# An improved method of identifying mislabeled instance and the mislabeled instances in MNIST and CIFAR-10

**Xinbin Zhang**
Sydney Machine Learning Study Group
abner.zhangxinbin.anz@gmail.com

## Abstract

The quality of the dataset plays an important role in machine learning. Many mislabeled data detection techniques have been proposed; however, there are few similar works done in deep learning and there is no mislabeled instance being reported in MNIST and CIFAR-10 datasets, whose accuracy is an important criterion for algorithms in machine learning community. In this paper I develop an improved method to identify mislabeled instance and find 675 mislabeled instances in MNIST, 118 mislabeled instances in CIFAR-10, which proves that the improved method is efficient and could apply on other datasets to improve the quality of the dataset.

## 1 Introduction

The reliably labeled data are often expensive and time consuming to obtain. In real-world there are always noise labels including mislabeled instances in datasets.

There are two standardized image classification datasets MNIST and CIFAR-10 that are widely used and regarded as clean datasets. And these two datasets are important because machine learning community uses them to record holistic approach by assessing and characterizing the performance of an algorithm which is shown table 1 and table 2 and it is called the state-of-the-art.

Table1 the state of the art of MNIST

| Result | Method | Venue |
|---|---|---|
| 0.21% | Regularization of Neural Networks using DropConnect | ICML 2013 |
| 0.23% | Multi-column Deep Neural Networks for Image Classification | CVPR 2012 |
| 0.23% | APAC: Augmented PAttern Classification with Neural Networks | arXiv 2015 |
| 0.24% | Batch-normalized Maxout Network in Network | arXiv 2015 |
| 0.29% | Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree | AISTATS 2016 |
| 0.31% | Recurrent Convolutional Neural Network for Object Recognition | CVPR 2015 |

Table 2 the state of the art of CIFAR-10

| Result | Method | Venue |
|---|---|---|
| 96.53% | Fractional Max-Pooling | arXiv 2015 |
| 95.59% | Striving for Simplicity: The All Convolutional Net | ICLR 2015 |
| 94.16% | All you need is a good init | ICLR 2016 |
| 93.95% | Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree | AISTATS 2016 |
| 93.72% | Spatially-sparse convolutional neural networks | arXiv 2014 |
| 93.63% | Scalable Bayesian Optimization Using Deep Neural Networks | ICML 2015 |

The result is based on the assumption that all labels in test dataset are correct. In this paper, I develop an improved method to identify the mislabeled instance and apply it to MNIST and CIFAR-10. The result shows that such assumption is wrong, there are some mislabeled instances in test dataset, which could change the state-of-the-art. It also shows that after correcting mislabeled instances in training dataset, the accuracy increases.

This paper is organized as follows. Section 2 discusses related work on identifying mislabeled data. Section 3 introduces the improved method. Section 4 shows the mislabeled data in MNIST including the types of error, the of correcting data and the result of correcting data. Section 5 shows the mislabeled data in CIFAR-10 including the types of error, the of correcting data and the result of correcting data. Section 6 concludes this paper.

## 2 Related Work

The most used method of detecting mislabeled instance is done by [Brodley *et al.*, 1999]. He use a set of different classifiers that serve as noise filters for the training data, which includes majority filtering and consensus filtering by constructing a set of base-level classifiers and then using their classifications to identify mislabeled instances in ensemble method.

[Muhlenbach *et al.*, 2004] proposes a filtering algorithm by removing or relabeling the suspect samples before the learning stage.

In order to improve the quality of the training data and to reduce the overlapping among regions of different classes [Sanchez *et al.*, 2003] proposes several methods, based on the nearest neighbor classifiers.

The second approach is detecting the mislabeled samples as outliers. In general, a mislabeled sample need not be outlying, and an outlier is not necessarily mislabeled. Outlier detection for high-dimensional data has received a lot of attention in recent years. For example, [Aggarwal *et al.*, 2001] studies the problem of outlier detection for high-dimensional data using projections into subspaces. But this approach is clearly not scalable for large data set.

The third approach is to detect outliers with the use a distance-based, which one has to define a distance function between samples. In a high-dimensional space, it is often difficult to do so.

# 3. The method

## 3.1 The traditional method

The algorithms used by Brodley are decision trees, nearest neighbor classifiers and linear machines.

The data base used by Brodley are 11 types 3398 instances.

Table 3 the data set used by Brodley

| Class Name | Instances |
|---|---|
| broadleaf evergreen forest | 628 |
| coniferous evergreen forest & woodland | 320 |
| high latitude deciduous forest & woodland | 112 |
| tundra | 735 |
| deciduous-evergreen forest & woodland | 57 |
| wooded grassland | 212 |
| grassland | 348 |
| bare ground | 291 |
| cultivated | 527 |
| broadleaf deciduous forest & woodland | 15 |
| shrubs and bare ground | 153 |
| sum | 3398 |

The general procedure for identifying mislabeled instances is shown in Figure1. The first step is to split data set into n parts and then identify candidate instances by using filter algorithms to tag instances as correctly or incorrectly labeled. For each of the n parts, the filter algorithms are trained on the other n-1 parts. The m resulting classifiers are then used to tag each instance in the excluded part as either correct or mislabeled. An individual classifier tags an instance as mislabeled if it classifies the instance as belonging to a different class than that given by its training label.

Majority filtering tags an instance as mislabeled if more than half of the m base level classifiers classify it incorrectly. And consensus filtering requires that all base-level classifiers must fail to classify an instance as the class given by its training label for it to be eliminated from the training data.

The reason to employ ensemble classifiers in majority filtering and consensus filtering is that ensemble classifier has better performance than each base-level classifier on a dataset.

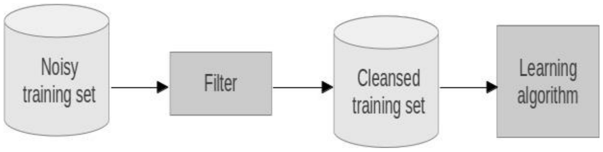


Figure. 1. Brodley’s procedure for learning in the presence of label mislabeled data with training set cleansing,

## 3.2 The improved method

Compared with Brodley’s algorithms, dataset and procedure, I introduce an improved method and the procedure is descripted as Figure2

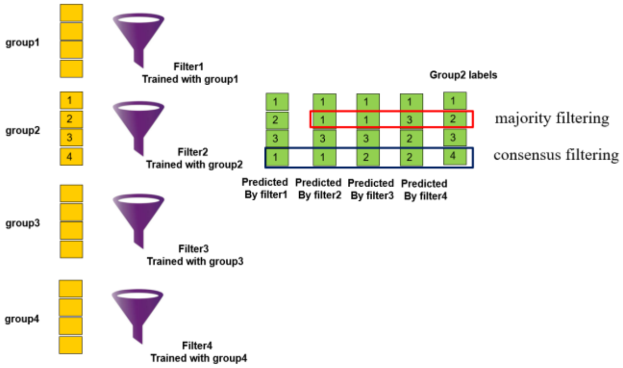


Figure. 2. Improved procedure of identifying mislabeled data

Compared with Brodley’s using n-1 part to train the filter, I split 60000 MNIST training data and 10000 test data into 7 groups, each group has 10000 digits. For example, group 1 is from 0-9999 in training dataset. As the same procedure I split 50000 CIFAR-10 training data and 10000 test data into 6 groups, each group has 10000 pictures.

I use each group as training data to train convolutional neural networks (CNNs) as filter algorithms, which has 91%-95% accuracy, then use the filter to predict all 7 groups and count the wrong predictions.

## 3.3. Base Filters

3.3.1CNN used in filter MNIST

CNN used in filtering MNIST is 4 layers CNN including:
First convolutional lay with filter size 4*4, channels 4;
Second convolutional lay with filter size 4*4, channels 16;
Third layer full connected size 2048;
The fourth layer uses softmax;
The irritation is 10000;
The performance is 92.5%-95.5%.

3.3.2CNN used in filter CIFAR-10

The CNN used in filter CIFAR-10 is Network in Network which has 9 convolutional layers and has 90% accuracy.

### 3.4. Ensemble Filters

In filtering ensemble classifier detects mislabeled instances by constructing a set of filtering detectors and then using their classification errors to identify mislabeled instances.

The general approach is to tag an instance as mislabeled if the filters cannot classify it correctly. The consensus filter tags an instance as mislabeled if all base filter detectors fail to classify an instance as the class given by its training label for it to be eliminated from the training data.

The majority filter tags an instance as mislabeled if more than half of the base filters classify it incorrectly.

3.4.1Consensus Filters

If all the base filters fail to classify the instance correctly, an instance is tagged as mislabeled instance.

One base filter Fi fail to classify xj as label yj:

$P(\mathrm{F}i(xj) \,!= \, yj) = (1 – \mathrm{accFi})$

accFi = accuracy of Filter i on data group G, xj∈G

All n base filters fail to classify xj as label yj:

$$P(\mathrm{F}(xj)\,!= yj) = \prod_{i=1}^{n} (\mathrm{P}(Fi(xj)\,!= yj)$$

3.4.2Majority filter

If more than half of the base filters fail to classify the instance correctly, an instance is tagged as mislabeled instance.

n-1 base filters fail to classify the xj as label yj:

$$\sum \prod P(Pi(xj))! = yj)P(Pk(xj) = yj)$$

All majority filters include from n-1 base filters fail to n/2 base filters fail.

## 4. Findings in MNIST

The accuracy of each filters are in the range of 0.925-0.955, the detail are in the table

Table 4 the accuracy of different filters

| | filter1 | filter2 | filter3 | filter4 | filter5 | filter6 | filter7 |
|---|---|---|---|---|---|---|---|
| 1 | 0.9487 | 0.929 | 0.9344 | 0.936 | 0.9287 | 0.9161 | 0.9261 |
| 2 | 0.9382 | 0.9445 | 0.9364 | 0.9415 | 0.9345 | 0.9171 | 0.9325 |
| 3 | 0.9374 | 0.9311 | 0.9455 | 0.9346 | 0.9313 | 0.9113 | 0.9273 |
| 4 | 0.9382 | 0.9295 | 0.938 | 0.9482 | 0.9307 | 0.9173 | 0.9289 |
| 5 | 0.9359 | 0.9272 | 0.9326 | 0.9344 | 0.9427 | 0.9127 | 0.926 |
| 6 | 0.9533 | 0.9462 | 0.9522 | 0.9507 | 0.9466 | 0.9428 | 0.9443 |
| 7 | 0.9453 | 0.9392 | 0.9434 | 0.9452 | 0.9394 | 0.9226 | 0.9508 |

Based the accuracy of above table

$$P(\mathrm{F}(xj)\,!= yj) = \prod_{i=1}^{n} (\mathrm{P}(Fi(xj)\,!= yj)$$

The P(Consensus(xj)=1) is from 1.007E-09 to 6.760E-09.

Table 5 counts of consensus filter and majority filter

| count of wrong predicted times | count of digits |
|---|---|
| 0 | 62161 |
| 1 | 2059 |
| 2 | 1008 |
| 3 | 733 |
| 4 (Majority Vote) | 622 |
| 5 (Majority Vote) | 593 |
| 6 (Majority Vote) | 722 |
| 7 (Consensus Filters) | 2023 |

Table 6 Distribution of Consensus Filters

| digit | count of digits |
|---|---|
| 0 | 92 |
| 1 | 102 |
| 2 | 251 |
| 3 | 325 |
| 4 | 220 |
| 5 | 216 |
| 6 | 103 |
| 7 | 185 |
| 8 | 252 |
| 9 | 277 |

### 4.1 Type of errors of mislabeled data

The most significant type of error is mistakes made during data input, which means no reason to be mislabeled.

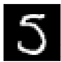
Subjectivity error may arise when observations need to be ranked in some way such as different type of handwriting.

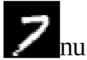
A third cause of labeling error arises when the information used to label each observation is inadequate.

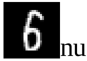
Table 7 count of 3 types of errors

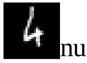
| digit | Count of input error | Count of subjectivity error | Count of insufficient data error |
|---|---|---|---|
| 0 | 0 | 2 | 28 |
| 1 | 0 | 3 | 38 |
| 2 | 0 | 3 | 56 |
| 3 | 1 | 1 | 12 |
| 4 | 1 | 3 | 60 |
| 5 | 1 | 1 | 48 |
| 6 | 1 | 0 | 41 |
| 7 | 0 | 1 | 24 |
| 8 | 0 | 2 | 39 |
| 9 | 0 | 1 | 14 |

four input error digits

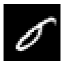

number 2080 labeled as 3

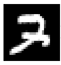

number 54915 labeled as 4

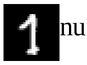

number 30310 labeled as 5


number 44960 labeled as 6

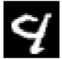
some examples of subjectivity error

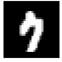

number 82 labeled as 0


number 45340 labeled as 3


number 21560 labeled as 7

some examples of insufficient data error


number 23652 labeled as 4


number 3637 labeled as 7

Distribution of Majority filter is shown in tabel8

Table 8 Distribution of Majority filter

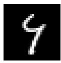
| digit | count of digits |
|---|---|
| 0 | 87 |
| 1 | 61 |
| 2 | 224 |
| 3 | 275 |
| 4 | 262 |
| 5 | 252 |
| 6 | 85 |
| 7 | 221 |
| 8 | 243 |
| 9 | 227 |

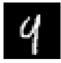
Different types of errors in Majority Vote is shown in table 9.

Table 9 count of 3 types of errors

| digit | Count of input error | Count of subjectivity error | Count of insufficient data error |
|---|---|---|---|
| 0 | 0 | 2 | 12 |
| 1 | 0 | 0 | 21 |
| 2 | 0 | 2 | 57 |
| 3 | 0 | 0 | 3 |
| 4 | 0 | 2 | 81 |
| 5 | 0 | 2 | 22 |
| 6 | 0 | 0 | 28 |
| 7 | 0 | 0 | 26 |
| 8 | 0 | 0 | 20 |
| 9 | 0 | 1 | 15 |

some examples of subjectivity error


number 50598 labeled as 4


number 62771 labeled as 4, which in test dataset.

### 4.2 Correcting these mislabeled data

For input errors, we can change the labels of these mislabeled data.

For subjectivity error and insufficient data error, we need the original form of these digits to give them a correct label. For example, d2564_59_00004.png in NIST SD19v2 is mislabeled as 1, and the original form proves that the label is 9.

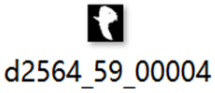


Figure. 3. d2564_59_00004.png

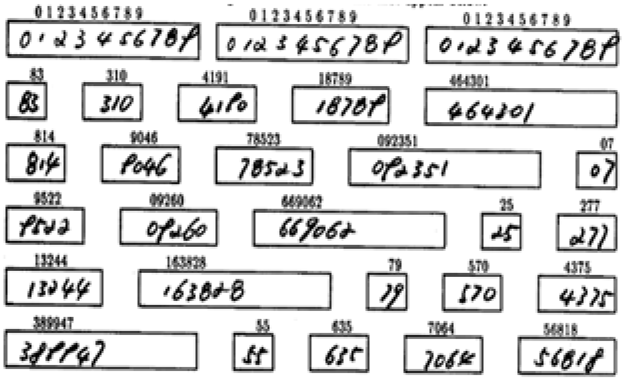

0123456789
0123456789
0123456789
83
310
4191
18789
464301
814
9046
78523
092351
07
9522
09260
669062
25
277
13244
163828
79
570
4375
389947
55
635
7064
56818

Figure. 4 original form of writer 2564 shows that writer write 9 in a different style

### 4.3 Effect of correcting data

After the changing of 4 input errors labels, the accuracy of the new training data set increases from 94.23% to 94.33%. (the accuracy is the average result of 10 times of training and testing).

The confusion matrix is from table 10 to table 11:

Table10 confusion matrix before changing data labels

| | y'=0 | y'=1 | y'=2 | y'=3 | y'=4 | y'=5 | y'=6 | y'=7 | y'=8 | y'=9 |
|---|---|---|---|---|---|---|---|---|---|---|
| y=0 | 963 | 0 | 0 | 0 | 0 | 1 | 11 | 1 | 4 | 0 |
| y=1 | 0 | 1114 | 5 | 1 | 1 | 1 | 3 | 1 | 9 | 0 |
| y=2 | 9 | 3 | 972 | 8 | 8 | 1 | 9 | 9 | 11 | 2 |
| y=3 | 3 | 2 | 17 | 935 | 0 | 17 | 0 | 11 | 16 | 9 |
| y=4 | 1 | 2 | 4 | 0 | 928 | 0 | 16 | 1 | 3 | 27 |
| y=5 | 10 | 3 | 2 | 15 | 7 | 819 | 17 | 1 | 15 | 3 |
| y=6 | 8 | 2 | 2 | 1 | 6 | 8 | 929 | 1 | 1 | 0 |
| y=7 | 0 | 6 | 24 | 4 | 7 | 1 | 0 | 951 | 3 | 32 |
| y=8 | 9 | 4 | 6 | 9 | 12 | 7 | 14 | 11 | 891 | 11 |
| y=9 | 8 | 6 | 1 | 9 | 45 | 4 | 1 | 12 | 8 | 915 |

Table11 confusion matrix before changing data labels

| | y'=0 | y'=1 | y'=2 | y'=3 | y'=4 | y'=5 | y'=6 | y'=7 | y'=8 | y'=9 |
|---|---|---|---|---|---|---|---|---|---|---|
| y=0 | 960 | 0 | 1 | 1 | 0 | 3 | 9 | 2 | 4 | 0 |
| y=1 | 0 | 1117 | 4 | 2 | 0 | 0 | 4 | 1 | 6 | 1 |
| y=2 | 7 | 0 | 970 | 8 | 8 | 3 | 6 | 14 | 12 | 4 |
| y=3 | 2 | 2 | 21 | 935 | 0 | 15 | 0 | 9 | 17 | 9 |
| y=4 | 0 | 1 | 5 | 1 | 909 | 1 | 12 | 3 | 3 | 47 |
| y=5 | 5 | 4 | 5 | 19 | 3 | 828 | 12 | 3 | 9 | 4 |
| y=6 | 8 | 3 | 1 | 0 | 5 | 7 | 931 | 0 | 3 | 0 |
| y=7 | 1 | 5 | 28 | 4 | 5 | 1 | 0 | 949 | 2 | 33 |
| y=8 | 10 | 1 | 5 | 17 | 6 | 12 | 7 | 9 | 893 | 14 |
| y=9 | 9 | 6 | 4 | 9 | 21 | 7 | 0 | 5 | 4 | 944 |

Only part of accuracy of digits 3,4,5,6,7 increases and other digits’ accuracy also increases.

After changing 33 mislabeled data of input errors and subjectivity error in MNIST, the accuracy of the new training data set increases from 94.23% to 94.4%.
Since I have subjective bias too, so the best way is to check the original form of 675 possibly mislabeled instances in MNIST.

## 5 Findings in CIFAR-10

The accuracy of each filters are in the range of 0.67-0.97, the detail are in the table
Accuracy of 6Filters on 6 groups:

Table 12 the accuracy of different filters

| group | filter1 | filter2 | filter3 | filter4 | filter5 | filter6 |
|---|---|---|---|---|---|---|
| 1 | 0.8242 | 0.7799 | 0.7828 | 0.7742 | 0.7778 | 0.7698 |
| 2 | 0.6727 | 0.9706 | 0.782 | 0.7621 | 0.7728 | 0.7677 |
| 3 | 0.6794 | 0.7849 | 0.9617 | 0.77 | 0.7773 | 0.7721 |
| 4 | 0.6716 | 0.7734 | 0.7786 | 0.9649 | 0.771 | 0.764 |
| 5 | 0.6735 | 0.7833 | 0.7829 | 0.7706 | 0.9627 | 0.7704 |
| 6 | 0.6725 | 0.7741 | 0.7821 | 0.765 | 0.7669 | 0.9609 |

Based the accuracy of such table

$$P(F(xj) != yj) = \prod_{i=1}^{n} (P(Fi(xj) != yj)$$

The P(Consensus(xj)=1) is from2.634E-05 to9.707E-05.
Count of Consensus Filters and Majority filter is shown in table 13.

Table 13 counts of consensus filter and majority filter

| count of wrong predicted times | count of digits |
|---|---|
| 0 | 30354 |
| 1 | 10617 |
| 2 | 5970 |
| 3 | 4450 |
| 4 (Majority Vote) | 3856 |
| 5 (Majority Vote) | 3553 |
| 6 (Consensus Filters) | 1200 |

Distribution of Consensus Filters is shown in table 14.

Table 14 Distribution of Consensus Filters

| digit | count of digits |
|---|---|
| 0 | 110 |
| 1 | 37 |
| 2 | 205 |
| 3 | 311 |
| 4 | 73 |
| 5 | 234 |
| 6 | 55 |
| 7 | 96 |
| 8 | 29 |
| 9 | 50 |

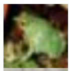
### Type of errors of mislabeled data

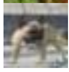
The input error has two subtypes, one is mislabeled, such as bird labeled as cat, the other type is the wrong pictures such as two people labeled as truck.
Subjectivity error has two subtypes, one is mislabeled, such as a dog might look like a cat, the other type is the wrong pictures such as a cow mislabeled as a horse.
A third cause of labeling error arises when two objects in one picture, such as a deer is beside a car.
The fourth type of error is when the information used to label each observation is inadequate.

Table 15 count of 4 types of errors

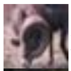
| digit | Count of input error | Count of subjectivity error | count of two object error | Count of insufficient data error |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1 | 1 | 3 |
| 3 | 4 | 2 | 1 | 8 |
| 4 | 1 | 3 | 1 | 6 |
| 5 | 0 | 1 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 2 |
| 8 | 0 | 0 | 0 | 2 |
| 9 | 1 | 1 | 1 | 11 |

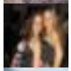
Input error with wrong labels

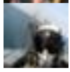

52405 labeled as cat


52804 labeled as cat

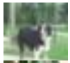
Input error with wrong pictures

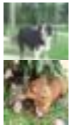

1569 labeled as a deer

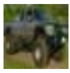

18310 labeled as truck


57524 is labeled as a cat

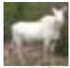
Subjectivity error with wrong labels

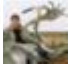

17455 labeled as a cat


57002 labeled as a cat

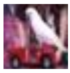

57967 labeled as a truck

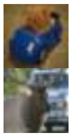
Subjectivity error with wrong pictures

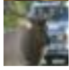

5074 labeled as a deer


4924 labeled as a deer

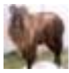
Two objects errors


15696: a bird on a car


56859 is a cat on a car


8208: a deer in front of a car

some examples of insufficient data error


45981 labeled as a deer

36710 labeled as a bird

Distribution of Majority filter is shown as table 16.

Table 16 Distribution of Majority filter

| digit | count of digits |
|---|---|
| 0 | 830 |
| 1 | 367 |
| 2 | 1146 |
| 3 | 1555 |
| 4 | 661 |
| 5 | 1004 |
| 6 | 486 |
| 7 | 678 |
| 8 | 316 |
| 9 | 366 |

Table 17 count of 4 types of errors

| digit | Count of input error | Count of subjectivity error | count of two object error | Count of insufficient data error |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 3 |
| 1 | 0 | 0 | 1 | 4 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 1 | 1 | 0 | 7 |
| 4 | 2 | 1 | 1 | 5 |
| 5 | 0 | 0 | 0 | 3 |
| 6 | 0 | 0 | 0 | 2 |
| 7 | 0 | 1 | 0 | 1 |
| 8 | 0 | 0 | 0 | 4 |
| 9 | 1 | 0 | 1 | 28 |

Images 56859, 57967, 57002, 57524, 52405 and 52804 are in the test dataset.

### 5.2 Correcting these mislabeled data

For input errors of wrong labels, we can change the labels of these mislabeled data.

For input errors of wrong pictures, we can replace it with new pictures.

For subjectivity error with wrong labels, we can change the labels.

For subjectivity error of wrong pictures, we can replace it with new pictures.

For two objects, we can recut it into two pictures with single object, or replace it with new pictures.

For insufficient data error, we need the original pictures to give them correct labels.

### 5.3 Effect of correcting data

After changing the label of 52405, 52804, 21347, 17455, 33079, 57002, 57967, 25684, 16359, 3879, 20738, the accuracy of the new training data set increases from 89.88% to 90.02%. (the accuracy is the average result of 10 times of training and testing).

## 6 Conclusions

Such examples show that the new method of identifying mislabeled data is an efficient method to detect mislabeled data. And there are still some pictures of MNIST and CIFAR10 need to be checked again with more information for the correct label. The state-of-the-art for MNIST and CIFAR-10 needs to change based on more accurate datasets.

# References

[1] [Brodley et al., 1999] C. E. Brodley and M. A. Friedl, “Identifying mislabeled training data,” J. Artif. Intell. Res., vol. 11, pp. 131–167, Jun. 1999.

[2] [Frenay et al., 2014] Classification in the Presence of Label Noise: a Survey, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 25, NO. 5, MAY 2014

[3] [Muhlenbach et al., 2004]Muhlenbach F., Lallich S., Zighed D.A.. Identifying and handling mislabelled instances, J. Intell. Inform. Syst. , 2004, vol. 22 (pg. 89-109)

[4] [Sanchez et al., 2003]Sanchez J.S., et al. Analysis of new techniques to obtain quality training sets, Patt. Recogn. Lett. , 2003, vol. 24 (pg. 1015-1022)

[5] [Aggarwal et al., 2001]Aggarwal C.C., Yu P.S.. Outlier detection for high dimensional data, 2001Proceedings of ACM SIGMOD 2001Santa Barbara, CA(pg. 37-46)