ENLD: Efficient Noisy Label Detection for Incremental Datasets in Data Lake

Xuanke You*, Lan Zhang*, Junyang Wang*, Zhimin Bao†, Yunfei Wu† and Shuaishuai Dong†
*University of Science and Technology of China, †Tencent Group
yxkyong@mail.ustc.edu.cn, zhanglan@ustc.edu.cn, iswangjy@mail.ustc.edu.cn, {zhiminbao, marcowu, shuaidong}@tencent.com

***Abstract*—Due to the difficulty of obtaining high-quality data in real-world scenarios, datasets inevitably contain noisy labeled data, leading to inefficient data usage and poor model performance. Thus, noisy label detection is an important research topic. Previous efforts mainly focus on noisy label detection on specific datasets that have been collected. Some works select clean samples based on relations between representations during the training process; some works utilize confidence outputs of a pre-trained model for noisy label detection. However, how to perform efficient and fine-grained noisy label detection on constantly arriving datasets in a data lake with a large amount of inventory data has not been explored. The rapidly growing volume and changing distribution of data make conventional methods either incur large computation overhead due to repeated training or become increasingly ineffective on newly arriving data. To address these challenges, in this work, we propose a novel approach ENLD to perform efficient and accurate noisy label detection on incremental datasets. Our extensive experiments demonstrate that ENLD outperforms the next best method in both efficiency and accuracy, which achieves $3.65\times$-$4.97\times$ detection speedup and higher average f1 scores with various noise rate settings.**

I. Introduction

In recent years, deep learning has made great achievements in various academic and industrial fields, which usually rely on a large number of labeled datasets [1] [2]. However, in the real world, both amateurs and experts inevitably produce noisy labeled data [3]. Therefore, noisy label detection and learning with noisy data have attracted much attention.

In industry, ubiquitous data lakes or data platforms provide massive data for deep learning systems, which also pose a huge challenge to data quality management [4]. There are two mainstream approaches to deal with noise labels, robust architecture and sample selection. Robust architecture reduces the influence of noisy labels to obtain a deep model with better performance by proposing robust training methods, such as noisy adaptation layer [5] [6], loss correction [7] [8] and label refurbishment [9] [10]. Sample selection explicitly filters noisy labeled data considering the impact of samples on training loss or the softmax output of deep models. Compared with the robust architecture, it can obtain a clean dataset with stronger reusability. A widely adopted idea for sample selection is to use some selection metrics (e.g. loss tracking) on samples during multiple rounds of the training process, such as O2U-Net [11] and INCV [12]. Topofilter [13] proposes a graph-based method in the latent representational space to collect clean data and drop isolated data. Confident learning [14] designs a framework to filter noisy labeled data with directly estimated joint distribution of noisy labels and unknown true labels based on confidence outputs of the deep model which is trained on noisy datasets.

Previous work, however, focus on datasets that have been collected. But for real-world data lakes and platforms, new data usually arrive constantly. Many platforms need to constantly perform accurate and efficient label quality assessments on newly arriving data, such as crowdsourcing platforms and data trading platforms [15] [16] [17]. Directly adopting existing training-based methods, e.g., Topofilter [13] and other loss tracking methods [11] [12], to detect noisy labels in incremental data is difficult to achieve good performance due to the lack of sample diversity and unbalanced categories in the incremental dataset. But applying those methods to both the inventory dataset and incremental dataset leads to a huge computation overhead due to the excessive sample number of the inventory data. Besides, the noisy label detection model trained on the inventory dataset usually cannot well adapt to specific incremental datasets. Pretrain-based methods, like confident learning [14], have low computation overhead but poor performance of noisy label detection for incremental datasets due to the changing data distribution. How to achieve efficient and accurate noisy label detection on constantly arriving datasets in a data lake is still an unexplored problem.

In this work, we focus on efficient and adaptive noisy label detection on constantly arriving incremental datasets in a data lake with a large amount of inventory data, and address the following challenges:

(1) How to leverage the knowledge from massive inventory data and how to adapt to the unknown data distribution of incremental data? Incremental datasets usually only contain a small number of samples from a part of classes of the inventory data and have unbalanced class distributions. Using the incremental datasets only cannot achieve satisfactory noisy label detection. It is crucial to mine and establish associations between incremental datasets and the inventory data, as well as to select proper samples from the inventory data as contrastive samples to improve the detection performance and reduce the training cost. During the selection of contrastive samples, it is necessary to consider the data distribution of incremental datasets for better adaptivity.

(2) How to ensure efficiency and performance during per-

forming continuous noisy label detection tasks? The platform will receive a large number of continuous noisy label detection tasks, each of which is time-consuming and computationally expensive. This requires our approach to be designed and implemented in a way that ensures both efficiency and performance.

Facing the above challenges, we propose a novel framework ENLD to efficiently perform noisy label detection on incremental datasets. The core idea of our design is to sample contrastive samples in inventory data, which greatly benefit identifying ambiguous samples in incremental datasets, and discover clean samples by majority voting through multiple fine-tuning processes. Specifically, ENLD is a two-stage framework. First, ENLD trains a general model and estimates the conditional probability of label mislabeling through inventory data. Then, ENLD conducts fine-grained noise label detection with contrastive sampling for specific incremental datasets, including multiple re-sampling and model fine-tuning. Our contributions are summarized as follows:

•We propose a novel framework ENLD to efficiently perform noisy label detection on incremental datasets. We consider label probabilities, output confidences of samples, and relationships between feature representations, and carefully design a set of techniques including contrastive sampling and fine-grained noisy label detection. ENLD achieves superior noisy label detection performance for newly arriving datasets, requiring only a small amount of fine-tuning.

•We analyze the rationality of the selected samples in contrastive sampling. Our analysis proves that the high-quality samples in inventory data that are close to the representations of ambiguous samples in incremental datasets can bring greater benefits to the training process. We also compare the influence of different sampling strategies on the fine-grained noise label detection in experiments.

•We extensively evaluate our framework on public datasets with various noise settings. Experiments demonstrate that our framework outperforms existing methods in both performance and efficiency for noisy label detection on incremental datasets. The average f1 score of ENLD achieves 0.9191 for EMNIST and 0.8194 for CIFAR100 for various noise settings, which outperforms the next best method, Topofilter. Compared with Topofilter, ENLD also achieves $4.09\times$ and $3.65\times$ detection speedup on average process time for EMNIST and CIFAR100, respectively. For a more complex classification task, Tiny-Imagenet, ENLD performs significantly better than the baseline methods. It achieves an average f1 score of 0.7297 while the average f1 score of Topofilter is only 0.6171, and achieves $4.97\times$ detection speedup on average process time.

II. Related Work and Preliminaries

A. Noisy Label Detection Methods

In noisy learning, recent works focus on methods of sample selection [18] [19], which attempts to first select clean samples in the dataset and train the DNN on the filtered cleaner dataset. Decouple [20] maintains two DNNs and selects clean samples for the model update by the difference in label predictions between two DNNs. MentorNet [21] completes sample selection through a collaborative learning method, in which the pre-trained mentor DNN guides the training of a student DNN, and the student receives clean samples with a high probability provided by the mentor. Co-teaching [22] maintains two DNNs, each DNN completes the selection of small loss samples and shares the results with another DNN for future training. Based on Co-teaching, Co-teaching Plus [23] integrates the disagreement strategy of Decouple. INCV [12] randomly splits the dataset into two parts and selects clean data through cross-validation. SELFIE [24] selects clean data by small-loss criteria and selective refurbishment of samples. [13] proposes a graph-based method in the latent representational space named Topofilter to collect clean data and drop isolated data. Confident learning [14] proposes a framework to filter noisy label data with directly estimated joint distribution of noisy label and unknown true label based on the softmax output of the deep model, which is trained on noisy datasets. However, previous works focus on collected datasets. It is not applicable to the scenario where noisy label detection needs to be performed repeatedly on the newly added datasets. In this work, we mainly focus on how to conduct efficient and accurate noisy label detection for incremental datasets.

B. Sample Selection Strategy

ENLD involves a sample selection process in inventory data for incremental datasets during the training process, and there are also many data selection strategies used in active learning methods [25] and semi-supervised learning methods [26]. And in active learning, the information entropy and confidence are widely used metrics to measure the uncertainty of samples for current models. It means samples with large uncertainty will bring great benefits to the training of the current model. Methods [27] [28] adopt the uncertainty-based sampling strategies to select samples during the training process. Moreover, the samples with the highest confidence tend to be selected and given a pseudo label to participate in the training in semi-supervised learning methods [29] [30] [31] and active learning methods [32]. In this work, we also conduct experiments on replacing different sampling strategies in the fine-grained noisy label detection method of ENLD to explore the impact of different sample selection strategies in Section V.

III. Problem and Main Idea

A. Problem Description

Given a large amount of inventory data (e.g. in a data lake) $I = \{(x_i^I, \tilde{y}_i^I)\}$ with a number of classes and samples, the system needs to perform noisy label detection on incoming incremental datasets $D = \{(x_i^D, \tilde{y}_i^D)\}$. Here, $\tilde{y}_i$ represents the observed label. y_i^* represents the unknown true label. The noise label in both I and D is generated by a label probability transition matrix $T_{i,j} = P(\tilde{y} = j|y^* = i)$. It represents the probability of mislabeling between labels in manual experience. In the actual scenario, D_i may be the dataset collected by the data platform or the dataset expected to obtain noisy label detection results from the data platform. The

TABLE I: Notation used in ENLD.

Notation	Definition
$\tilde{y}$	The observed label of the sample
y^*	The true label of the sample
I	The inventory data in the data platform
D	The constantly arriving incremental datasets
H	The high-quality samples in the inventory data
A	The ambiguous samples in the incremental dataset
θ	The general deep model trained with the inventory data
θ'	The finetuned model for incremental datasets based on θ
$M(x, \theta)$	The confidence output of sample x by the deep model θ
$\hat{M}(x, \theta)$	The feature vector of sample x by the deep model θ
$x_i^L, \tilde{y}_i^L$	The samples and observed labels in set L

goal of our framework is to efficiently perform accurate noisy label detection on the incremental dataset. Important notations are summarized in Table. I.

B. Main Idea

If we directly use the confidence outputs of a pre-trained general model on the incremental dataset to detect noisy samples, the performance is very dependent on the generalization ability of the general model trained by noisy labels, which often performs poorly on complex classification tasks. And previous training-based methods on the inventory dataset and incremental data will introduce a lot of computing overhead, which is not applicable to our scene as well.

To achieve requirements of high efficiency and accuracy, we expect to spend only a small amount of fine-tuning to achieve superior noisy label detection results for specific new datasets. Thus, we propose a two-stage framework for noisy label detection on incremental datasets, which maintains a general model and find-tune on different incremental datasets. Meanwhile, different incremental datasets have different data distributions and ambiguous samples for the general deep model. Here, ambiguous samples mean that their observed labels and predicted labels of the current model are inconsistent as defined in Definition. 1. The main idea of our work is to select high-quality contrastive samples for ambiguous samples in incremental datasets, and then finetune the model on specific data distribution to achieve accurate noisy label detection results. We consider label probabilities, output confidences, and feature representations of the current model to select contrastive samples which greatly benefit identifying ambiguous samples in incremental datasets in contrastive sampling.

IV. Framework of ENLD

In this section, the detailed design and implementation of ENLD will be introduced. We will first describe the framework overview of ENLD, then contrastive sampling, fine-grained noisy label detection, and finally the model update.

A. Framework Overview

We describe and introduce the framework overview of our proposed ENLD as shown in Algorithm 1 and Fig. 1. The platforms suitable for deploying the ENLD framework have a certain amount of inventory data, and incremental datasets with noise label detection requests arrive continuously. As for the platform, first, ENLD divides the inventory data I into I_t and I_c randomly. And then, ENLD initializes a general model θ with I_t and estimate the probability of $\tilde{P}(y^* = j|\tilde{y} = i)$. After the initialization of ENLD, the noisy label detection of incremental data sets can be performed. For example, when an incremental dataset D arrives, ENLD first performs contrastive sampling on current D to obtain an initial contrastive sample set C. Then a fine-grained noisy label detection method with re-sampling will be performed to obtain the selected clean part S and noisy part N of D based on the general model θ. Moreover, during the noisy label detection process of incremental datasets, the system can also perform data selection for the inventory data. The platform can choose to update the general model and re-estimate the probability of $\tilde{P}(y^* = j|\tilde{y} = i)$.

Algorithm 1 Framework of Efficient Noisy Label Detection (ENLD)

```
Input: the inventory data $I = \{(x_i^I, \tilde{y}_i^I)\}$, the incremental datasets $\{D_i\}_{i=1}^t$ ,the parameter of contrastive samples size $k$
Output: the noisy label detection result $S_i$, $N_i$
$\theta, \tilde{P}, I_t, I_c = model\_init(I)$;
$H = \{(x, \tilde{y}) \in I_c : argmax\ M(x, \theta) = \tilde{y}\}$;
$S_c = \emptyset$;
while $D_i$ arrives do
    $H' = \{(x, \tilde{y})\ :\ \tilde{y} \in label(D_i)\ and\ (x, \tilde{y}) \in H\}$;
    $A = \{(x, \tilde{y}) \in D_i : argmax\ M(x, \theta) \neq \tilde{y}\}$;
    $C = contrastive\_sampling(A, H', \tilde{P}, k, \theta)$;
    $S_i, N_i, S_c' = fined\_grained\_NLD(C, D_i, \theta)$;
    $S_c = S_c \bigcup S_c'$;
    $\theta, \tilde{P}, I_t, I_c = model\_update(S_c, I_t, I_c)$; // Optional;
end while
```

B. Model Initialization & Probability Estimation

In this part, the system needs to obtain a general model and estimate the probability of label mislabeling.

Model Initialization: First, we divide the inventory data I into I_t and I_c uniformly and randomly. Here, I_t represents the training set which is used to initialize and train a general model θ, and I_c is the candidate set for contrastive samples to accommodate special incremental datasets. In the system implementation, we use I_t to train the initialization model with the augmentation method Mixup [33]. Mixup randomly mixes the samples and labels with a Beta distribution for generalization performance as shown in Eq. 1 and Eq. 2, where $\lambda \sim Beta(\alpha, \alpha)$. We set the parameter of the Beta distribution $\alpha = 0.2$ in all experiments in Section V.

$$\hat{x} = \lambda x_i + (1 - \lambda) x_j \tag{1}$$

$$\hat{y} = \lambda y_i + (1 - \lambda) y_j \tag{2}$$

Probability Estimation: According to the assumption $\tilde{y}^* = argmax\ \tilde{p}(\tilde{y}; x, \theta)$ in [12], it means that the predicted label and the true label have the same distribution. We utilize the confidence output of the model $M(x, \theta)$ on I_c and observed label of each sample to estimate the joint distribution J of true

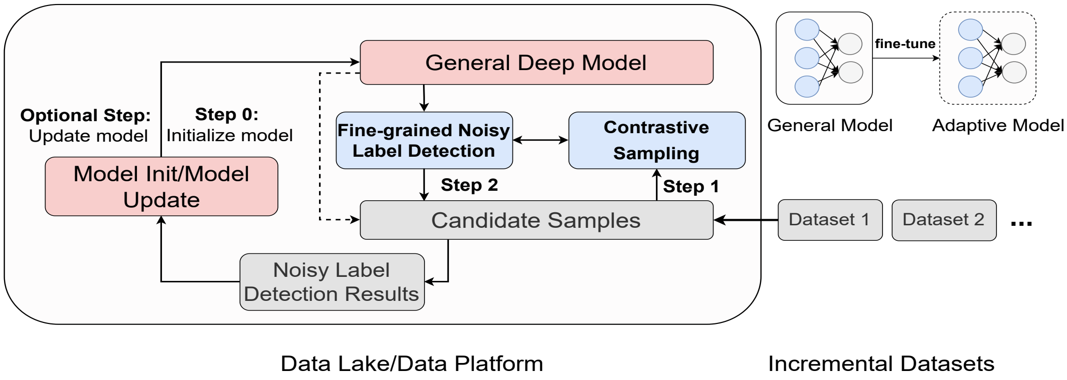


Fig. 1: Overview of ENLD framework. *Step 0:* ENLD initializes a general model θ and estimated the conditional probability. *Step 1& Step 2:* ENLD performs fine-grained noisy label detection with contrastive sampling for each dataset when incremental datasets arrive. *Optional Step:* ENLD can choose to update the general model and re-estimate the conditional probability by the model update process. The dash line between the general deep model and candidate samples means that ENLD utilizes the general model to select high-quality samples from candidate samples for contrastive sampling.

label y^* and observed label $\tilde{y}$ as shown in Eq. 3 and Eq. 4. Here, $M(x, \theta) = (o_1, o_2, ..., o_l)$ represents the softmax output of each class by deep model θ and the input sample x. o_i represents the confidence of class i and l represents the total categories of the classification task. And $argmax\ M(x, \theta)$ represents the predicted label of the sample x.

$$J_{i,j} = |D_{\tilde{y}=i, y^*=j}| \tag{3}$$

$$D_{\tilde{y}=i, y^*=j} = \{x \in D_{\tilde{y}=i} : argmax\ M(x, \theta) = j\} \tag{4}$$

As shown in Eq. 5, we can estimate the conditional probability $\tilde{P}(y^* = j | \tilde{y} = i)$ of observation labels and true labels through the estimated joint distribution J:

$$\tilde{P}(y^* = j | \tilde{y} = i) = \frac{\tilde{P}(y^* = j, \tilde{y} = i)}{\tilde{P}(\tilde{y} = i)} = \frac{J_{i,j}}{\sum_k J_{i,k}} \tag{5}$$

Finally, we obtained the general model θ for fine-grained noisy label detection and the estimated conditional probabilities $\tilde{P}$ that will be used in the contrastive sampling method.

C. High-quality and Ambiguous Samples

Definition 1. *We define the samples with $argmax\ M(x, \theta) \neq \tilde{y}$ in the incremental dataset D as the set of ambiguous samples A. And we define the samples with $argmax\ M(x, \theta) = \tilde{y}$ in the inventory data I as the set of high-quality samples H.*

In this section, we introduce the definition of high-quality samples in inventory data and ambiguous samples in the incremental data as shown in Definition. 1 by the predicted label of the model θ and observed label. We define the sample in the incremental dataset D whose predicted label is inequal to the observed label as an ambiguous sample. We only sample contrastive samples for ambiguous samples rather than all samples in the incremental dataset in order to reduce the number of contrastive samples. And we define the sample in the inventory dataset I_c whose predicted label is equal to the observed label as a high-quality sample. In the subsequent comparative sampling process, we expect the selected sample in contrastive sampling to be a clean sample.

D. Contrastive Sampling

In this section, we propose contrastive sampling to provide high-quality contrastive samples for the ambiguous samples in D. The core idea is to select contrastive samples with great training benefits for ambiguous samples in fine-tuning of fine-grained noisy label detection. To achieve this goal, we expect to select samples that have proximate feature representations with the targeted ambiguous sample and have the same true labels as the ambiguous samples. For example, a sample with an observed label 'bowl' in D is an ambiguous sample. Intuitively, selecting a clean sample with the label 'bowl' and similar feature representations to finetune the general model is helpful to determine whether the ambiguous sample is a noise sample. We carry out theoretical and experimental analysis on this intuition.

As shown in Algorithm 1 and Algorithm 2, when the noisy label detection request of a new dataset arrives, contrastive sampling will be utilized to obtain an initial contrastive sampling set C, and then it will be performed repeatedly in the fine-grained noisy label detection method to update the set C during the training process. First, we give a hyperparameter k, which represents the size of contrastive samples $k|A|$ in each sampling process. For each ambiguous sample, we first determine the label according to the estimated probability $\tilde{P}$. According to Corollary 1, for a specific incremental dataset D, we only select contrastive samples in a subset H' in I_c which contains observed labels in $label(D)$. Here, $label(D)$ represents the label set of D. Because, according to Corollary 1, the true label of a sample will be contained in $label(D)$ with probability of $1-(1-P(\tilde{y} = m | y^* = m))^{|D^m|}$. In practice, the probability of mislabeling $1 - P(\tilde{y} = m | y^* = m)$ is usually low. Therefore, as long as there is a certain number of D^m in the D, the true label m will have a great probability of being included in $label(D)$. Here, D^m represents samples in dataset D whose true labels are class m. And then, for an ambiguous sample x_i, we choose the k nearest samples in the high-

Algorithm 2 Contrastive Sampling

Input: the ambiguous samples of incremental dataset A, the high-quality samples of inventory data H, the estimated probability $\tilde{P}$, parameter of contrastive samples size k, the general model θ

Output: the contrastive samples C

```
$C = \emptyset$;
$A = \{A_i\}$;
for  $A_i$ in $A$  do
    for  $x_i$ in $A_i$  do
        $j = random\_label(i, \tilde{P}, label(H'))$;
        $C_i = k\_nearest(M(x_i, \theta), H_j, k)$;
        $C = C \cup C_i$;
    end for
end for
return  $C$
```

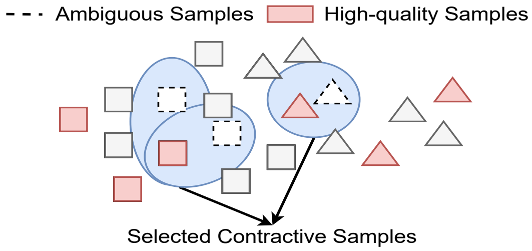


Fig. 2: An example of contrastive sampling.

quality samples in inventory data by the output representations $\hat{M}(x, \theta)$ in Euclidean distance as the contrastive samples C_i as shown in Eq. 7. $\hat{M}(x, \theta) = (v_1, v_2..., v_c)$ represents the feature output in front of softmax classifier layer by the deep model θ with the input sample x. Here, c represents the length of feature representations $\hat{M}(x, \theta)$.

Corollary 1. *In an incremental dataset D, if samples of class $D^m = \{(x_i, \tilde{y}_i)|y^* = m\}$ in D is collected uniformly from the true data distribution of class m. Then, the probability of class m not in $label(D)$ is $(1 - P(\tilde{y} = m|y^* = m))^{|D^m|}$.*

Poof Sketch. *According to the conditional probability $P(\tilde{y} = m|y^* = m)$, the probability of mislabeling represents:*

$$\hat{P} = 1 - P(\tilde{y} = m|y^* = m) \tag{6}$$

The probability of class m not in $label(D)$ is equivalent to that all samples D^m are mislabeled, and the probability is $(1 - P(\tilde{y} = m|y^* = m))^{|D^m|}$.

$$S(x_i, x_j) = ||\hat{M}(x_i, \theta) - \hat{M}(x_j, \theta)|| \tag{7}$$

Finally, we obtain a contrastive sample set C for the ambiguous set A. According to Corollary 2, ideally, if the estimated probability $\tilde{P}(y^* = i|\tilde{y} = k)$ is equal to the true probability $P(y^* = i|\tilde{y} = k)$, the label distribution $L(C)$ of sampled contrastive set will be the same as the true label distribution $L(A)$ of set A.

Corollary 2. *(Ideal) $L(C)$ represents the label distribution of set C. If the estimated probability $\tilde{P}(y^* = i|\tilde{y} = k) = P(y^* = i|\tilde{y} = k)$, the sampled contrastive set satisfies $E(L(C)) = L^*(A)$, where $P(y^* = i|\tilde{y} = k)$ represents the true conditional probability and $L^*(A)$ represents the true label distribution of set A.*

Poof Sketch. *According to the Algorithm 2, the expected label distribution of set C represents:*

$$E(L(C))_i = \sum_k L(A)_k \cdot \tilde{P}(y^* = i|\tilde{y} = k) \tag{8}$$

According to the total probability theorem:

$$L^*(A)_i = \sum_k L(A)_k \cdot P(y^* = i|\tilde{y} = k) \tag{9}$$

Thus, when $\tilde{P}(y^* = i|\tilde{y} = k) = P(y^* = i|\tilde{y} = k)$, we obtain $E(L(C))_i = L^*(A)_i$.

Moreover, we analyze the rationality of contrastive samples we select. First, we define the objective function of our model as $min\ loss_{test}(\theta, A_{test})$. Here, A_{test} is an unknown validation dataset that contains the same samples as the ambiguous set A and contains true labels rather than observed labels. As for $x_{test} = (x_i, y_i) \in A_{test}$, the contribution of adding a contrastive sample $x^I = (x_i + \epsilon, y_i)$ is shown in Definition 2. It means the loss gain of adding x^I in epoch t on x_{test} after training. According to the Corollary 3, if the gradient of loss function $\nabla_{\theta_t} loss(\theta_t, x)$ satisfies L Lipschitz smooth condition, the $\triangle I$ between adding x^I and directly adding x_{test} will be less than $\alpha L||\nabla_{\theta_{t-1}} loss(\theta_{t-1}, x_{test})|| \cdot ||\epsilon||$. It means adding correct contrastive samples with closer representations will bring greater benefits to the training process. This explains why we select the nearest samples as contrastive sampling in Algorithm 2.

Definition 2. *$I = \{(x_i^I, y_i^I)\}$ is the set of inventory data. $A = \{(x_i^A, y_i^A)\}$ is the set of ambiguous data in the incremental dataset. $A_{test} = \{(x_i, y_i)\}$ is the test set of correctly labeled data corresponding to A. The object function is $min\ loss_{test}(\theta, A_{test})$.*

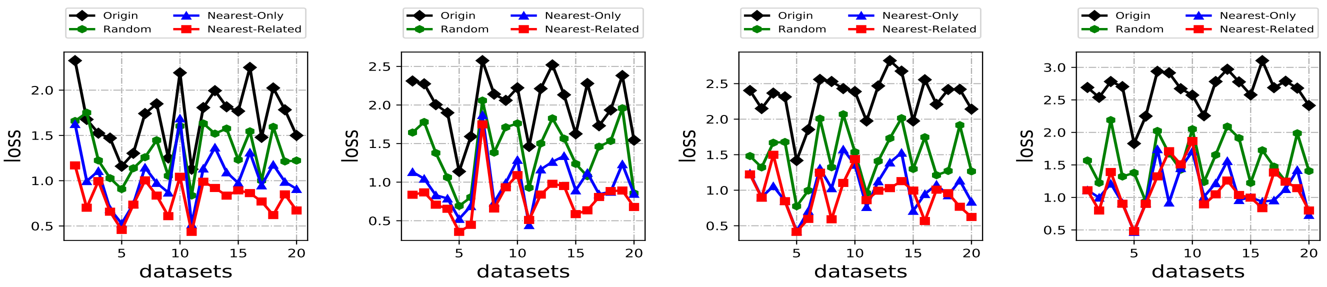
Definition 3. *The contribution of a sample x for each sample $x_{test} \in A_{test}$:*

$$I(x, x_{test}) = loss(\theta_{t-1}, x_{test}) - loss(\theta_t, x_{test}) \tag{10}$$

Corollary 3. *$x_{test} = (x_i, y_i)$ represents a sample in A_{test}, and $x^I = (x_i + \epsilon, y_i)$ represents a sample in the candidate set of constrasive samples. And $\triangle I = I(x_{test}, x_{test}) - I(x^I, x_{test})$ represents the contribution gap between adding x^I and x_{test}. If the gradient of loss function $\nabla_{\theta_t} loss(\theta_t, x)$ satisfies Lipschitz Smooth condition, at epoch t, we get $\triangle I \leq \alpha L||\nabla_{\theta_{t-1}} loss(\theta_{t-1}, x_{test})|| \cdot ||\epsilon||$.*

Poof Sketch. *At epoch t, with stochastic gradient descent after adding a sample x, the model will be updated as follows:*

$$loss_{test} = loss(\theta_t, A_{test}) \tag{11}$$



(a) Noise Rate: 0.1

(b) Noise Rate: 0.2

(c) Noise Rate: 0.3

(d) Noise Rate: 0.4

Fig. 3: Evaluation loss of the validation set D_{test} on incremental datasets of CIFAR100. Origin represents the original loss of general model θ. Random, Nearest-Only and Nearest-Related represent the loss after an epoch training by adding samples with true labels using different strategies.

$$\theta_t = \theta_{t-1} - \alpha \nabla_\theta loss(\theta_{t-1}, x) \tag{12}$$

Here, α is the learning rate. And the contribution of the sample x:

$$I(x, x_{test}) = loss_{test}(\theta_{t-1}, x_{test}) - loss_{test}(\theta_t, x_{test}) \tag{13}$$

According to the Lagrange mean value theorem:

$$loss(\theta_t, x_{test}) = loss(\theta_{t-1} - \alpha \nabla_\theta loss(\theta_{t-1}, x_i), x_{test}) \tag{14}$$

$$\begin{aligned} &loss(\theta_t, x_{test}) = loss(\theta_{t-1}, x_{test}) \\ &- \nabla_{\theta_{t-1}} loss_{test}(\theta_{t-1}, x_{test}) \cdot \alpha \nabla_{\theta_{t-1}} loss(\theta_{t-1}, x) \end{aligned} \tag{15}$$

Then we get:

$$I(x, x_{test}) = \alpha \nabla_{\theta_{t-1}} loss(\theta_{t-1}, x_{test}) \cdot \nabla_{\theta_{t-1}} loss(\theta_{t-1}, x) \tag{16}$$

And:

$$\triangle I = I(x_{test}, x_{test}) - I(x^I, x_{test}) \tag{17}$$

$$\begin{aligned} \triangle I \leq &\alpha ||\nabla_{\theta_{t-1}} loss(\theta_{t-1}, x_{test})|| \\ &\cdot ||\nabla_{\theta_{t-1}} loss(\theta_{t-1}, x_{test}) - \nabla_{\theta_{t-1}} loss(\theta_{t-1}, x^I)|| \end{aligned} \tag{18}$$

According to the Lipschitz Smooth:

$$\triangle I \leq \alpha L ||\nabla_{\theta_{t-1}} loss(\theta_{t-1}, x_{test})|| \cdot ||\epsilon|| \tag{19}$$

And we also conduct experiments to support this conclusion as shown in Fig. 3(a) $\sim$ Fig. 3(d). We set the general model θ as the initial training model for fine-tuning. $D_{test} = \{(x_{test}, y^*)\}$ of D represents the validation set of noisy set in the incremental datasets. We add samples with true labels and train for an epoch to explore the impact of adding strategies on the contribution to the current model. Random represents adding $|D_{test}|$ samples with true label randomly from I_c. Nearest-Only represents adding $|D_{test}|$ samples with closest representations to each x_{test} from I_c and corresponding true labels. Nearest-Related represents adding $|D_{test}|$ samples with closest representations to each x_{test} from I_c and true labels which are consistent with the true labels of x_{test}. It can be concluded that the nearest strategy effectively enables the model to obtain adaptive training and the final loss on the validation set is significantly lower than the original loss and the loss of random sample selection. Compared with Nearest-Only, Nearest-Related is more likely to select samples that make greater contributions to the training process. And contrastive sampling is also a re-weighting process of sampled contrastive samples. Although contrastive sampling has sampled $k|A|$ times, the final sampled set C actually contains fewer samples than $k|A|$. This is because some samples will be sampled more than once as shown in Fig. 2, which also indicates that these samples are more important for the current ambiguous sample set A. For example, a sample can be the contrastive sample for multiple ambiguous samples at the same time. The samples in the final sampled set C are equivalent to having different weights and then participate in the training process of fine-grained noisy label detection.

Implementation: Since constantly arriving datasets involves multiple k-nearest query operations, the original time complexity is $O(c|A||H'|)$. Thus, in implementation, we build KD-Tree structures for each category in H. KD-tree is a binary tree to organize vectors for more efficient nearest neighbor searching. This will reduce the time complexity of k-nearest operations to $O(k|A|log|H'|)$. It improves the efficiency of the contrastive sampling that needs to be executed repeatedly.

E. Fine-grained Noisy Label Detection

In this section, we introduce the fine-grained noisy label detection method in ENLD as shown in Algorithm 3. It mainly consists of the following four parts: (1) warming up process; (2) training and sample selection; (3) sample update and re-sampling; (4) data selection of inventory data. After contrastive samling, we can obtain an initial contrastive sample set C which is related to ambiguous samples in the incremental dataset. And the core idea of fine-grained noisy label detection is to fine-tune the general model on contrastive samples to select clean samples in incremental datasets. And the algorithm will adjust the representation and update the contrastive samples during the training process.

Algorithm 3 Fine-grained Noisy Label Detection

Input: the general model θ, the contrastive samples C, the incremental dataset D, the candidate set of contrastive samples I_c, the estimated probability $\tilde{P}$, parameter of contrastive samples size k, the training iteration t and the step s in each iteration

Output: the clean set S and the noisy set N of the incremental dataset D, the selected clean samples S_c of I_c

```
$S, N, S_c = \emptyset$;
$count_c = zeros(|D|)$;
$I' = \{(x, \tilde{y}) \ : \ \tilde{y} \in label(D) \ and \ (x, \tilde{y}) \in I_c\}$;
$\theta' = warming\_up(\theta, C, validate = D)$;
for $i$ in $iteration$ do
    $count = zeros(|D|)$;
    for $s$ in $step$ do
        $\theta' = train(C, \theta')$;
        $S_u = \{(x, \tilde{y}) \ : \ argmax \ M(x, \theta') = \tilde{y}, \ x \in D\}$;
        $count = update(count, S_u)$;
        $S_u = majority\_voting(count, S_u)$;
        $S = S \bigcup S_u$;
        $N = \{(x, \tilde{y}) \mid x \in D \ and \ x \notin S\}$;
    end for
    $A = \{(x, \tilde{y}) \in D : argmax \ M(x, \theta') \neq \tilde{y}\}$;
    $H' = \{(x, \tilde{y}) \in I' : argmax \ M(x, \theta') = \tilde{y}\}$;
    $count_c = update(count_c, H')$;
    $S'_c = majority\_voting(count_c, H')$;
    $S_c = S_c \bigcup S'_c$;
    $C = contrastive\_sampling(A, H', \tilde{P}, k)$;
    $C = C \bigcup S$;
end for
return $S$, $N$, $S_c$
```

Warming Up: At the first stage of the fine-grained noisy label detection, we utilize the initial contrastive sample set C and the incremental dataset D to train a better initialization model as shown in Algorithm 3. We use C to train the model θ for a given warming epoch number and verify the model on the incremental dataset D, and we selected the model with the highest validation accuracy during the warming up process.

Training and Sample Selection: There are two parameters to control the training process t and s. Here, t represents the total iterations of the training process and s represents the number of steps for training and clean sample selection in each iteration. In each iteration, we first initial a counting list and use it to count whether the predicted label in each step is equal to the observed label. In each step, we add the samples with more than $\lfloor \frac{s}{2} \rfloor + 1$ count times to the clean samples set S. For example, if a sample with an observed label 'bowl' in the incremental dataset is predicted as 'bowl' by the deep model θ' for more than $\lfloor \frac{s}{2} \rfloor + 1$ times after an iteration of finetuning, the sample will be selected as a clean sample. And then, with the updated model θ', we update the ambiguous samples A and the high-quality samples H' together with the representations $\hat{M}(x, \theta)$. Finally, we perform contrastive sampling to obtain a new contrastive set C and merge it with the selected set S to form a new C to ensure the stability of the training process. Here, we expect to select samples in high-quality samples as clean as possible, so we use the confidence output of θ to filter high-quality samples In practice, we filter the high-quality samples by average predicted probability $p(y_x^f = i) \geq \frac{\sum_x p(y_x^f = i)}{|\{y_x^f = i\}|}$ for cleaner contrastive samples. Here, y_x^f represents the predicted label $argmax \ M(x, \theta)$.

Sample Update and Re-sampling: In the end of each iteration, we utilize current model θ to update the ambiguous samples in D and the high-quality samples in I'. Then, the contrastive sampling method is called again to select contrastive samples for current ambiguous samples set A. Since the contrastive samples participate in the finetune training of fine-grained noisy label detection, the model will be more accurate in the selection of clean samples. The set of ambiguous samples in D will gradually decrease as shown in Fig. 13(b). We only use the current ambiguous samples set A to sample the contrastive samples, which can not only save the training cost by reducing the size of the contrastive samples, but also make the sampled contrastive samples more suitable for the current model and ambiguous samples.

Data Selection of Inventory Data: With the knowledge of noisy label detection on incremental datasets, we propose to select clean label S_c in each noisy label detection process for the model update of ENLD. We use the same counting method to count the number of times that each sample in the inventory data sample is determined to be a clean sample. In real scenarios, the inventory data usually serves multiple downstream tasks, so it is required that the selected samples of inventory data are as clean as possible. We adopt stringent clean data filter criteria as default for inventory data. Thus, we add the sample set S'_c with t count times to the selected samples set S_c in each iteration.

F. Model Update

After multiple noisy label detection tasks of incremental datasets, the system can choose to update the general model and re-estimate the probability. In this part, we introduce the model update process of ENLD as shown in Algorithm 4. ENLD utilizes the selected clean samples S_c in inventory data to update the model θ^u and validate the model on I_t to update the estimation probability P. Instead, in the later stage, the original I_t is used as the candidate set I_c of contrastive samples. In Section V, we verify that model update does improve the generalization ability of the general model.

V. Evaluations

In this section, we introduce the evaluation results of our proposed framework ENLD and various compared methods with public datasets and various noise rate settings.

A. Experimental Configuration

1) Datasets & Data Split: We use public image datasets, EMNIST [34], CIFAR100 [35] and Tiny-Imagenet [36]. We conduct three classification tasks, including a 26-categories

Algorithm 4 Model Update

```
Input: the selected set $S_c$ on $I_c$, the inventory data $I_c$ and $I_t$
Output: the updated model $\theta^u$, the estimated probability $P^u$, the updated $I_t$, $I_c$
$\theta^u = train(S_c)$;
$I_t, I_c = swap(I_t, I_c)$;
$P^u = evaluate(\theta^u, I_c)$;
return $\theta^u$, $P^u$, $I_t$, $I_c$
```

classification task on EMNIST letters with figure size $(28, 28, 1)$ and a 100-categories classification task on CIFAR100 with figure size $(32, 32, 3)$ and a 200-categories classification task on Tiny-Imagenet with figure size $(64, 64, 3)$. Firstly, We randomly divided each dataset into inventory data I and incremental dataset D according to the ratio of 2:1. As for EMNIST, we divide D into 10 unbalanced incremental datasets with 5 or 6 categories. As for CIFAR100, we divide D into 20 unbalanced incremental datasets with 10 categories. As for Tiny-Imagenet, we divide D into 20 unbalanced incremental datasets with 20 categories.

2) Asymmetric Noisy Label: To generate noisy labels, We corrupt the labels in our datasets with asymmetric noise, which is more realistic than symmetric (or uniform) noise. Asymmetric noise [3] means $\forall_{i=j}T_{ij} = 1 - \eta$ and $\exists_{i \neq j, i \neq k, j \neq k}T_{ij} > T_{ik}$. In this work, we adopt the pair asymmetric noise (widely used in previous work), which means $\forall_{i=j}T_{ij} = 1 - \eta$ and $\exists_{i \neq j}T_{ij} = \eta$. In our experiments, we adopt four noise rate settings $\eta \in \{0.1, 0.2, 0.3, 0.4\}$.

3) Metrics: In our experiments, we mainly focus on the performance and time cost of noise label detection on incremental datasets. As for performance, we focus on the precision, recall, and f1 score of the noisy label dataset $\tilde{D}_N^i$ detected from the original dataset D_i. And D_N^i represents the groundtruth of noisy label set in D. Thus, the precision metric is defined as $P = \frac{|D_N^i \cap \tilde{D}_N^i|}{|\tilde{D}_N^i|}$. The recall metric is defined as $R = \frac{|D_N^i \cap \tilde{D}_N^i|}{|D_N^i|}$. The f1 score is defined as $F1 = 2 \cdot \frac{P*R}{P+R}$.

Time Cost: The cost time of performing noisy label detection on each incremental dataset, including the process time of each incremental dataset and the setup time. The process time represents the waiting time to obtain the noisy label detection results when a new dataset arrives. The setup time represents the time of system initialization, which mainly refers to the training time of model initialization before processing noise label detection requests in our experiments.

4) Baseline Methods: We compare methods of explicitly selecting clean samples or noise samples as the comparison method of noise label detection in recent years.

Default represents utilizing the general model θ to select the noisy label data by $argmax\ M(x, \theta) \neq \tilde{y}$. Topofilter [13] utilizes the feature representation to construct KNN graphs and compute the largest connected component on each subgraph class by the class during a training process. Confident Learning [14] proposes a framework to filter noisy label data with directly estimated joint distribution of noisy label and unknown true label based on the softmax output of the deep model, which is pre-trained on noisy datasets. In our experiments, we utilize the general model θ trained on I_t and validate on I_c together with D_i. We report two methods in confident learning with the highest f1 score. Moreover, for a fair comparison, we perform Topofilter only on a subset of inventory data I which is related to the label set of incremental dataset $label(D_i)$.

5) Sampling Methods: We adjust the sample selection strategy in the fine-grained noisy label detection method in ENLD to analyze the impact of different strategies on the performance of noisy label detection on incremental datasets.

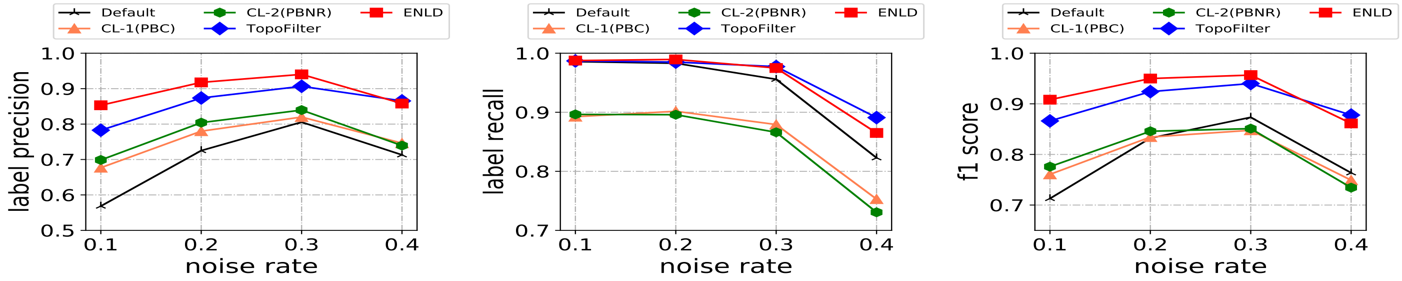
Random Policy: Random-ENLD uniformly and randomly selects samples in I_c; Highest Confidence Policy: HC-ENLD selects samples $(x_i, \tilde{y}_i)$ with highest confidence $max(M(x, \theta))$ according to outputs of current model in I_c; Least Confidence Policy: LC-ENLD selects samples $(x_i, \tilde{y}_i)$ with lowest confidence $max(M(x_i, \theta))$ according to outputs of current model in I_c; Entropy Policy: Entropy-ENLD selected samples with highest entropy of $M(x, \theta)$ according to outputs of current model in I_c; Moreover, we also propose Pseudo-ENLD to select samples with highest confidence $max(M(x, \theta))$ and replace the observed label $\tilde{y}$ by a pseudo label $argmax\ M(x, \theta)$ by the current model θ.

6) Experiment Settings: Unless otherwise noted in our experiments, we use Resnet-110 [37] with universal cross-entropy loss function in all of our experiments for various methods. To observe the generalization capability of ENLD, we also conduct experiments on Densenet-121 [38] and Resnet-164 [37] as shown in Section V-G. And we employ evaluations on the server with Inter(R) Xeon(R) CPU E5-2650 with 2.20GHz and Tesla P100 GPU. Unless otherwise noted in our experiments, we set the size of contrastive samples $k = 3$, the training step $s = 5$, and the warming up epoch equal to 2. We set the training iteration $t = 5$ for EMNIST and $t = 17$ for CIFAR100 and Tiny-Imagenet.

B. Results of Incremental Noisy Label Detection

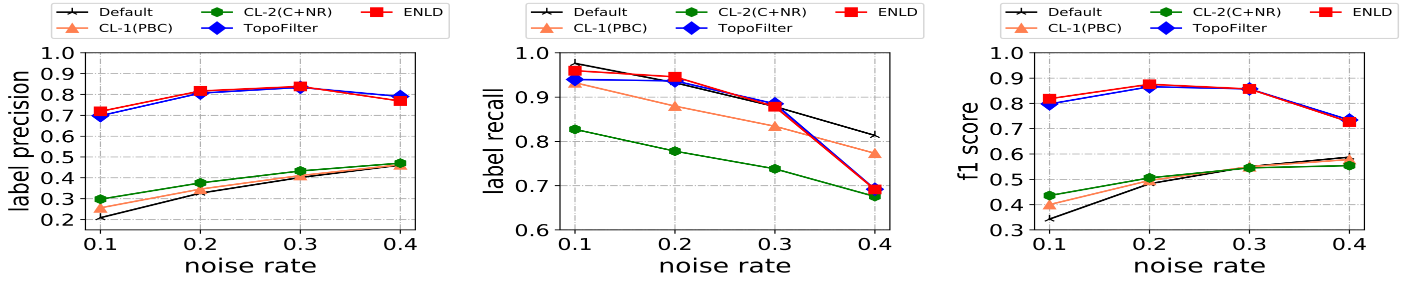
In this section, we compare the performance of various methods on incremental datasets of EMNIST, CIFAR100 and Tiny-Imagenet with various noise settings as shown in Fig. 4, Fig. 5 and Fig. 7. We demonstrate the cost time of noisy label detection on each incremental dataset as shown in Fig. 8 which contains both the setup time and process time. Default, Confident Learning, and ENLD have the same setup time of model initialization before performing noisy label detection for incremental datasets with 5438.2s for EMNIST, 18058.4s for CIFAR100, and 19716.7s for Tiny-Imagenet.

As shown in Fig. 4(c), Fig. 5(c) and Fig. 7(c), the training-based method, Topofilter, and ENLD, is obviously superior to the methods using only the confidence output of the general model, Default, Confident Learning methods (CL-1 and CL-2), but training process also brings additional computing overhead. Compared with the next-best method, Topofilter, ENLD achieves average f1 scores of 0.9191 for EMNIST and 0.8194 for CIFAR100 of various noise rate settings better than 0.9021 for EMNIST and 0.8139 for CIFAR100 of Topofilter. And as



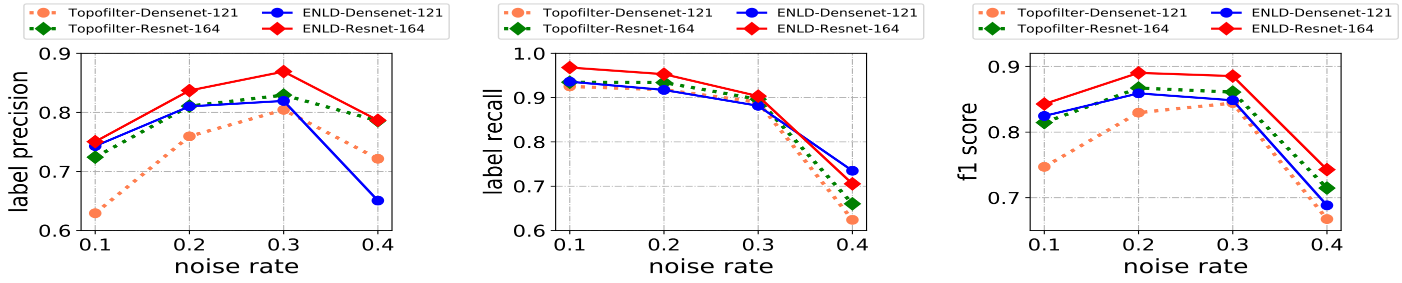
(a) Precision (b) Recall (c) F1 Score

Fig. 4: Performance of noisy label detection results with various detection methods on EMNIST. Average precision, recall and f1 score of 10 incremental datasets.



(a) Precision (b) Recall (c) F1 Score

Fig. 5: Performance of noisy label detection results with various detection methods on CIFAR100. Average precision, recall and f1 score of 20 incremental datasets.



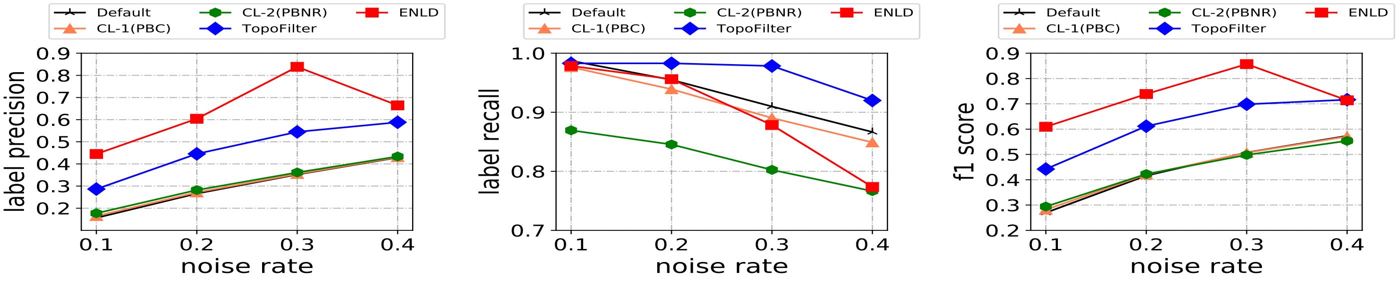
(a) Precision (b) Recall (c) F1 Score

Fig. 6: Performance of noisy label detection results with Densenet-121 and ResNet-164 on CIFAR100. Average precision, recall and f1 score of 20 incremental datasets.

shown in Fig. 8, ENLD also improves the average process time of each incremental dataset by 4.09× for EMNIST and 3.65× for CIFAR100 compared with Topofilter. For a more complex classification task, Tiny-Imagenet, ENLD is significantly better than baseline methods in terms of performance and time cost. Compared with the next-best method, it achieves an average f1 score of 0.7297 better than 0.6171 of Topofilter and saves 4.97× process time. As for Default and CL methods, since there is no additional training process, the performance of its noise label detection depends very much on the initialized model. Therefore, when the classification task is relatively simple, such as EMNIST, the performance is better than that of CIFAR100 and Tiny-Imagenet when the data and classification are more complex. In summary, ENLD can efficiently and accurately obtain the noise label detection results of new arrival datasets compared with other baseline methods.

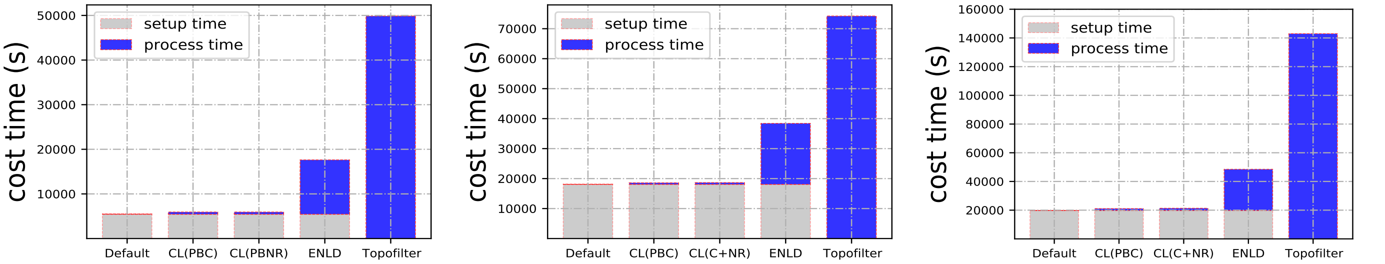
C. Training Process of ENLD

In this section, we demonstrate the noisy label detection process as shown in Fig. 9 when the noise rate is 0.1 $\sim$ 0.4 on CIFAR100. At the early stage of fine-grained noisy label detection, most samples are selected as noisy samples, so there is a high recall rate of noisy label detection. With the updating of the model and the re-sampling of the comparative samples, the precision and f1 score of noise label detection gradually increases while the recall slowly decreases. Finally, with the convergence of the method, the change tends to be gentle. In the case of low noise rate, the process of the fine-grained noise detection process is relatively stable, resulting in a slow decline of label recall with the discovery of noise label



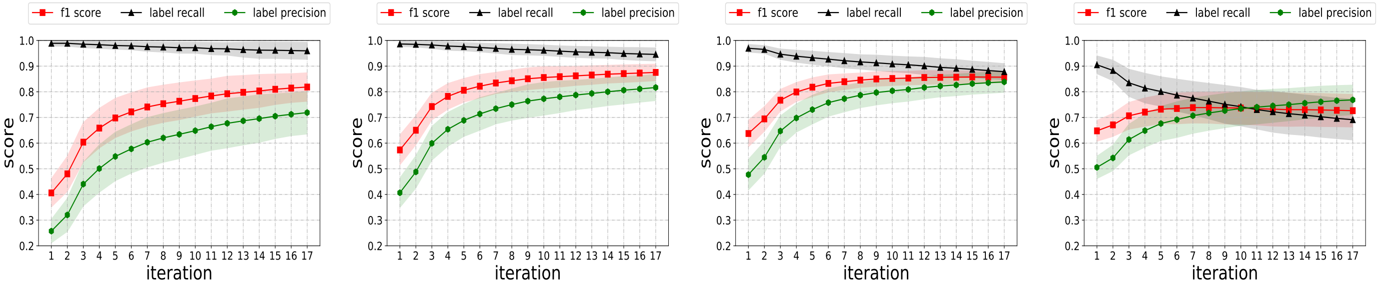
(a) Precision (b) Recall (c) F1 Score

Fig. 7: Performance of noisy label detection results with various detection methods on Tiny-Imagenet. Average precision, recall and f1 score of 20 incremental datasets.



(a) EMNIST (b) CIFAR100 (c) Tiny-Imagenet

Fig. 8: Setup time and process time cost of various methods on incremental datasts of datasets with various noise rate settings.



(a) 0.1 (b) 0.2 (c) 0.3 (d) 0.4

Fig. 9: Noisy label detection process of ENLD when the noise rate is 0.1∼0.4 on CIFAR100. Shaded regions indicate standard deviation over 20 incremental datasets.

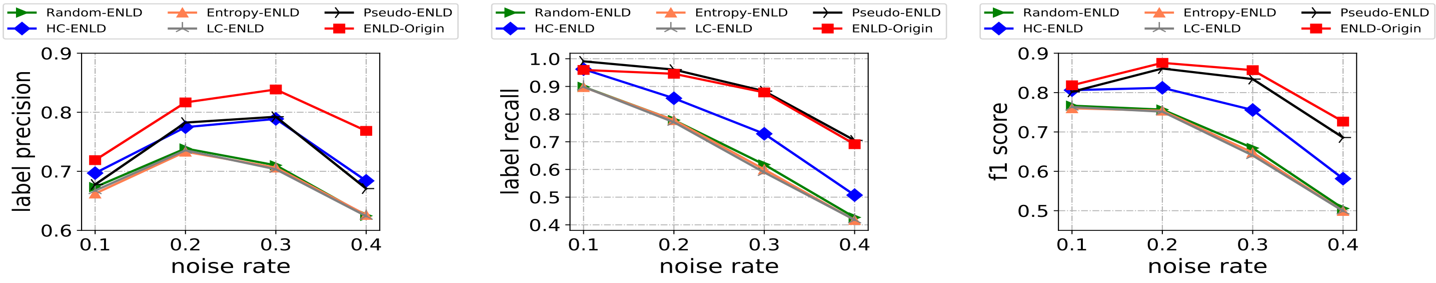
data, and a large increase in the f1 score. However, when the noise rate is 0.4, the label recall will decrease greatly with the discovery of noisy samples, and the increase of the f1 score is small and tends to flatten quickly. Therefore, under different system requirements, the performance and process time can be balanced by setting training iterations t. In practice, for scenes with higher noise rates, smaller t can be selected to save the process time of fine-grained noisy label detection.

D. Results of Sample Selection Strategy

In this section, we compare the performance of utilizing various sample selection methods in the fine-grained noisy label detection method on incremental datasets of CIFAR100 with various noise settings 0.1∼0.4 as shown in Fig. 10. It can be concluded that the overall performance of original contrastive sampling is superior to other strategies for the noisy label detection tasks. Different from active learning, because the true label of the sample cannot be obtained, the gain of noisy label detection by adding the most uncertain sample of the current model selected by entropy and least confidence is low and close to the random policy. Compared with the entropy, least confidence, and random policy, the highest confidence policy, and pseudo policy have a higher probability to select cleaner samples or obtain clean labels, so they can provide a reference for the process of noise label detection. Therefore, its performance is obviously better than entropy, least confidence, and random policy.

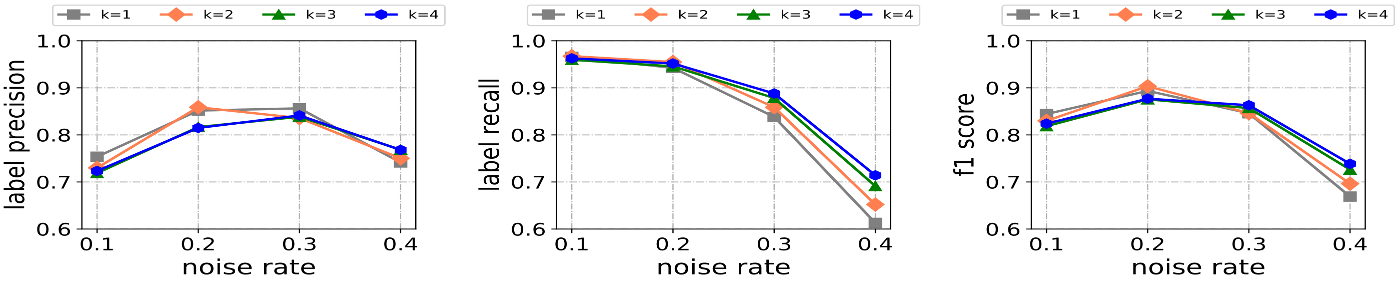
E. Model Update

In this section, we show the results of the model update process and data selection in ENLD. As shown in Table II, we demonstrate the validation accuracy on the entire set of incremental data and the other part of inventory data with original model θ and updated model θ^u by the data selection



(a) Precision (b) Recall (c) F1 Score

Fig. 10: Performance of noisy label detection results with various sample selection methods on CIFAR100. Average precision, recall and f1 score of 20 incremental datasets.



(a) Precision (b) Recall (c) F1 Score

Fig. 11: Performance of noisy label detection results with various hyperparameter settings on CIFAR100. Average precision, recall and f1 score of 20 incremental datasets.

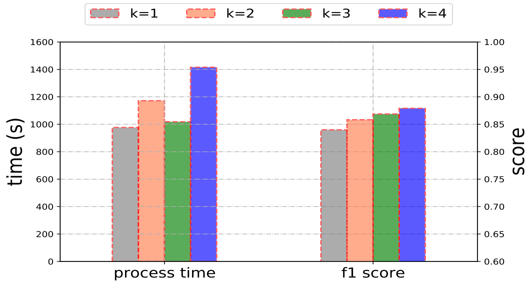


Fig. 12: Average process time cost and average f1 score of hyperparameter settings on incremental datasets with CIFAR100 with various noise rate settings.

Noise Rate	0.1	0.2	0.3	0.4
Origin Model	58.93%	52.85%	45.08%	37.17%
Update Model	61.31%	57.06%	49.40%	37.23%

TABLE II: Validation accuracy on remaining data on CIFAR100 by original model θ and updated model θ^u before and after the model update process.

result S_c when the noise rate is 0.1∼0.4 on CIFAR100. With clean samples selected by multiple noisy label detection tasks on incremental datasets, the generalization performance of the updated model has been significantly improved compared with the original one.

F. Hyperparameter Settings

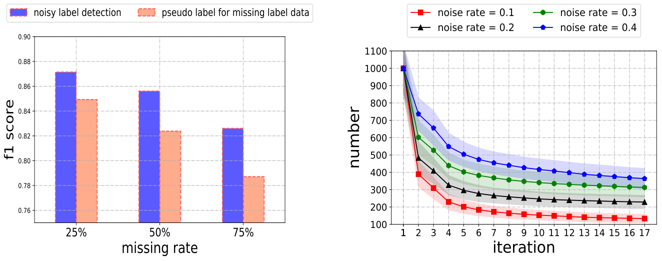
In this part, we conduct experiments on various hyperparameter settings of contrastive samples size $k = \{1, 2, 3, 4\}$ as shown in Fig. 11 and Fig. 12. It can be concluded that the performance of fine-grained noisy label detection increases gradually with the number of samples sampled by contrastive sampling, which also consumes more process time generally. However, compared with the process time of $k = 2$ and $k = 3$, the average process time does not increase but decreases. This is because choosing a larger k represents that there will be more contrastive samples for each ambiguous sample, which will lead to faster convergence of the model in the finetune training process. We think that the difference between setting $k = 2$ and $k = 3$ becomes significant. This finally leads to the average process time of setting $k = 2$ higher than that of setting $k = 3$. In our experiments, we choose a sampling size $k = 3$ with moderate performance and process time for all datasets and noise rate settings. Especially, Fig. 4(c), Fig. 5(c) and Fig. 7(c) show the f1 score of ENLD is slightly lower than that of the comparison method when the noise rate is 0.4. As shown in Fig. 11, increasing the sampling size can improve f1 scores when the noise rate is 0.4. Thus, we conduct experiments when $k = 4$ for each dataset. Finally, ENLD achieves average f1 scores of 88.06%, 73.86% and 72.62% for EMINST, CIFAR100 and Tiny-Imagenet, which are higher than 87.78%, 73.45% and 71.64% of the next best method, Topofilter. Therefore, we suggest that ENLD should choose a larger sampling size in the scene with a high noise rate.

G. Different networks

To observe the generalization capability of ENLD, we also conduct experiments on ENLD and Topofilter with Densenet-121 and ResNet-164 on incremental datasets of CIFAR100 as shown in Fig.6(a). For different networks, ENLD achieves better performance than Topofilter and saves 2.46× and 2.64× process time for Densenet-121 and ResNet-164.

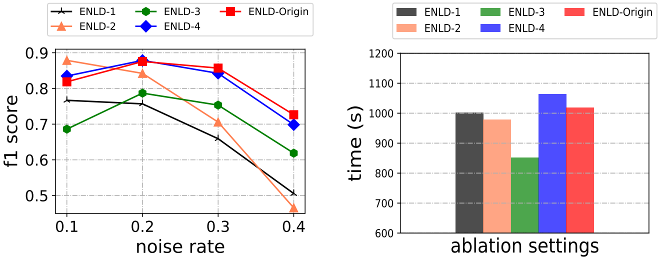
H. Missing label cases

Missing label can be regarded as a special case of the noisy label. We carried out extensive experiments on ENLD to explore its ability to deal with missing labels. First, we randomly set 25%, 50% and 75% samples in incremental datasets of CIFAR100 as missing label data when the noise rate is 0.2. ENLD will give a pseudo label for each sample without the observed label in each step of fine-grained noisy label detection. Each sample without the observed label will obtain a final label by voting with pseudo labels. Fig. 13(a) shows the average f1 scores of the pseudo label and noisy label detection with different missing rates. It demonstrates that the higher the missing rate of the incremental dataset, the lower the performance of pseudo labels and noisy label detection.



(a) Performance (b) Number of ambiguous samples

Fig. 13: (a) Average f1 scores of the pseudo label and noisy label detection with different missing rates of incremental datasets when the noise rate is 0.2 on CIFAR100. (b) Numbers of ambiguous samples during the fine-grained noisy label detection process on incremental datasets on CIFAR100.



(a) Performance (b) Average Process Time

Fig. 14: Ablation study results on ablation settings with various noise rate settings.

I. Ablation Study

And we conduct ablation study on ENLD to figure out the importance of each part by removing each part of ENLD separately when the noise rate is 0.1∼0.4 on incremental datasets of CIFAR100. ENLD-Origin represents the original version of the ENLD method; Removing contrastive learning (ENLD-1), ENLD with randomly selected data from contrastive samples set, represents utilizing randomly chosen samples to update the model in each step instead of contrastive sampling; Removing majority voting (ENLD-2) represents update the clean set once the predicted label is equal to the observed one; without adding clean samples of the incremental dataset (ENLD-3), which means removing $C = C \bigcup S$ in fine-grained noisy label detection. And we also propose ENLD-4 by using $j = i$ directly instead of $j = random\_label(i, \tilde{P}, label(H'))$ to query the nearest samples with the same observed label in the contrastive sampling method. As shown in Fig. 14, removing contrastive learning (ENLD-1) cause the overall performance of noise label detection to decline from 0.8139 to 0.6721 on the average f1 score. Therefore, contrastive learning is an essential part of ENLD. Removing majority voting (ENLD-2) means a more aggressive clean sample selection strategy. When the noise rate is low, the overall model is superior and the classification task is simple. A more aggressive clean sample selection strategy will bring a certain performance improvement. However, when the noise rate rises, removing majority voting will lead to greater randomness in the clean sample selection process, and the overall performance will be greatly reduced. Although without adding clean samples of incremental datasets during the training process (ENLD-3) reduces the process time of fine-grained noisy label detection to a certain extent, the performance is also greatly reduced due to the instability of its training process. As for ENLD-4, for the case of low noise rate 0.1, the strategy of directly selecting nearest samples that have the same observed label with ambiguous samples in contrastive sampling is better. However, for higher noise rates, such as 0.3 and 0.4, it is better to estimate the true labels of ambiguous samples according to the estimated conditional probability and then select high-quality samples that have proximate representations.

VI. Conclusion

In this work, we propose a novel framework ENLD to efficiently perform noisy label detection on incremental datasets, including the fine-grained noisy label detection method with contrastive sampling. The fine-grained noisy label detection method has the ability to achieve superior noisy label detection results for incremental datasets using only a small amount of fine-tuning, which involves label probabilities, output confidences, and relationships between the feature representations. The extensive experiments show the effectiveness of ENLD to perform noisy label detection on incremental datasets with various noise rate settings.

VII. Acknowledgment

Lan Zhang is the corresponding author. This research was supported by the National Key R&D Program of China 2021YFB2900103, China National Natural Science Foundation with No. 61932016, and the Fundamental Research Funds for the Central Universities WK2150110024.

References

[1] M. M. Kamani, S. Farhang, M. Mahdavi, and J. Z. Wang, "Targeted data-driven regularization for out-of-distribution generalization," in *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020.

[2] Y. Tang, F. Borisyuk, S. Malreddy, Y. Li, and S. Kirshner, "Msuru: Large scale e-commerce image classification with weakly supervised search data," in *the 25th ACM SIGKDD International Conference*, 2019.

[3] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[4] F. Nargesian, E. Zhu, R. J. Miller, K. Q. Pu, and P. C. Arocena, "Data lake management: challenges and opportunities," *Proceedings of the VLDB Endowment*, vol. 12, no. 12, pp. 1986–1989, 2019.

[5] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," *arXiv preprint arXiv:1406.2080*, 2014.

[6] X. Xia, B. Han, N. Wang, J. Deng, J. Li, Y. Mao, and T. Liu, "Extended t: Learning with mixed closed-set and open-set noisy labels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[7] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1944–1952.

[8] Y. Yao, T. Liu, B. Han, M. Gong, J. Deng, G. Niu, and M. Sugiyama, "Dual t: Reducing estimation error for transition matrix in label-noise learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 7260–7271.

[9] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *International Conference on Learning Representations*, 2015.

[10] P. Chen, J. Ye, G. Chen, J. Zhao, and P.-A. Heng, "Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 13, 2021, pp. 11 442–11 450.

[11] J. Huang, L. Qu, R. Jia, and B. Zhao, "O2u-net: A simple noisy label detection approach for deep neural networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3326–3334.

[12] P. Chen, B. B. Liao, G. Chen, and S. Zhang, "Understanding and utilizing deep neural networks trained with noisy labels," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1062–1070.

[13] P. Wu, S. Zheng, M. Goswami, D. Metaxas, and C. Chen, "A topological filter for learning with label noise," *Advances in neural information processing systems*, vol. 33, pp. 21 382–21 393, 2020.

[14] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *Journal of Artificial Intelligence Research*, vol. 70, pp. 1373–1411, 2021.

[15] L. Zhang, Y. Li, X. Xiao, X.-Y. Li, J. Wang, A. Zhou, and Q. Li, "Crowdbuy: Privacy-friendly image dataset purchasing via crowdsourcing," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 2735–2743.

[16] M.-C. Yuen, I. King, and K.-S. Leung, "A survey of crowdsourcing systems," in *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*. IEEE, 2011, pp. 766–773.

[17] D. Hettiachchi, V. Kostakos, and J. Goncalves, "A survey on task assignment in crowdsourcing," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–35, 2022.

[18] Y. Shen and S. Sanghavi, "Learning with bad training data via iterative trimmed loss minimization," in *International Conference on Machine Learning*, 2019.

[19] H. Song, M. Kim, D. Park, and J. G. Lee, "Prestopping: How does early stopping help generalization against label noise?" 2019.

[20] E. Malach and S. Shalev-Shwartz, "Decoupling "when to update" from "how to update"," *Advances in neural information processing systems*, vol. 30, 2017.

[21] J. Lu, Z. Zhou, T. Leung, L. J. Li, and F. L. Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *ICML 2018*, 2018.

[22] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[23] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama, "How does disagreement help generalization against label corruption?" in *International Conference on Machine Learning*. PMLR, 2019, pp. 7164–7173.

[24] H. Song, M. Kim, and J. G. Lee, "Selfie: Refurbishing unclean samples for robust deep learning," in *Proceedings of the 36 th International Conference on Machine Learning*, 2019.

[25] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM computing surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[26] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[27] T. He, X. Jin, G. Ding, L. Yi, and C. Yan, "Towards better uncertainty sampling: Active learning with multiple views for deep convolutional neural network," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 1360–1365.

[28] N. Ostapuk, J. Yang, and P. Cudré-Mauroux, "Activelink: deep active learning for link prediction in knowledge graphs," in *The World Wide Web Conference*, 2019, pp. 1398–1408.

[29] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013, p. 896.

[30] W. Dong-DongChen and Z.-H. WeiGao, "Tri-net for semi-supervised deep learning," in *Proceedings of twenty-seventh international joint conference on artificial intelligence*, 2018, pp. 2014–2020.

[31] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. Yuille, "Deep co-training for semi-supervised image recognition," in *Proceedings of the european conference on computer vision (eccv)*, 2018, pp. 135–152.

[32] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2591–2600, 2016.

[33] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[34] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," in *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 2921–2926.

[35] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[36] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[38] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.