On the Labeling Correctness in Computer Vision Datasets

Mohammed Al-Rawi and Dimosthenis Karatzas

Computer Vision Center, Universidad Autonoma de Barcelona, Bellaterra, Spain
al-rawi@cvc.uab.es

Abstract. Image datasets have heavily been used to build computer vision systems. These datasets are either manually or automatically labeled, which is a problem as both labeling methods are prone to errors. To investigate this problem, we use a majority voting ensemble that combines the results from several Convolutional Neural Networks (CNNs). Majority voting ensembles not only enhance the overall performance, but can also be used to estimate the confidence level of each sample. We also examined Softmax as another form to estimate posterior probability. We have designed various experiments with a range of different ensembles built from one or different, or temporal/snapshot CNNs, which have been trained multiple times stochastically. We analyzed CIFAR10, CIFAR100, EMNIST, and SVHN datasets and we found quite a few incorrect labels, both in the training and testing sets. We also present detailed confidence analysis on these datasets and we found that the ensemble is better than the Softmax when used estimate the per-sample confidence. This work thus proposes an approach that can be used to scrutinize and verify the labeling of computer vision datasets, which can later be applied to weakly/semi-supervised learning. We propose a measure, based on the Odds-Ratio, to quantify how many of these incorrectly classified labels are actually incorrectly labeled and how many of these are confusing. The proposed methods are easily scalable to larger datasets, like ImageNet, LSUN and SUN, as each CNN instance is trained for 60 epochs; or even faster, by implementing a temporal (snapshot) ensemble.

Keywords: Data annotation and labeling, ensembles, convolutional neural networks, semi-supervised learning

1 Introduction

Recent developments in deep neural network approaches have greatly advanced the performance of visual recognition systems. Most research and development are based on standard computer vision datasets that have been annotated manually[1] or automatically. Moreover, the computer vision community is devoted to building larger datasets containing tens, or even hundreds, of millions of samples, for example the JFT-300M

[1] Amazon Mechanical Turk; Human intelligence through an API: https://www.mturk.com/

data [1]. Dataset annotation and/or labeling is a difficult, confusing and time consuming task; and even after labeling, it is difficult to assess a dataset for label correctness, whether manually or automatically. One way, however, to verify the labeling is by having a system that returns a confidence-level for each sample in the dataset, and not an overall system/classifier confidence, we illustrate the implementation of our ideas in **Fig. 1**.

Although state-of-the-art deep learning architectures can produce posterior probabilities, these probabilities may not be adequate to estimate the per-sample confidence-level value [2]. However, one promising approach that can be used to measure the per-sample confidence-level is by using ensemble classification methods. In ensemble learning, multiple classifiers can be combined to solve a specific classification task and they can be used to enhance the classification performance by compensating for the low performance of a poor classifier. Other important outcomes of ensemble learning include assigning a confidence-level, and/or posterior probability, to each sample in the testing set. Neural networks ensembles, nonetheless, have been investigated long before deep learning [3, 4]. After the deep learning boom in 2012, there has been quite a few works on ensembles built with deep nets deploying Convolutional Neural Networks (CNNs) [5-7]. Ensembles, in fact, can well be connected with deep learning frameworks and they are currently being used in many research and development aspects, including challenges and competitions [8].

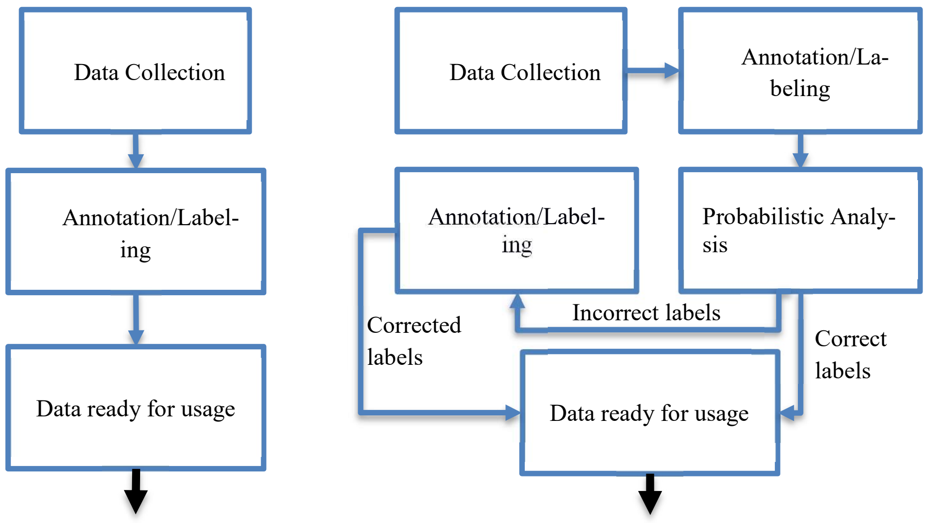


Fig. 1. Data annotation that is normally used (left) and the proposed probabilistic analysis (right).

Ensembles' research work, however, have focused on improving the classification performance, on different applications and not only image understanding, compared to using a single learning model [5, 6, 9-11], and quite a few of them won computer vision challenges, see for example [6]. While the ensemble performance-improvement hypothesis is effective and even supported by theoretical material, confidence analysis has not taken its expected share in the literature. Apart from this, and whenever compared to works tackling the overall confidence-level of the classifier, the confidence should go down to a low-level similar to humans' decision ability to be confident in their classification/decision for each sample/image. Such confidence analysis would highly be useful in weekly supervised learning, which was the goal of the work in [12], where the authors successfully implemented temporal ensembling. However, the authors of [12] have not examined the per-sample confidence levels to perceive how this could be useful in data cleaning, i.e. in a semi-supervised fashion. Other works that differentiate between expert and novice annotators, and between strongly and weakly annotated, in the so called active learning [13-15]. These works usually focus on uncertainty-based methods that usually ignore incorrectly labeled samples, and thus are sensitive to outliers [16, 17]. Furthermore, these works have not incorporated deep CNNs into active learning. The major aim therefore of this work is using deep CNNs to investigate the per-sample confidence level and compare it to the Softmax posterior probability, and to examine the possibility of using it to verify the labeling in computer vision datasets. The proposed approach can also find important application in weekly-supervised / active learning scenarios. We also aim to scrutinize the possibility of building ensemble classifier from one type of CNNs and compare the result to using different types of CNNs, including temporal ensembles, which will allow us to study the independence between the same kind of randomly trained CNN structures.

2 Methods

We tried different types of CNNs that have been trained with ImageNet (aka "Pre-Trained Models"). Generally speaking, a pre-trained model can learn the features from images faster than a model that starts from scratch (i.e. by randomly initializing its weights) [1]. In fact, some pre-trained models can reach an accuracy of 80% in three epochs on CIFAR10. For the ensemble classifier, we implemented voting schemes based on the predicted labels of the used classifiers. It has been proven that majority voting combination will always lead to a performance improvement for sufficiently large number of classifiers provided that the classifier outputs are independent [18]. To illustrate this further, consider a binary classifier and assuming that each classifier has a probability p of making a correct decision, the ensemble's probability (p_{ENS}) of making a correct decision has a binomial distribution [19]:

$$p_{ENS} = \sum_{k=(M/2+1)}^{M} \binom{M}{k} p^k (1-p)^{M-k}, \tag{1}$$

where M is the number of classifiers used to build the ensemble. From the above, if $p > 0.5$, $p_{ENS} \rightarrow 1$ when $M \rightarrow \infty$. Note that $p > 0.5$ (above chance-level) is almost present in most successfully trained binary classifiers. A similar argument can simply be conjectured for multiclass ensembles as combining binary classifiers for multi-class classification is a very familiar approach [20]. The vital issue that can be of concern here is the independence of the output of different classifiers.

2.1 A measure to quantify the classified labels

In this work, we used majority voting ensemble based on the classifiers' output labels, and the ensemble chooses the category/class that receives the largest total vote. The higher the votes each sample gets, the higher the confidence and the lower the votes the lower the confidence. We then used the highest confidence as a key indicator to find any incorrectly labeled samples; which is the per-sample confidence level when the ensemble votes are equal to the number of classifiers used to build it. To make some inferences from the high confidence of the ensemble we make use of 1) N_{inco}, which is the number of incorrect samples that have been classified with high confidence (these are the false positives) with probability $p_{inco} = (N_{inco}/N)$, and 2) N_{corr}, which is the number of correct samples that have been classified with high confidence with probability $p_{corr} = (N_{corr}/N)$, where N is the number of testing samples. The value of p_{inco} is of most interest as it indicates that all classifiers of the ensemble agreed (with high confidence) to incorrectly classify a sample. The high-confidence incorrectly classified samples will further be investigated to verify their labels. It is also possible that these incorrectly classified samples contain some of the difficult / confusing details that deceived the ensemble, or the classifiers that were used to build the ensemble were not independent. To compare the performance of different ensembles, we will calculate the Odds Ratio (OR) using the formula [21]:

$$OR = p_{inco}(1 - p_{corr})/(p_{corr}(1 - p_{inco})). \tag{2}$$

The value of OR will be used to estimate the likeliness that the ensemble may produce false positives but with high confidence (on the assumption that all samples are correctly labeled/annotated), i.e. how likely the incorrect samples will be classified as correct ones with high confidence; hence, the lower the OR the better. An OR equals to one indicates that the classification of correct and incorrect samples with high confidence is equally likely to occur.

We used CIFAR10 and CIFAR100 [22] in all confidence analyses' experiments. CIFAR10 is a well-known dataset that has heavily been investigated in the computer vision literature. It essentially has 50K 32×32 RGB image samples for training and 10K 32×32 RGB image samples for testing, where each image belongs to one of ten classes; airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Each class thus has 5000 images in the training set and 1000 images in the testing set. CIFAR100, on the other hand, has a similar image structure but it has 100 classes distributed on 600 samples and 500 samples in the training set and 100 samples in the testing set. To implement our algorithms, we used PyTorch [23] as our main deep learning framework. Further details on the used methods and experimental setting can be found in the supplemental material.

We chose the VGG CNN family [24] (we will refer to VGG Ensemble; 'VGG-E') as they require less training time than other CNNs, and they can reach higher accuracy than other CNNs, when trained up to 60 epochs. We chose 60 epochs for the following reasons: 1) to see how fast and how well the ensemble classifier can learn with confidence 2) to reduce the execution time of ensembles, 3) following the Schapire's idea on the strength of weak learnability [25], and 4) for the proposed methodology to be efficient and scalable when used on larger data sets. Some VGGs have Batch Normalization (BN) others do not, thus they have been postfixed with 'BN', VGG11BN thus denotes VGG11 with batch normalization. In most analysis, we used eight VGG CNNs, these are: VGG11, VGG11BN, VGG13, VGG13BN, VGG16, VGG16BN, VGG19, VGG19BN.

3 Results

3.1 The Softmax Posterior Probability

It is widely known that the CNNs, and neural networks in general, yield Posterior Probabilities (PPs) as their outputs, when Softmax is used. It is not known, however, if these posterior probabilities can be used to estimate the per-sample confidence to an accurate degree. To investigate the confidence distribution of the correctly and incorrectly classified labels via Softmax outputs, we used CIFAR100 to train VGG19BN with the previously mentioned settings except that we increased the number of training epochs to 600. We will refer to the condition where Softmax posterior probability equals one as high confidence. The typical situation of the PP of the incorrectly classified samples is to have an exponential distribution or, in the worst case, a normal distribution. The results of the training and testing are demonstrated in **Table 1** and show that Softmax posterior probabilities of a single VGG have high OR values, and thus, may not be used as good estimates of the per-sample confidence level. This deduction is clearly depicted in **Fig. 2** that shows the posterior probability distributions, where the incorrectly classified samples have a peek at PP=1 (PP=1 denotes high confidence as the probability is 100%). We also perceive from **Fig. 2** that the distribution of the PP values is right-skewed for the incorrect labels, and this means using these PP values for the per-sample confidence level is not reliable. The presented Softmax results, in fact, copes with the neural networks posterior probabilities as being over-confidence estimates, as has been detailed in [2]. Our probability analysis provides further evidence of why adversarial attacks [26] are possible when using Softmax to state the confidence of the classified object/image.

3.2 Ensembles Built with Different VGG Types

Using CIFAR10, each VGG type was trained for up to 16 times, VGG-E thus has a total of 118 VGGs (Skipping chance-level local minima resulted sometimes in less than the planned 16×8 = 128 VGGs). The results of the discovered images with incorrect labels in the testing set are presented in **Table 2**. Our tests showed that there are a 9 incorrect samples with high-confidence (voting is equal to the number of VGGs used to build the ensemble). Investigating the 9 false positives, we found that most of them have incorrectly been labeled in the testing set of CIFAR10. We also present in the supplementary material a few samples that have high per-sample confidence level values but were incorrectly classified. After examination, however, these samples appear to be confusing.

Table 1. VGG19BN output as posterior probability computed with Softmax, trained for 600 epochs.

		Training set size	**Testing set size**	**Classification accuracy%**	**#Incorrectly classified at PP=1**	**#Correctly classified at PP=1**	**OR**
CIFAR10		50K	10K	93.0	248	8702	0.004
		10K	50K	86.2	2957	39419	0.017
CIFAR100		50K	10K	72.8	377	5072	0.038
		10K	50K	58.8	2295	16733	0.095

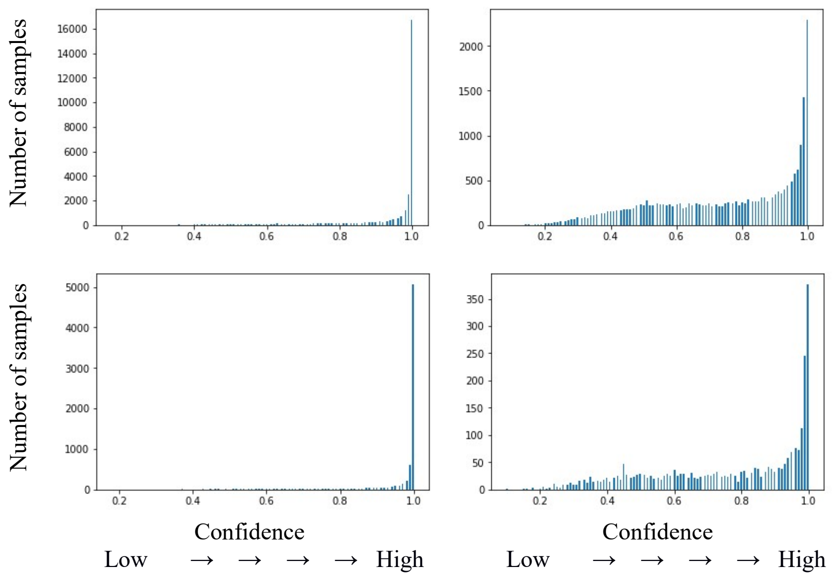
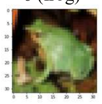
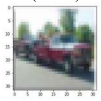
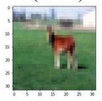
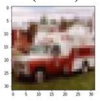
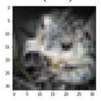
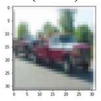
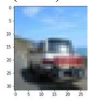


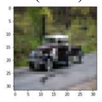
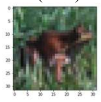
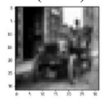
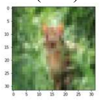
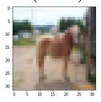
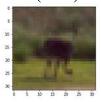
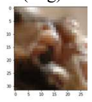
Fig. 2. VGG19BN Softmax posterior probability distribution of the correctly classified labels (left column) and the incorrectly classified labels (right column); training with 10K and testing with 50K samples of CIFAR100 (top row) and training with 50K and testing with 10K samples of CIFAR100 (bottom row).

To investigate correctness of the training set labels, we revert the training and testing datasets that are used to train the VGGs. In this case, we used CIFAR10 testing set (10K samples) for building the ensemble classifier and CIFAR10 training set (50K) for testing it. After training, the VGG-E has a total of 126 different VGGs. This, in principle, is more challenging than the previous experiment, and could be useful in weakly supervised learning. The results of the discovered incorrect labels are presented in **Table 2**. By investigating the 81 false positives, we found that some have incorrectly been labeled in the training set of CIFAR10, but some images have confusing content. A few samples discovered by the VGG-E are not only be confusing CNNs, but also to the human observer, see the supplementary material. We repeated the same above experiments on Cifar100, which resulted in a VGG-E with 128 VGGs. The ensemble enhanced the accuracy by ~9% compared to the average of VGGs. A few incorrect labels in CIFAR100's testing, as well in the training set, are demonstrated in supplemental material. The analysis of these ensembles are summarized in **Table 3**, and the per-sample confidence distributions are shown in **Fig. 3**.

We can see from **Table 2** that the frog (which has an index 2405 in the data) is labeled as a cat in CIFAR10 testing set, but the VGG-E managed to predict the correct label (more results are shown in the supplemental material). The amount of incorrect labels in CIFAR100 is higher, for example, a bottle (which has an index of 7762) is labeled as a cup, other images with incorrect labeling also exist. Nonetheless, by inspecting these images, one can admire the work that CNNs can achieve in classifying these CIFAR images, as most of the times the details are not clear even for the human observer, due to using the so called tiny images (as each image has a size of 32×32). Thus, using CNNs ensembles would assist inspecting and verifying the labeling, as proposed in this work.

Table 2. Some Incorrect Labels Discovered by VGG-E in CIFAR10.

Testing set	2405*
	3 (cat)
	6 (frog)
	
		9227
		1 (car)
		9(truck)
		
			3309
			4 (deer)
			7(horse)
			
				3560
				1 (car)
				9(truck)
				
					5141
					6(frog)
					3 (cat)
					
						9227
						1(car)
						9(truck)
						
							7311
							1 (car)
							9 (truck)
							

Training set	5747
	1 (car)
	9 (truck)
	
		35103
		6 (frog)
		4 (deer)
		
			30814
			1 (car)
			9 (truck)
			
				6319
				3 (cat)
				4 (deer)
				
					33079
					5 (dog)
					7 (horse)
					
						7008
						2 (bird)
						4 (deer)
						
							8803
							3(cat)
							5(dog)
							

* Index (The Index of the image in CIFAR10); Original label; Predicted label; and Image of Predicted label.

3.3 Experiments with the EMNIST dataset

The same experimental strategy used for CIFAR10&100 has been implemented on the EMNIST dataset [27]. The EMNIST dataset, which is derived from the NIST Special Database, has been compiled from a set of handwritten English characters and Arabic digits and has been suggested as a more challenging replacement to the MNIST dataset. The EMNIST 'By Class' split has 814,255 images distributed over 62 unbalanced classes. Pixel image format and dataset structure that directly matches the MNIST dataset, each image is 28x28 gray-level. EMNIST is extremely challenging, on the labeling and testing levels, as it has upper and lower case confusion, in addition to numeral value one (1) versus letter (lower case L; l), O versus 0/zero, 9 versus q, etc. In fact, our analysis shows that this confusion has been present at the labeling/annotation stage.

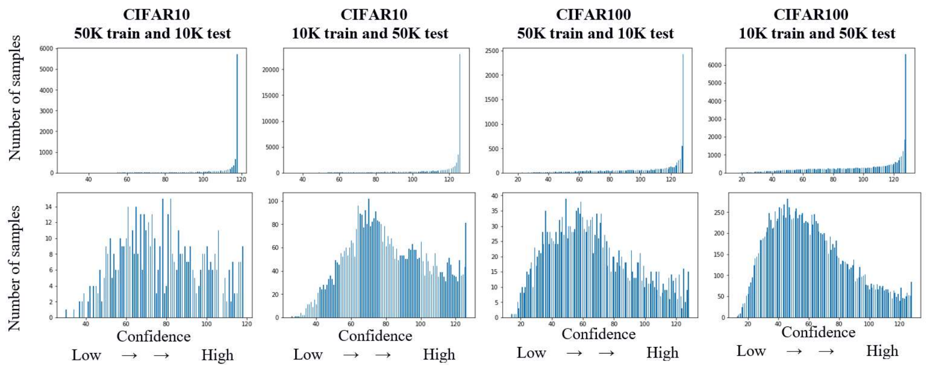


Fig. 3. Per-sample confidence distribution, correctly classified labels (top row) and incorrectly classified labels (bottom row) in CIFAR10 and CIFAR100.

Table 3. VGG-E using 16 classifier instances of each VGG type, a total of 128 VGGs to build the VGG-E.

	Training set size	**Testing set size**	**Average accuracy over VGGs %**	**Accuracy of VGG-E %**	**#Confidently classified, but incorrect**	**#Confidently classified and correct**	**OR**
CIFAR10	50K	10K	90.84 ± 0.95	94.2	9	5734	0.0006
	10K	50K	86.10 ± 1.07	90.2	81	23024	0.002
CIFAR100	50K	10K	69.5 ± 0.14	78.2	15	2446	0.005
	10K	50K	57.06±1.5	67.06	84	6687	0.01

As for the results, the ensemble gave a classification accuracy 0.87 when trained using the training set. Furthermore, in the testing set, the incorrectly labeled samples that got recognized by the ensemble, with high confidence, is 2,837, while the correct samples that got recognized by the ensemble, with high confidence, is 83,467, and the quantitative measure OR is 0.0098. To inspect the labeling of the training set, we trained the ensemble with the testing, which gave classification accuracy of 0.85. The number of incorrect samples that got recognized by ensemble with confidence is 9,154, the correct samples that got recognized by the ensemble with high confidence is 408,588, and the quantitative measure OR is 0.0094. The confidence distributions are illustrated in **Fig. 4**. Due to space limitations, incorrect/confusing EMNIST images are demonstrated in the supplemental material.

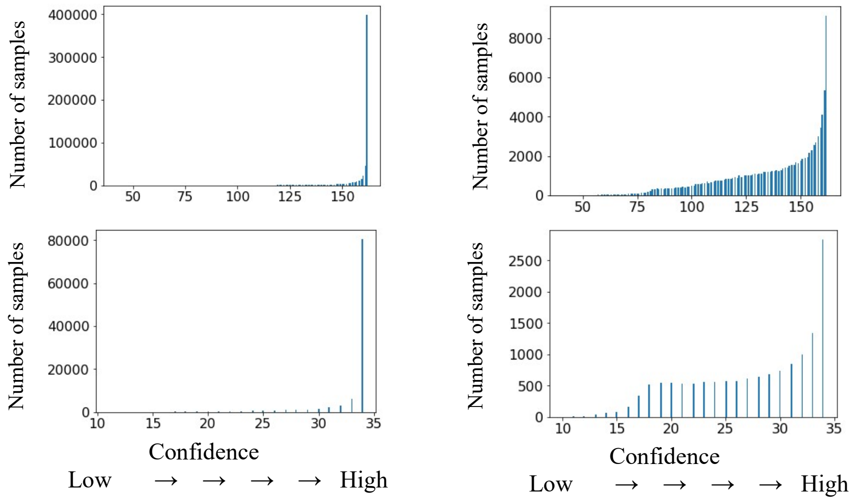


Fig. 4. Per-sample confidence distribution training set (top row) and testing set (bottom row); correctly classified labels (left) and incorrectly classified labels (right) in EMNIST dataset.

3.4 Experiments with the SVHN dataset

SVHN [28] is a real-world image digit dataset that has been inspired by MNIST structure (e.g., the images are of small cropped digits) but comes from a significantly harder, unsolved, real world problem (recognizing digits and numbers in natural scene images). SVHN, which contains 73257 digits for training and 26032 digits for testing, has been obtained from house numbers in Google Street View images. Using the training set for training, the classification accuracy of the ensemble is 95.14. The number of incorrect samples that got recognized by all the CNNs is 129 and the number of correctly-labeled samples that got recognized by the ensemble is 20887, yielding OR= 0.0012. Inspecting the label correctness in the training set showed that the number of incorrect labels that got recognized by the ensemble with high confidence is 267 with OR= 0.0018 (classification accuracy of the ensemble is 93.69). The confidence distributions are illustrated in **Fig. 5**. Due to space limitations, selected incorrect/confusing SVHN images that have been detected with our approach are demonstrated in the supplemental material.

4 Conclusion

It is of high interest in computer vision to have a system that can conjecture with confidence what is wrong and what is right, i.e., to confidently guess which labels are correctly and/or incorrectly classified. This work is a step in that direction. This paper presents the use of CNN ensembles to detect incorrect labels in image classification datasets. Essentially, if the ensemble is confident on a result which is incorrect, either the sample is indeed visually confusing or it was incorrectly labelled. Probabilistic confidence analyses showed that some images with incorrect labeling and confusing content exist. **Fig. 6** summarizes the results of CIFAR10 & 100 and illustrates that the OR values of Softmax posterior probabilities are higher than the OR values of the ensemble posterior probabilities; the lower the OR values the better. Hence, the posterior probability of a CNN, measured with Softmax, cannot be used to accurately estimate the per-sample confidence level. Furthermore, the proposed OR analysis provided a novel evidence that batch normalization increases the ensemble confidence, thus, could be related to improving generalization.

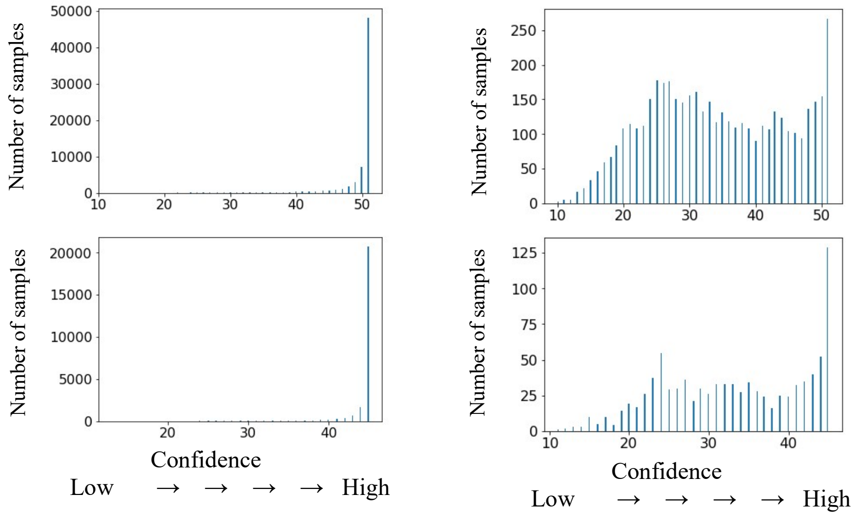


Fig. 5. Per-sample confidence distribution training set (top row) and testing set (bottom row); correctly classified labels (left) and incorrectly classified labels (right) in SVHN dataset

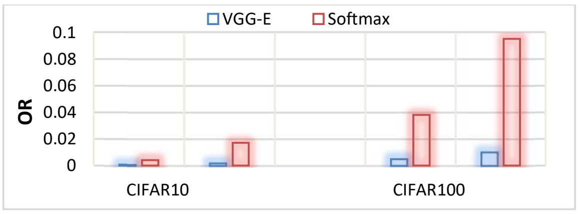


Fig. 6. VGG-E (ensemble) versus Softmax posterior probability.

Our analyses also agreed with previous ensemble works as the overall accuracy has been increased by around 5%, 9%, 2%, 5% for CIFAR10, CIFAR100, SVHN, and EMNIST respectively. Based on the proposed probabilistic methods and by making use of the snapshot ensemble (supplemental material), we are currently building a labeling verification tool to be implemented in PyTorch framework. This tool will be useful not only in labeling verification, but can also be used in semi-supervised and active learning applications. Evaluations on other datasets are left for future work.

References

1. Sun, C., et al. *Revisiting Unreasonable Effectiveness of Data in Deep Learning Era*. in *16th IEEE International Conference on Computer Vision (ICCV)*. 2017. Venice, ITALY.
2. Ju, C., A. Bibaut, and M. J. van der Laan, *The Relative Performance of Ensemble Methods with Deep Convolutional Neural Networks for Image Classification*. 2017, http://arxiv.org/abs/1704.01664.
3. Hansen, L.K. and P. Salamon, *Neural Network Ensembles.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990. **12**(10): p. 993-1001.
4. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* Nature, 2015. **521**(7553): p. 436-444.
5. Chen, J.L., et al. *An Ensemble of Convolutional Neural Networks for Image Classification Based on LSTM*. in *2017 International Conference on Green Informatics (ICGI)*. 2017.
6. Ding, C.X. and D.C. Tao, *Trunk-Branch Ensemble Convolutional Neural Networks for Video-Based Face Recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018. **40**(4): p. 1002-1014.
7. Schmidhuber, J., *Deep learning in neural networks: An overview.* Neural Networks, 2015. **61**: p. 85-117.
8. Minetto, R., M. Pamplona Segundo, and S. Sarkar. *Hydra: an Ensemble of Convolutional Neural Networks for Geospatial Land Classification*. in *https://arxiv.org/abs/1802.03518*. 2018.
9. Chen, G.B., et al. *Ensemble Application of Convolutional and Recurrent Neural Networks for Multi-label Text Categorization*. in *International Joint Conference on Neural Networks (IJCNN)*. 2017. Anchorage, AK.
10. Duan, M.X., K.L. Li, and K.Q. Li, *An Ensemble CNN2ELM for Age Estimation.* IEEE Transactions on Information Forensics and Security, 2018. **13**(3): p. 758-772.
11. Díaz-Vico, D., et al., *Deep Neural Networks for Wind and Solar Energy Prediction.* Neural Processing Letters, 2017. **46**(3): p. 829-844.
12. Laine, S. and T. Aila, *Temporal Ensembling for Semi-Supervised Learning*, in *International Conference on Learning Representations (ICLR)*. 2017.
13. Yang, Y.Z. and M. Loog, *A variance maximization criterion for active learning.* Pattern Recognition, 2018. **78**: p. 358-370.
14. Reyes, O., A.H. Altalhi, and S. Ventura, *Statistical comparisons of active learning strategies over multiple datasets.* Knowledge-Based Systems, 2018. **145**: p. 274-288.
15. Wang, K.Z., et al., *Cost-Effective Active Learning for Deep Image Classification.* IEEE Transactions on Circuits and Systems for Video Technology, 2017. **27**(12): p. 2591-2600.
16. Freund, Y., et al., *Selective sampling using the query by committee algorithm.* Machine Learning, 1997. **28**(2-3): p. 133-168.
17. Bujrbidge, R., J.J. Rowland, and R.D. King, *Active learning for regression based on query by committee.* Intelligent Data Engineering and Automated Learning - Ideal 2007, 2007. **4881**: p. 209-218.
18. Bishop, C.M., *Pattern Recognition and Machine Learning (Information Science and Statistics)*. 2006: Springer-Verlag New York, Inc. .

19. Monteith, K., et al. *Turning Bayesian Model Averaging Into Bayesian Model Combination*. in *International Joint Conference on Neural Networks (IJCNN)*. 2011.
20. Shiraishi, Y. and K. Fukumizu, *Statistical approaches to combining binary classifiers for multi-class classification.* Neurocomputing, 2011. **74**(5): p. 680-688.
21. Rao, P.S., *Proportions, Odds Ratios and Relative Risks*, in *Statistical Methodologies with Medical Applications*. 2017, Wiley.
22. Krizhevsky, A., *Learning Multiple Layers of Features from Tiny Images, Technical Report*. 2009, http://www.cs.toronto.edu/~kriz/cifar.html: Canadian Institute for Advanced Research.
23. Paszke, A., et al. *Automatic differentiation in PyTorch*. in *NIPS 2017 Autodiff Workshop*. 2017.
24. Simonyan, K. and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. in *International Conference on Learning Representations*. 2015.
25. Schapire, R.E., *The Strength of Weak Learnability.* Machine Learning, 1990. **5**(2): p. 197-227.
26. Li, X. and F.X. Li. *Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics*. in *16th IEEE International Conference on Computer Vision (ICCV)*. 2017. Venice, ITALY.
27. Cohen, G., et al., *EMNIST: an extension of MNIST to handwritten letters*. (2017): https://www.nist.gov/itl/iad/image-group/emnist-dataset http://arxiv.org/abs/1702.05373.
28. Netzer, Y., et al., *Reading Digits in Natural Images with Unsupervised Feature Learning*, in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. 2011: http://ufldl.stanford.edu/housenumbers/.

Supplemental Material

Experimental Setting

The following parameters have been used in all experiments: dropout probability is 0.2; maximum number of epochs is 60; learning rate is 0.01 (the learning rate is set to decrease by half according to the following milestones = {8, 20, 48); unless mentioned otherwise); momentum=0.95; the seed was randomly pulled from the time function; weight-decay=0.0005; Nestrove momentum was used; SGD optimizer; 100 mini batches, random shuffling enabled, and Cross Entropy Loss. The run/training, however, was skipped if the CNN is stuck at a chance-level local minima (~10% for CIFAR10 and ~1% for CIFAR100), and a new training instance is launched with a new random seed. To demonstrate the possible variations in training each CNN of the ensemble, we present the training progress of various VGGs in Fig-Sup. 1.

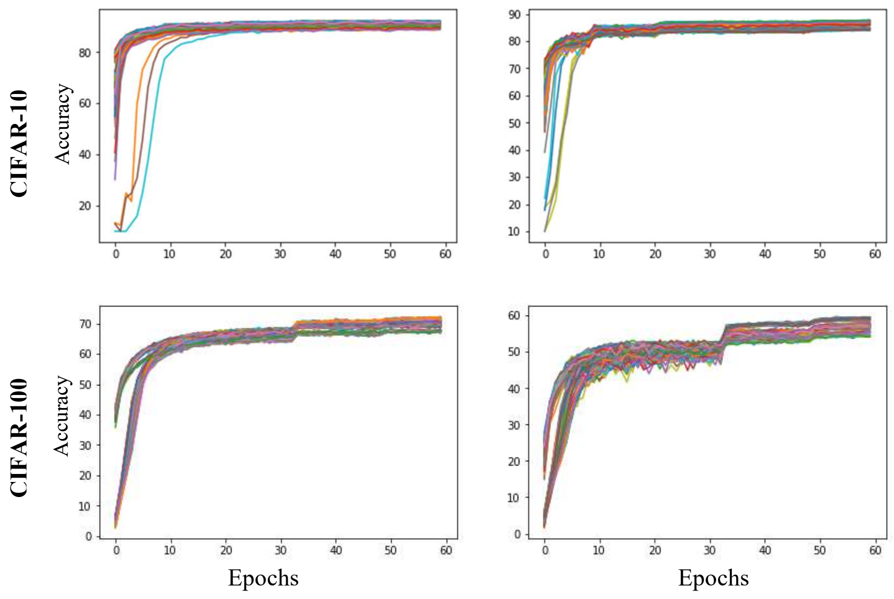


Fig-Sup. 1. Variations of the training progress of the different ensembles built from CIFAR10 and CIFAR100 data. Training with 50K and testing with 10K (left column) and training with 10K and testing with 50K (right column).

In our preliminary analysis, we built ensembles using different CNN architectures; including, different types of ResNets*, VGGs* DualPathNets* (DPNs), DenseNets*, NasNetLarge, etc. However, we chose to build the ensembles via the VGG net family as they require less training time than the other CNN types, they give similar performance to the ensemble built from different CNN architectures, and they result much higher accuracy than other CNNs, when trained up to 60 epochs. To give an example, NasNetLarge requires 9X times the training time of VGG11 and 5X times of VGG19BN. The classification accuracy of NasNetLarge gets to 75% compared to above 85% for all VGG types, when trained up to 60 epochs. To clarify further, for a maximum of 60 epochs, VGG11 reaches 85% accuracy in less than 6 minutes, while ResNet18 gets to 78% accuracy in 7 minutes, but NasNetLarge gets to 75% accuracy in 71 minutes. In general, the DPN, SqueezNet, and ResNet (including Resnext*) families are slower than VGGs and/or can get less than 80% accuracy in 60 epochs.

Ensembles Built with a Single VGG Type

In this experiment, we used 16 classifier instances to see how do they perform compared to using different VGG classifiers. The training has been performed with the 10K testing set, and the testing has been performed using the 50 training set, as it is more challenging than using the sets in training the other way around. Table 4 summarizes the results. From Table 4 we notice that Batch-Normalization always leads to better confidence, when the same CNN is used, as the OR is less when using BN, that is it is less likely to have false positives with high confidence when BN is used.

Table 4. VGG-E using 16 classifier instances of each VGG type; using 10K for training and 50K for testing of CIFAR10.

VGG-E	Average accuracy over VGGs %	Accuracy of VGG-E %	#Confidently classified, but incorrect	#Confidently classified and correct	OR
VGG19BN	86.78 ± 0.11	90	424	33073	0.0043
VGG16BN	87.48 ± 0.15	91	446	33892	0.0042
VGG13BN	86.95 ± 0.11	90	466	33559	0.0046
VGG11BN	84.79 ± 0.12	88	556	31506	0.0066
VGG19	86.27 ± 0.33	89	607	33657	0.0059
VGG16	86.31 ± 0.22	89	651	34120	0.0061
VGG13	86.09 ± 0.13	89	706	34156	0.0076
VGG11	84.11 ± 0.15	87	834	32379	0.0092

From Table 4 we see that VGG11 has the weakest performance compared to other VGGs, thus, we took this experiment further to build one VGG-E using 128 VGG11s and another using 128 VGG13BN using CIFAR10 testing set for training. The VGG-E increased the performance by ~5%., but the confidence levels, as shown in the OR values, are better when using different VGG models than using only one VGG model, as shown in Table 5. Thus, a VGG-E built using 128 VGGs results, given by the OR values, are not as good as a VGG-E built using different types of VGGs.

Table 5. VGG-E with 126 VGGs of the same type (CIFAR10); using 10K for training and 50K for testing.

	Mean accuracy over VGGs %	**Accuracy of VGG-E %**	**#Confidently classified, but incorrect**	**#Confidently classified and correct**	**OR**
VGG13BN	87.50 ± 0.13	90.9	116	26569	0.0020
VGG11	87.48 ± 0.15	87.3	229	25277	0.0045

Temporal (snapshot) Ensemble (VGG-ET)

We used VGG19BN to build a temporal ensemble for CIFAR100; training with 50K and testing with 10K. In this case, each epoch resulted a classifier. We used 150 epochs and neglected the results of the first ten epochs, as we opted for the training to reach a state of stability. Similar to VGG-E, VGG-ET was able to determine quite a few incorrect labels, and to produce descent per-sample confidence values. The VGG-ET reached an accuracy of 76.8% (slightly lower than VGG-E), and an OR (at high confidence) of 0.004. Thus, this snapshot/temporal ensemble could be used instead of an ensemble built from the different CNN architectures, which can be used to build a fast and efficient labeling verification tool, which is a future work we are trying. The confidence distributions are demonstrated in Fig-Sup. 2.

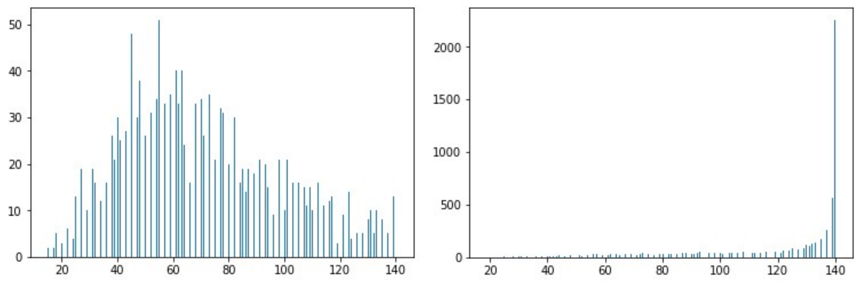
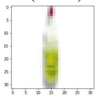
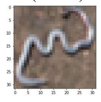
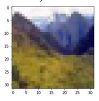
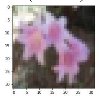
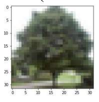
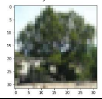
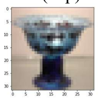
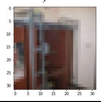
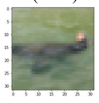
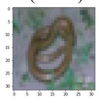
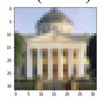
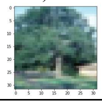
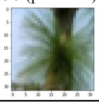
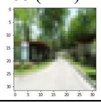
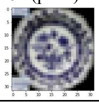
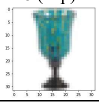
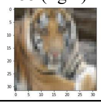
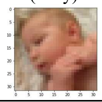
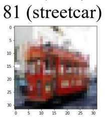
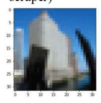
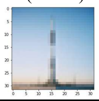
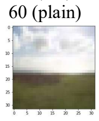
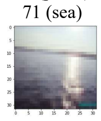
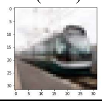


Fig-Sup. 2. Per-sample confidence distribution, incorrectly classified labels (left) and correctly classified labels (right), using temporal (snapshot) ensemble VGG-ET (VGG19BN).

Extended Results

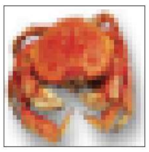
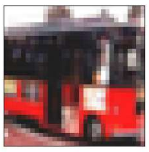
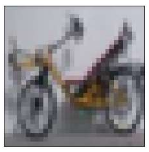
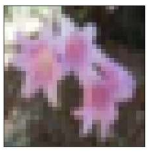
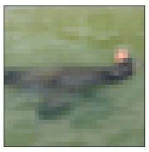
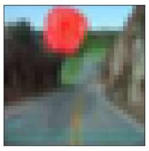
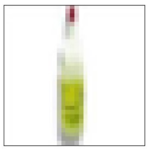
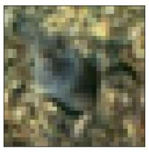
In the tables below, we demonstrate using a few samples ***that we have selected from the incorrect labeled ones detected by the probability analysis***. The labels of the samples and the corresponding predicted labels, along with the corresponding image, are shown. To double check the incorrectness, by third parties, the readers of this article may use the index of the sample to examine it in the dataset, i.e. by loading the image and the corresponding label. The tables contain data from CIFAR10, CIFAR100, SVHN, and EMNIST datasets.

Table 6. Selected incorrectly labeled images detected in CIFAR100 training set; predicted with high confidence

Testing set	7762 28 (cup) 9 (bottle) 	5764 99 (worm) 78 (snake) 	9601 33 (forest) 49 (moun-tain) 	6927 92 (tulip) 54 (orchid) 	8951 47 (maple tree) 52 (oak tree) 	4460 59 (pine tree) 52 (oak tree) 
	1557 10 (bowl) 28 (cup) 	8071 5 (bed) 94 (ward-robe) 	1100* 72 (seal) 55 (otter) 	2172 99 (worm) 78 (snake) 	9298 17 (castle) 37 (house) 	1357 96 (willow tree) 52 (oak tree) 
Training Set	687 59 (pine tree) 56 (palm tree) 	780 37 (house) 68 (road) 	7006 10 (bowl) 61 (plate) 	12455 40 (lamp) 28 (cup) 	10178 42 (leopard) 88 (tiger) 	39441 11 (boy) 2 (baby) 
	32814 90 (train) 81 (streetcar) 	47936 12 (bridge) 76 (sky-scraper) 	35209 76 (sky-scraper) 69 (rocket) 	23509 71(sea) 60 (plain) 	16305 60 (plane) 71 (sea) 	22395 81 (street-car) 90 (train) 

* We found the same image in the training test with index 24083 but has the label 55 (otter). So not only the same image was included in both training and testing sets, but with an incorrect/opposite label.

Table 7. Selected incorrectly labeled images detected in CIFAR100 testing set; predicted with high confidence

Index	Original label (class)	Predicted label (class)	Image	Remarks
1214	45 (lobster)	26(crab)		
5278	81(streetcar)	13(bus)		
6799	8(bicycle)	48(motorcycle)		
6927	92(tulip)	54(orchid)		
1100	72(seal)	55(otter)		
7429	70(rose)	68(road)		
7762	28(cup)	9(bottle)		
5873	50(mouse)	74(shrew)		

Confusing Images Detected in CIFAR10

Table 8. Selected confusing images detected in CIFAR10 test set; predicted with high confidence

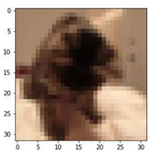
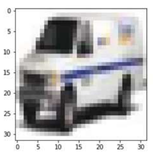
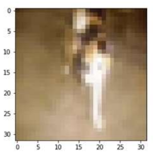
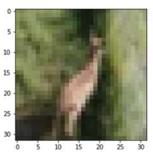
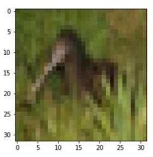
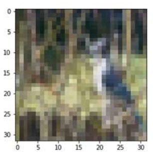
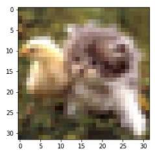
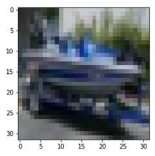
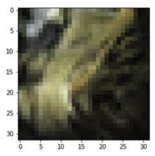
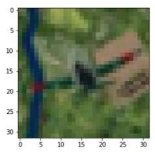
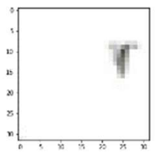
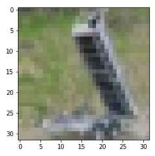
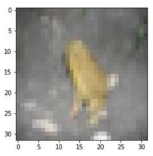
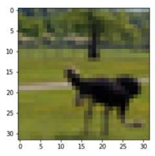
Index	Original label (class)	Predicted label (class)	Image	Remarks
811	3 (cat)	5 (dog)		Hard to tell what this is!
5416	9 (truck)	1 (car)		This is a minivan, probably looks more like a car than a truck
7099	3 (cat)	5 (dog)		Probably the label is correct, but the tail is dominating the photo
4794	4 (deer)	2 (bird)		Difficult to infer which one is deer and which one is bird, even for the human observer
9832	2 (bird)	4 (deer)		Difficult to infer which one is deer and which one is bird, even for the human observer
9503	2 (bird)	4 (deer)		Difficult to infer which one is deer and which one is bird, even for the human observer

Table 9. Selected confusing images detected in CIFAR10 training set; high confidence pred

Index	Original label (class)	Predicted label (class)	Image	Remarks
32085	3 (cat)	5 (dog)		Bird and cat in one picture! Classified as dog
13694	8 (ship)	1 (car)		Boat on top of a pull cart with wheels identified as car
43283	3 (cat)	6 (frog)		The image is not clear, probably of cat category, but classified as frog
9119	0 (plane)	4 (deer)		The image should be of category plane, yet it is not clear; classified as deer
5867	2 (bird)	0 (plane)		A very confusing object
36288	9 (truck)	2 (bird)		A truck with a ladder is hard to identify as a truck
36788	6 (frog)	3 (cat)		A frog that is hard to tell for the human observer
34305	2 (bird)	4 (deer)		Something that does not look very much like a bird has been identified as deer

Experiments with SVHN

Table 10. Selected incorrectly labeled images detected in SVHN training set; predicted with high confidence

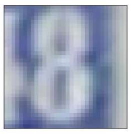
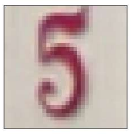
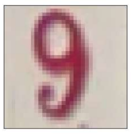
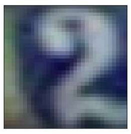
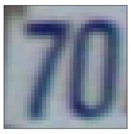
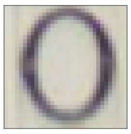
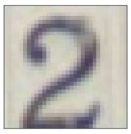
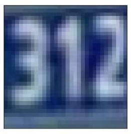
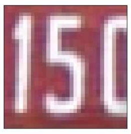
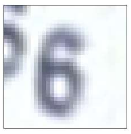
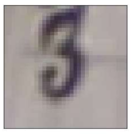
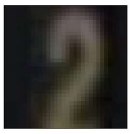
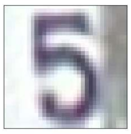
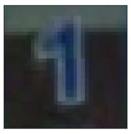
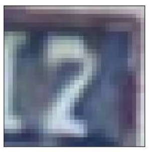
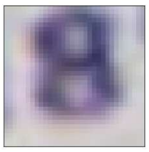
Index	Original label (class)	Predicted label (class)	Image	Remarks
3692	9	8		
6290	9	5		
6291	5	9		
6875	1	2		
6960	0	7		This image has two digits, although there should only be one. It was labeled with zero, but the ensemble got the other digit correctly with high confidence (7)
9502	2	0		
9503	0	1		
17666	3	1		As each image should only have one digit, this image has been incorrectly segmented and labeled with 3, the ensemble labeled it as 1 with high confidence.

Table 11. Selected incorrectly labeled images detected in SVHN testing set; predicted with high confidence

Index	Original label (class)	Predicted label (class)	Image	Remarks
1317	1	5		The image is incorrectly segmented, as it should have only one digit. Interestingly, the attention of the CNN is brought to the center
3916	1	6		Partially occluded with 5, but the ensemble got it correctly as 6. The original labels was incorrect with a value of 1.
5397	0	3		
6500	1	2		
8250	1	5		
11844	5	1		
14678	7	2		
14526	1	8		

Experiments with EMNIST

Table 12. Selected incorrectly labeled images detected in EMNIST training set; predicted with high confidence

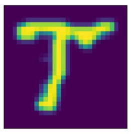
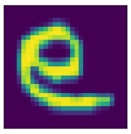
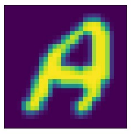
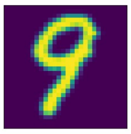
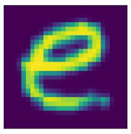
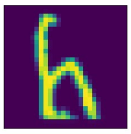
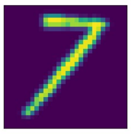
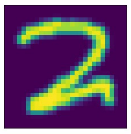
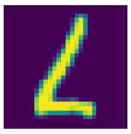
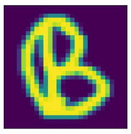
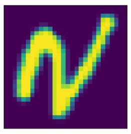
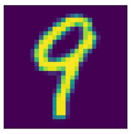
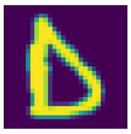
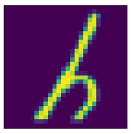
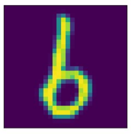
Index	Original label (class)	Predicted label (class)	Image	Remarks
104	t	T		
13980	F	e		
14283	a	A		
15830	g	9		
18892	r	e		
19781	6	h		
21248	T	7		
29785	z	2		

Table 13. Selected incorrectly labeled images detected in EMNIST testing set; predicted with high confidence

Index	Original label (class)	Predicted label (class)	Image	Remarks
2371	(lower case L; l)	L		
2202	b	B		
2127	n	N		
2618	g	9		
2640	B	D		
2820	b	h		
3369	6	b		
3515	r	P