Liu et. al. proposed an unsupervised one-class learning (UOCL) approach which utilizes a max margin classifier and is robust to high outlier proportions. More recently DRAE [11] uses the reconstruction error of an autoencoder to identify outliers. As the autoencoder passes images through an intermediate, low-dimensional representation, it is forced to learn only the statistical regularities of the dataset. In this way, outliers receive a higher reconstruction error and can be identified as such. Comparisons on several datasets place DRAE as the state of the art in this category having convincingly outperformed UOCL.

Semi-Supervised Label Noise Cleaning. Semi-supervised approaches require a subset of training examples to be manually labelled in order to learn a model of label noise (e.g. label prop [16] and label spread [17]). Yu et. al. introduced an iterative, human in the loop approach which was used to construct the large scale LSUN dataset [7] at a fraction of the cost of full supervision. More recently, CleanNet [8] learns a relevance score for each image based on its similarity to a class prototype. Crucially it does so in such a way that verification labels are only required for a subset of classes. CleanNet is demonstrated to be effective on several webly-supervised benchmarks despite only a small percentage of images being manually labeled. Despite the impressive efficiency of CleanNet, labeling even a very small proportion of training examples will remain unfeasible for datasets that are order of magnitudes larger than ImageNet.

Curriculum Learning for Label Noise. The curriculum learning (CL) paradigm first introduced by Bengio et. al. [18] has recently achieved notable success in dealing with label noise. CL methods do not seek to remove mislabelled images but instead learn a principled ordering of the training data to reduce their negative impact. MentorNet [19] guides the attention of a 'StudentNet' classifier (via sample weighting) based on feedback it receives during training. More recently, CurriculumNet [20] uses the relative proximity of images in feature space to construct and then train on three subsets of progressively noisier training examples. Direct comparisons place CurriculumNet as the state of the art among CL methods for label noise. CurriculumNet was also notably the top performing entry to the Webvision [21] challenge 2017 - a noisy equivalent of the ILSVRC ImageNet challenge.

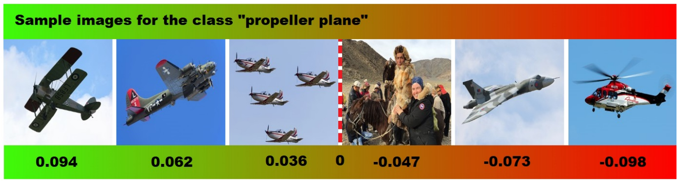

Fig. 2: Example images from the ‘propeller plane’ class of the Aircraft-7 dataset alongside their associated training-values. Images with a negative training-value are presumed mislabeled and discarded from the final training set.

III. Label Noise Cleaning with Training-ValueNet

In this section we present our method for performing label noise cleaning on weakly-supervised datasets for image classification. In section 3A we define training-value, our measure of the expected impact that a single training example has on the performance of an image classifier. We then introduce the Training-ValueNet in section 3B, a value function approximation network which enables us to efficiently estimate the training-values of large numbers of images. Images which are predicted to have a negative training-value are presumed to be mislabeled under our method and are discarded from the final training set.

A. *Defining training-value*

The training-value of an image is defined with respect to a specified classification task T with K possible classes. For this task, we assume access to small manually labelled validation and test sets X^V and X^T as well as a training set of weakly-labelled images $X^W{=}\{(x_i, y_i), .., (x_n, y_n)\}$, where x_i is the ith training example with noisy class label $y_i{\in}\{1, .., K\}$. Let $f(x; \theta)$ be our image classifier with weights θ and let L be the loss function which we minimize on X^W using some iterative gradient descent algorithm.

We define the training-value $V(x_i)$ of image $x_i{\in}X^W$ as the expected immediate improvement in validation loss that is obtained as a result of training on x_i at a randomly selected time-step t. In practice however, it would be very difficult to discern the impact that a single training example has on the loss if training is carried out using batches of multiple images or with momentum smoothing. In this initial work, we circumvent this issue in the simplest way possible. We assert that, for the purposes of defining and later estimating training-value, training is to be carried out using the vanilla stochastic gradient descent algorithm, that is, using a batch size of one and no momentum smoothing or equivalent. We wish to emphasize however that these constraints are only enforced when we are obtaining direct estimates of training-value using Monte Carlo estimation. Once the training-values of all training examples have been estimated and mislabeled images removed from the dataset, there are no restrictions on the final training of the image classifier.

Under these constraints, the immediate change in validation loss at each time-step t is determined by a combination of the following three factors:

- The image $x^{(t)}$ which we train on at time t
- The current classifier weights $\theta^{(t)}$ at time t
- Stochastic factors such as dropout regularization [22]

Given this, we can express the immediate improvement in validation loss $\Delta L(X^V)$ obtained by training on example $x^{(t)}{=}x_i$ at time-step t, as an expectation over the updated network weights $\theta^{(t+1)}$ denoted by the random variable $\theta^{'}$,

$$\Delta L(X^V|\, x^{(t)}, \theta^{(t)}) = E_{\theta'}\Big[L(X^V|\,\theta^{(t)}) - L(X^V|\,\theta^{'})\Big] \qquad (1)$$

Finally, we arrive at our definition of the training-value $V(x_i)$ of image x_i by taking the expectation of this immediate change in loss over all all possible values for the current weights $\theta^{(t)}$ which we will denote by the random variable θ,

$$V(x_i|X^V) = E_{\theta}\Big[\Delta L(X^V|\, x^{(t)}{=}x_i, \theta^{(t)}{=}\theta)\Big] \qquad (2)$$

B. *Training-ValueNet*

Explicit evaluation of Eq.(2) is of course unfeasible for any reasonable sized problem. We can however obtain an unbiased, Monte-Carlo (MC) estimation $\bar{V}(x_i)$ for the training-value of image x_i by repeatedly training our image classifier using standard stochastic gradient descent (i.e. under the constraints laid out in 3A) and computing the mean improvement in validation loss that is obtained when we train on x_i. If X^W is large however, it would become prohibitively computationally expensive to obtain reliable estimates for all training examples in this manner.

It is for precisely this reason that we introduce the training-value approximation network (Training-ValueNet) which learns to predict the training-value of a training example based solely on the extracted convolutional features of that image. In this way, we can obtain direct MC estimates of training-value for just a small subset of X^W and then generalize this mapping to all remaining examples in the training set. This works under the assumption that visually similar images share consistent label correctness (i.e. they are both either mislabeled or correctly labeled).

Because the training-value of an image depends entirely on the correctness of it’s class label, it would not make sense to use a single training-value network to make predictions across all classes. In this case the network would have to predict the value of a training example in the absence any knowledge of its class label - an impossible task. Instead we train a separate network for each of the K classes of image. This allows each network to specialize in approximating the value function for just one class of image.

We have now formally defined the training-value of an image and explained the utility of the Training-ValueNet in allowing us to efficiently estimate the training-values of large numbers of images. We will now describe precisely how our method is carried out in four key steps:

Step 1: We first train a baseline convolutional neural network image classifier on the entire we training set X^W. We do this in order to extract the final convolutional layer 'bottleneck' features $f^c(x_i)$ which act as a fixed image representation moving forward. Using these bottleneck features during the subsequent MC estimation phase is drastically more efficient than training a large CNN from scratch over several repeated episodes.

Step 2: We obtain MC estimates of training-value for small, randomly chosen and class balanced subset $X^{W'}{\subset}X^W$ of n_T images per class. To do this we train a small MLP classifier using the previously extracted image features as input. We train for M total episodes of e epochs each, using a batch size of one and no momentum throughout (as per the constraints laid out in section 3A). At each time-step during training, we record the immediate change in validation loss $\Delta L^{(t)}(X^V)$ alongside the image $x^{(t)}$ which was trained on at that time-step. Once all episodes are complete, unbiased estimates $\bar{V}(x_i)$ are obtained of all images in our training subset $X^{W'}$ by simply averaging all recorded changes in validation loss which were recorded for x_i,

$$\bar{V}(x_i) = \frac{1}{M{\cdot}e} \sum_{ep=1}^{M} \sum_{t=0}^{e\cdot n_T} \left\{ \Delta L^{(t)}(X^V) \mid x^{(t)}{=}x_i \right\} \tag{3}$$

Step 3. Using these MC estimates as training targets, we train a separate Training-ValueNet for each class $C{\in}\{1,..,K\}$ in a supervised manner. Each Training-ValueNet is a simple MLP regression network which takes an image's convolutional bottleneck features ($f^c(x_i)$ (extracted in step 1) as input and returns a prediction $\hat{V}(x_i)$ for the training-value of x_i.

Step 4. Once we have trained the Training-ValueNet for each class, we use them to predict the training-value of all training examples in X^W. We now have predictions for the training-values of all examples in our training set X^W despite only training on a small subset of them in the MC estimation phase.

Using these predictions for training-value, we perform label noise detection by applying a simple threshold δ to the training-value. In this work we apply a universal threshold of $\delta{=}0$ across all classes. This is the principled choice considering that a negative training-value indicates that a training example is detrimental and thus presumably mislabeled. This means that for a given training example x_i with estimated training-value $\hat{V}(x_i)$,

$$\text{Image } x_i \text{ is classified : } \begin{cases} \text{mislabeled} & if\ \hat{V}(x) < 0 \\ \text{correctly labeled} & otherwise \end{cases}$$

IV. Experiments

A. *Datasets*

Clothing 1M [3]: The clothing 1M dataset contains precisely 1M images belonging to 14 categories of clothing item (e.g. shirt, sweater). Image labels were inferred from the presence of the class name in the surrounding web page text and as a result the estimated label accuracy is just 61.54%. An additional 50K/14K/7K manually labeled images are included for train/validation/test purposes. Of these additional images, a subset of 25K/7K/5k also have their original noisy labels provided. Using these overlapping labels we obtain the ground-truth correctness of the noisy class labels for these 25K training examples.

Aircraft-7: We use Flickr to build a new webly-supervised dataset of 75K images spanning seven classes of aircraft (airliner, glider etc.). Images for each class were obtained by searching Flickr for the class name and downloading the most relevant results. The estimated label accuracy for this dataset is 70% although this varies substantially between classes (see Table 3 for statistics). We take a random sample of 100 images from the corresponding ImageNet synset for each class to form a validation set and a further 100 as a test set. The remaining ImageNet images from each class were combined to form a fully-supervised training set for the purposes of comparison only.

B. *Training Parameters*

Our method requires a number of parameters to be set which we detail in this section.

Baseline Model: Our method requires a baseline model from which to extract bottleneck features and perform final image classification. To provide a fair comparison with recent works [8] [23] we fine-tune an ImageNet pre-trained ResNet-50 model [24] on the entire noisy training set. We follow the fine-tuning procedure outlined in [23]. We extract the 2048-dimensional final layer convolutional features for each image from this baseline model.

Monte-Carlo Estimation: We obtain MC estimates for the training-values of a random subset of $n_T{=}1000$ images per class. Over the course of $M{=}100$ training episodes of $e{=}1$ epoch each, we train a MLP classifier with no hidden layers using the extracted bottleneck features for these images as input. At each-time step, we record the immediate change in loss on a subset of $n_V{=}100$ images per class from the validation set.

Training-Value Net: We train a separate Training-ValueNet for each class using the MC estimates of training-value. Each Training-ValueNet is a MLP regression network with a single hidden layer of 1024 units. Training is carried out using mini-batch SGD with a batch size of 32 and 0.9 Nesterov momentum [25]. We also use dropout [22] after the hidden layer at a rate of 0.7.

C. *Label Noise Detection on Clothing 1M*

We evaluate our method on a label noise detection task on Clothing 1M. For this, we use the set of 25K training examples for which we have obtained the ground-truth correctness of their class labels. We carry out our method and classify each image as mislabeled if its training-value falls below the δ=0 threshold. We compare these predictions with the ground-truth correctness of the noisy class labels to obtain an average detection error rate across all classes. In Table 1 we compare our results with a number of existing semi-supervised and unsupervised methods as reported in [8]:

- **Semi-supervised** methods use the ground-truth correctness labels for a subset of the 25K images in order to train their model for label noise detection. Baselines such as a 2-layer MLP (used in [7]), kNN, SVM, label prop [16], and label spread [17] are reported as well as the state of the art CleanNet [8] itself. Error rate is reported on a held-out set.
- **Unsupervised** methods have no access to verification labels during training. We compare with DRAE [11], the state of the art in unsupervised outlier detection which was re-implemented by [8]. We also compare with the 'CleanNet unsupervised baseline' baseline whereby image features are simply averaged to form the reference set and query set embedding. Finally we include a naive baseline which assumes all noisy labels are correct.

In general, our approach to label cleaning would be considered semi-supervised as we require at least a small, cleanly labeled validation set. We wish to emphasize however that unlike all other semi-supervised methods we compare with in this paper, we make no use of the ground-truth correctness labels for any training examples. Accordingly, we have separated our result in Table 1 to avoid this confusion.

Training-ValueNet achieves an error rate of 23.66% which is a 6.9% improvement on the best unsupervised method, the CleanNet unsupervised baseline [8] and a substantial 14.8% better than DRAE [11]. This demonstrates the superiority of our approach over outlier detection methods when tested on real world label noise.

Label Noise Detection Results	
Method	Detection error (%)
Semi-supervised methods:	
MLP	16.09
kNN	17.58
SVM	16.75
Label prop [16]	17.81
Label spread [17]	17.71
CleanNet [8]	15.77
Unsupervised methods:	
Naive baseline	38.46
DRAE [11]	38.95
CleanNet - unsup. base [8]	30.56
Training-ValueNet	**23.66**

TABLE I: Label noise detection results on Clothing 1M in terms of average error across all classes (%).

Image Classification on Clothing 1M					
#	paper	method for noise	init.	training set	accuracy (%)
1	[23]	noisy baseline	ImageNet	1M	68.94
2	ours	noisy baseline	ImageNet	1M	68.88
3	[23]	clean baseline	ImageNet	50K	75.19
Training on noisy 1M only					
4	[3]	loss correct.	ImageNet	1M	69.84
5	[8]	CleanNet	ImageNet	1M	74.69
6	ours	Training-ValueNet	#2	1M	72.03
Training on additional clean 50K					
7	[3]	None	#4	50K	80.38
8	[8]	None	#5	50K	79.90
9	[20]	CurriculumNet	ImageNet	1M + 50K	81.50
10	ours	None	#6	50K	78.06

TABLE II: Image classification results on Clothing 1M in terms of accuracy on test set (%). Results achieved using the additional 50K cleanly labeled images are segregated to avoid confusion with weakly-supervised learning.

D. *Image Classification on Clothing 1M*

In this section we investigate the effectiveness of our method for improving the classification performance on Clothing 1M. Our baseline ResNet model trained on the entire dataset achieves 68.88% accuracy on the test set. We use the Training-ValueNet(s) from the label noise detection experiment to obtain training-value predictions for all 1M noisy training images. We discard images whose predicted training-value falls below the $\delta = 0$ threshold. This leaves us with a cleaned set of 767K images. We fine-tune our baseline model on this cleaned set. Empirically, this yields marginally better performance than if we restart training with ImageNet pre-trained weights as well as being far quicker to train.

Table 2 lists the full results for image classification on clothing 1M. Training-ValueNet improves the accuracy of our classifier from 68.88% to 72.03% (+3.15%). This is notably better than [3] (#4) despite their advantage of using clean labels to estimate confusion between classes. We fall slightly short of CleanNet at 74.69% (#4), however again we stress that our results were achieved *without the use of any manually verified training labels*. We report on a further set of results in Table 2 whereby the manually labeled 50K training images are used for a final round of fine-tuning. This has consistently lead to substantial improvements in performance. Whilst this is not in the true spirit of what we are trying to achieve in this paper (learning without human supervision), we follow suit to provide a full comparison with others.

E. *Image Classification on Aircraft-7*

In this final experiment we evaluate the effectiveness of our method for improving classification performance on the Aircraft-7 dataset which we built using Flickr. The baseline model trained on the entire noisy 75K training set achieves a test accuracy of 82.4%. We proceed to carry out our method, removing a total 20K images whose predicted training-values fall below the $\delta = 0$ threshold. The estimated label accuracy for this cleaned subset is 85.4%, up 15.3% from 70.1% for the entire dataset (see Table 3 for breakdown by class). We fine-tune our baseline model on this cleaned training set and observe a substantial 4.6% improvement in accuracy from 82.4% to 87.0%. This final performance is only slightly short of the

Aircraft-7 Dataset Statistics				
Class Name	Before Label Cleaning		After Label Cleaning	
	# Images	Label acc. (%)	# Images	Label acc. (%)
Airliner	11.0k	87.3	8.7k	99.1
Fighter Jet	8.9k	88.3	7.1k	88.6
Glider	13.1k	53.5	7.6k	77.9
Helicopter	10.8k	74.6	9.5k	85.7
Prop Plane	11.3k	85.1	10.0k	82.9
Sea Plane	14.9k	55.3	8.2k	79.3
Stealth Bomber	4.6k	46.7	2.1k	78.8
Overall	74.6k	70.1	53.2k	85.4

TABLE III: Statistics for the Aircraft-7 dataset which we built using Flickr. Estimates for label accuracy were obtained before and after label noise cleaning by manually annotating a random sample of 500 images per class as either correctly or incorrectly labeled.

Image Classification on Aircraft-7				
#	Method	init.	Training set	accuracy (%)
1	Weakly supervised baseline	ImageNet	noisy 75K	82.4
2	Supervised baseline	ImageNet	clean ImageNet	88.6
3	Training-ValueNet	#1	noisy 75K	87.0

TABLE IV: Image classification results on Aircraft-7 dataset in terms of accuracy on test set (%).

88.6 % accuracy achieved using the fully-supervised training set from ImageNet. Full results are listed in Table 4. Through this experiment we have demonstrated the utility of our method for building large-scale datasets for image classification. Using Training-ValueNet, we were able to collect and clean this dataset in a just matter of hours and have achieved performance on par with fully-supervised learning without manually labeling a single training example.

V. Discussion

A reliable observation in AI research is that general methods leveraging computation and learning are ultimately more effective in the long term than those which rely on human knowledge and heuristic rules. Over the past decade, human knowledge in the form of image labelling and annotation has proved invaluable in the fields of computer vision and beyond. And yet, the associated costs of human labour make fully-supervised learning both undesirable and unsustainable moving forward. Weakly-supervised learning offers a promising alternative but in order to leverage this source of data, we must first develop methods for cleaning these datasets without human supervision.

In recent years, outlier detection has become the leading approach for performing label noise cleaning in this setting. We have demonstrated however that whilst the assumptions of outlier detection seem reasonable in theory, they typically prove overly simplistic and ultimately invalid when dealing with real world datasets. In this paper we have presented Training-ValueNet as a direct response to these concerns and as a first move towards a new class of data driven approaches for label cleaning. Finally, we wish to emphasize that whilst image classification has been the focus of this paper, the approach we have presented can be reasonably be applied to any supervised learning problem.

Acknowledgements

Nicolas Pugeault is supported by the Alan Turing Institute.

References

[1] R. Socher, J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 248–255, 2009.

[2] B. R. Fergus, L. Fei-fei, P. Perona, and A. Zisserman, "Learning Object Categories From Internet Image Searches," *Proc. IEEE*, 2010.

[3] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 2691–2699, 2015.

[4] B. Frénay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 5, pp. 845–869, 2014.

[5] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training Convolutional Networks with Noisy Labels," pp. 1–11, 2014.

[6] A. Drory, S. Avidan, and R. Giryes, "On the Resistance of Neural Nets to Label Noise," *CoRR*, pp. 1–19, 2018.

[7] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop," 2015.

[8] K.-h. Lee, X. He, L. Zhang, and L. Yang, "CleanNet: Transfer Learning for Scalable Image Classifier Training with Label Noise," *CVPR, 2016*, 2016.

[9] H. Lukashevich, S. Nowak, and P. Dunker, "Using one-class SVM outliers detection for verification of collaboratively tagged image training sets," in *Proc. - 2009 IEEE Int. Conf. Multimed. Expo, ICME 2009*, no. August 2009, pp. 682–685, 2009.

[10] W. Liu, G. Hua, and J. R. Smith, "Unsupervised one-class learning for automatic outlier removal," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 3826–3833, 2014.

[11] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1511–1519, 2015.

[12] L. Van Der Maaten and G. Hinton, "Visualizing Data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[13] A. Lapedriza, H. Pirsiavash, Z. Bylinskii, and A. Torralba, "Are all training examples equally valuable?," 2013.

[14] A. Zimek, E. Schubert, and H. P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," 2012.

[15] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, *LOF: Identifying Density-Based Local Outliers*, vol. 29. jun 2000.

[16] Z. G. X. Zhu, "Learning from labeled and un- labeled data with label propagation," *C. CALD-02-107*, 2002.

[17] D. Y. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," *Adv. Neural Inf. Process. Syst. 16*, vol. 16, pp. 321–328, 2004.

[18] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn. - ICML '09*, vol. 2, (New York, New York, USA), pp. 1–8, ACM Press, 2009.

[19] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "MentorNet - Regularizing Very Deep Neural Networks on Corrupted Labels.," *npj Digit. Med.*, 2017.

[20] S. Guo, W. Huang, H. Zhang, C. Zhuang, D. Dong, M. R. Scott, and D. Huang, "CurriculumNet: Weakly supervised learning from large-scale web images," in *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11214 LNCS, pp. 139–154, 2018.

[21] W. Li, L. Wang, W. Li, E. Agustsson, and L. Van Gool, "WebVision Database: Visual Learning and Understanding from Web Data," 2017.

[22] R. S. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting N," *J. Mach. Learn. Res.*, 2014.

[23] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2233–2241, 2017.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *CVPR, 2016*, dec 2016.

[25] A. Botev, G. Lever, and D. Barber, "Nesterov's accelerated gradient and momentum as approximations to regularised update descent," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2017-May, no. 2, pp. 1899–1903, 2017.