# EER-ASSL: Combining Rollback Learning and Deep Learning for Rapid Adaptive Object Detection

**Minhaz Uddin Ahmed, Yeong Hyeon Kim, and Phill Kyu Rhee***

Department of Computer Engineering, Inha University,
Incheon, South Korea
[e-mail: minhaz.ahmed@gmail.com, ohpely@gmail.com, pkrhee@inha.ac.kr]
*Corresponding author: Phill Kyu Rhee

*Received September 20, 2019; revised April 10, 2020; revised June 9, 2020; accepted August 24, 2020; published December 31, 2020*

## Abstract

We propose a rapid adaptive learning framework for streaming object detection, called EER-ASSL. The method combines the expected error reduction (EER) dependent rollback learning and the active semi-supervised learning (ASSL) for a rapid adaptive CNN detector. Most CNN object detectors are built on the assumption of static data distribution. However, images are often noisy and biased, and the data distribution is imbalanced in a real world environment. The proposed method consists of collaborative sampling and EER-ASSL. The EER-ASSL utilizes the active learning (AL) and rollback based semi-supervised learning (SSL). The AL allows us to select more informative and representative samples measuring uncertainty and diversity. The SSL divides the selected streaming image samples into the bins and each bin repeatedly transfers the discriminative knowledge of the EER and CNN models to the next bin until convergence and incorporation with the EER rollback learning algorithm is achieved. The EER models provide a rapid short-term myopic adaptation and the CNN models an incremental long-term performance improvement. EER-ASSL can overcome noisy and biased labels in varying data distribution. Extensive experiments shows that EER-ASSL obtained 70.9 mAP compared to state-of-the-art technology such as Faster RCNN, SSD300, and YOLOv2.

***Keywords:*** Object Detection, Active Learning, Semi-Supervised Learning, Convolutional Neural Network

## 1. Introduction

Recently, there have been remarkable advancements in artificial intelligence and machine learning. Pattern classification technologies [1-3], the tasks of extracting patterns, and making predictions based on the knowledge learned from the patterns [4, 5], are still challenging problems, especially in a complicated and changing real world environment. Object detecting is one the important sub domain of computer vision. Last few years, deep learning technology has been applied successfully in many computer vision areas since the breakthrough by Krischesky et al. in 2012 [6, 7]. Most of the performance improvements rely on the availability of large, correctly labeled datasets on the assumption of a simple static data distribution [8]. However, the underlying distributions in the real world is very often varied and imbalanced, and the rebuilding of the system requires difficult and time-consuming labor and effort. Furthermore, some collected samples tend to be biased or badly labeled and may lead to performance degradation. A fully labeled retraining is infeasible in practice due to the cost and time constraints, but the acquisition of unlabeled data is relatively inexpensive. It can be of great practical value if one can make full use of both the labeled and unlabeled data.

The semi-supervised learning (SSL) which enables unlabeled data to be used in conjunction with the labeled data, can improve a lot in a learning performance [4, 9, 10]. The active learning (AL) can be thought of as a special case of semi-supervised learning. The AL uses minimum queries to the oracle to obtain labels of unknown data to optimize the model performance. In the AL process, an experienced expert intuition is critical in most cases for a successful model convergence. Human labeling always requires a heavy cost since it is labor intensive, time consuming, and error-prone. Thus, how to utilize ordinary human effort effectively is one of the major concerns in the AL research. The advantage of AL is to explore unlabeled samples considering the potentials of each unlabeled sample [11]. AL employs selective sampling for exploring the most informative samples with the minimum labeling cost [8, 12-16], instead of relying on the labeled data in a passive manner. The labeling efforts of training data in AL is reduced a lot when it is compared with traditional supervised learning methods. When an unlabeled data sample is labeled, it is used in forward learning manipulation, but may lead to model performance degradation in the presence of noisy samples or labels. The noises from mislabeled samples or outliers effect negatively in the building of the generalization ability. It can also have a negative impact in [4], the authors focus on reliable label estimation and enhancement to improve learning performance. Recently, bi-directional active learning (BDAL) was proposed for the improvement of the model generalization capability. BADAL combines the forward and backward learning processes based on the EER based uncertainty formulation and achieved superior performance to the unidirectional efforts [16]. However, BDAL still requires the time consuming redo in data selection and undo in data labeling, and cannot be used in a changing streaming data circumstance, whereby a rapid adaptive learning functionality is necessary. Motion information in weakly labeled video can be used to learn high precision object proposals. Krishna et al. shows that integrating candidate object recognition with weakly supervised learning improves detection performance [17]. The deep neural networks such as CNN are one of the most powerful layered learning networks that provide generalization capability and have done so over three decades [6, 7]. However, the slow convergence of the deep neural network is still a very hard issue and is a critical obstacle, especially in a real time data stream [18]. Combining AL and SSL, called ASSL can efficiently improve the object detection performance under varying data distribution [18-21] to overcome the limitations of CNN approaches that require large scale labeled datasets and a long training time. Recently, the incremental ASSL approach has been proposed in [22] and

shows that it can handle a time varying problem efficiently owing to the knowledge transfer capability of CNN and incremental learning. However, the combined method of the AL and SSL still requires long CNN deep learning steps, which spend a non-trivial amount of training time to satisfy the rapid adaptive learning capability. This paper proposes a rapid adaptive learning framework for object detection, EER-ASSL that combines EER learning and the deep learning SSL algorithms. Similarly, to the BDAL [4], if suspicious data samples are inspected from the labeled dataset, the rollback learning is conducted to rebuild the model by reselection or a relabeling mechanism (see **Fig. 1**(b)). In the rollback bin-based SSL, the selected batch samples are divided into several bins, and each bin repeatedly transfers the discriminative knowledge of the CNN deep learning and the EER-based rollback learning for rapid adaptation. The rollback learning method is embedded in the bin-based SSL to eliminate rapidly the effects of noisy uncertain samples in an imbalanced distribution. The above process is repeated until convergence is achieved. The novelties of the proposed EER-ASSL are given in the following.

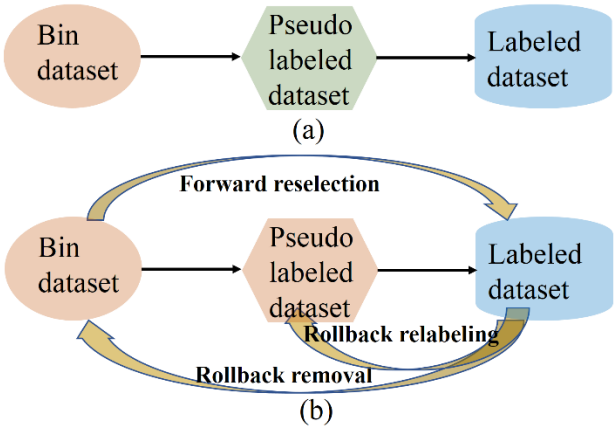


**Fig. 1.** The information flow snap shots of the bin-based SSL using (a) only forward deep learning, and (b) the EER rollback learning which consists of the forward reselection, rollback removal, and rollback relabeling, combined with the forward deep learning

1) The ERR-ASSL provides a rapid adaptive learning framework for efficient object detection in a changing environment. The method does not rely on the assumption of a static data distribution implicitly adopted by most state-of-the-art detection technologies. It combines the EER rollback learning and the CNN deep learning to transfer dynamic discriminative knowledge of the models from the current bin to the next one until convergence. Thus, EER-ASSL can overcome noisy and biased labels in an unknown data distribution. One can notice that the EER model enables a rapid short-term adaptation, and the CNN model helps to provide incremental long-term performance improvement.

2) The rollback learning method using the EER effectively leverages the discriminative capability by removing outliers or relabeling mislabeled samples. It provides a rapid adaptive learning functionality, while the unidirectional labeling and bi-directional approaches [4, 23] require a time-consuming redo in data selection and undo in a learning process. The EER based ASSL effectively utilizes the myopic adaptive learning capability of the EER and overcomes the drawback of the long training time of the deep learning object detector.

3) The proposed method is compared with state-of-the-art detection methods, such as Faster RCNN [5], SSD300 and YOLOv2, using the Pascal VOC 2007, 2012, MS COCO benchmark dataset and the local dataset. We significantly reduced the number of errors and improve the false object detection rate.

In the remaining section of this paper, we briefly discuss the latest related work in Section 2, and describe system overview in detail in Section 3. We mention the details of EER-ASSL algorithm in section 4. We show our results in Section 5, and outline our conclusion in Section 6.

## 2. Related Work

### 2.1 Active Semi-Supervised Learning (ASSL)

In many cases, we expect the labeled dataset is given and fixed. However, that is not always the case. It is possible for the labeled dataset to change if we employ an expert oracle or algorithm label and employ active learning (AL). AL has the ability to eliminate noisy data and correctly label more data types. AL plays a key role when we have to label a lesser amount of data and the algorithm has the ability to decide when to label and when not to label it. Zhu et al. (2003) first introduced the idea of effective combination of Semi-Supervised Learning (SSL) and Active Learning (AL) [23]. He applied the Gaussian random field model with AL so that unlabeled data can minimize the risk of the harmonic energy minimization function. By using pool-based active learning or selective sampling it is possible to minimize the number of query selections. AL and SSL combined together can improve the classification performance by exploiting both labeled and unlabeled data. Similar research has been conducted by Tong and Koller et al. (2000) where they reduce the version space size for SVM [25]. Cohn et al. (1996) minimize the estimated generalization error by the reduction of the variance components [26]. Once the queries are selected, most of these active learning methods do not take the chance of exploiting large amount of unlabeled data. Some of the researchers applied semi-supervised learning during the training phase. Chaloner and Verdinelli (1995) applied the Bayesian approach [27]. The Gaussian random field works well with harmonic function due to the combination of SSL and AL. There are many benefits to ASSL because ASSL can efficiently estimate a querying point and then randomly select a sample with ambiguity and still estimate the expected generalization error. Better label selection criteria can remove the ambiguity and the imbalances of sample distribution and overcome the imperfect labeling and selection biases while training an object detector [20, 24].

### 2.2 Expected error reduction (EER)

EER reduces the generalization error as a selection criterion when labeling a new sample based on retaining-based active learning. The effectiveness of the method has been proven in text classification applications [28], and the works in [16, 29, 30]. The rationale is to select the sample that is minimizing the future generalization error. The unlabeled pool is a representative of the test distribution and used as a validation set. Since no knowledge is available about the labels of unlabeled samples, EER estimates the average-case potential loss in terms of the average case in [28, 29, 31], worst-case in [32], or even the best-case criteria in [28]. EER explores the change of an expected error model by selecting the sample leading to the maximum change of the current model. A similar approach can be found in the variance reduction method, which tries to minimize the output variances. After adopting variance and logistic regression, it is extended to the expected variance reduction method on logistic

regression in [16]. In the mini-max view active learning, the worst-case criterion is adapted to select the sample which minimizes the objective function after labeling a new sample. This method is extended by considering all the unlabeled data when calculating the objective function. However, the EER-based retraining approaches require heavy computation since all the unlabeled data and all the possible labels are explored [16].

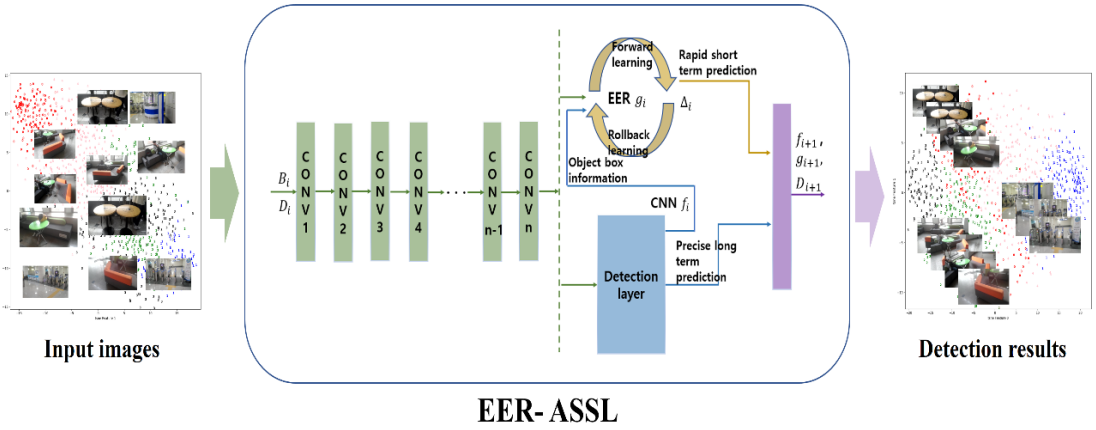


**Fig. 2.** EER-ASSL architecture

## 3. System Overview

The proposed method consists of collaborative sampling and EER-ASSL. The proposed EER-ASSL incorporates the rapid modeling capability from the EER method [4, 16] and the precise and incremental learning capability of CNN [21]. The method takes advantage of the exploration capability of the AL algorithm [34] and the step-by-step exploitation of the bin-based SSL algorithm with the rollback functionality. The method minimizes not only the training time but also costly human efforts, and at the same time keeps a high-quality labeled dataset and high-accurate object detector in an adaptive learning process [33]. The collaborative sampling method considering the uncertainty, diversity, and confidence criteria can select more informative and reliable samples with low redundancy. Since the uncertainty criterion may cause the selection of noisy or redundant samples, the diversity criterion is applied in the clustering based redundant removal algorithm [21]. In this paper, the collaborative sampling-based AL learning algorithm incorporates with the rollback bin-based SSL algorithm.

The brief sketch of the EER-ASSL for rapid adaptive object detection in a dynamically changing environment is given in **Fig. 2**. A batch of samples is selected based on uncertainty and diversity sampling from an image stream, instead of a single image at a time. After the collaborative sampling, the sampled batch stream is divided into bins for the rollback SSL algorithm. The samples in an image bin are unlabeled samples and pseudo-labeled by the bin-based SSL using both the CNN detector and EER-based rollback learning method. However, the pseudo dataset includes not only incorrectly labeled samples but also biased labels under imbalanced data distribution. The noisy samples should be excluded from the confident samples since such samples do harmful effect and never create any contribution to building a better object detector. The proposed EER-based learning method is adopted for the rapid

forward and rollback learning for more informative sampling and learning by the act of reselection and relabeling.

## 4. EER-ASSL Learning

We use a very limited number of labeled data for training the EER model and the fine-tuned CNN model. The fine-tuned CNN model and EER model are built using the limited labeled data samples. The ensemble network consists of the CNN detector and EER model conduct object detection. The incremental ASSL is adopted, whereby a batch of data samples are collected from an input data stream, and selectively sampled by the collaborative sampling algorithm [21]. The selected samples are partitioned into the bins, where the size of the batch and bin are decided in accordance with the image capturing quality. The initially labeled dataset and the pseudo training dataset by the CNN and EER ensemble are used for training a new CNN and EER models, i.e., used to update the models for the next bin cycle. The new EER model is involved in the rollback learning process which is consisting of the removal, relabeling, and reselecting samples from the bin, if necessary. The bin-based incremental learning is processed incorporation with the forward learning for the sample reselection, and the rollback learning for the removal and relabeling (see **Fig. 3**). The new CNN model is also used in the process of the new collaborative sampling. The process is repeated until convergence.

### 4.1 EER forward learning process

We adopt the expected error reduction (EER) method which was proposed and employed in pattern classification problems [4, 16, 28, 29]. The objective of the EER method is to choose a sample that minimizes a generalization error in the future step. Since the testing data is not available in advance, a portion of the streaming data set is used as the validation dataset to estimate the future error. The true labels of the unlabeled dataset are not known, and the future errors are approximately estimated using the expected log-loss over the unlabeled data [16]. Let $\boldsymbol{LD} = \{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{\boldsymbol{i=1}}^{\boldsymbol{m}}$ denote a labeled training dataset, and $\boldsymbol{UD} = \{\boldsymbol{x_i}\}_{\boldsymbol{i=m+1}}^{\boldsymbol{n}}$ is an unlabeled data set, where m $\ll \boldsymbol{n}$ . If a selected sample x is labeled y, and added to LD, it is denoted by $\boldsymbol{LD^+} = \boldsymbol{LD} \cup (\boldsymbol{x}, \boldsymbol{y})$. Let $\boldsymbol{g_{LD}}$ denote the EER model from $\boldsymbol{LD}$ and $\boldsymbol{g_{LD^+}}$ from $\boldsymbol{LD^+}$. The most informative data sample is assumed to maximize the expected error reduction by minimizing the expected entropy using the unlabeled dataset. Following to [16], we describe the most informative data sample in the unlabeled dataset is selected one that satisfies the following equation:

$$\boldsymbol{x}^* = argmin_{x \in UL} \sum_{y \in C} P(y|\boldsymbol{x}; g_{LD}) \times \left(-\sum_{x' \in UL, y' \in C} P(y'|\boldsymbol{x}'; g_{LD^+}) log(y', \boldsymbol{x}', g_{LD^+})\right), \quad (1)$$

where $\boldsymbol{C}$ indicates the object classes, the first term $\boldsymbol{P(y|x; g_{LD})}$ denotes the label information of the current model, and the second term is the sum of the expected entropy on the unlabeled data UD with the model $\boldsymbol{g_{LD^+}}$. Eq. (1) formulates the serial mode learning process, whereas a new model is updated immediately after the labeling of each new data sample in UD. But, the exhaustive repetition of the learning is challenged by heavy computational overhead in practice. In this paper, we divide the unlabeled stream dataset into batches, and each batch is divided into bins, instead of handling whole unlabeled data samples in UD as shown in **Fig. 3**.

In the rollback SSL, a bin of unlabeled data samples is used for each training step. For the time step i, Eq. (1) is rewritten considering bin $\boldsymbol{B_i}$ as follows:

$$x^*_{B_i} = argmin_{x \in B_i} \sum_{y \in C} P(y|x; g_{LD}) \times \left(-\sum_{x' \in B_i, y' \in C} P(y'|x'; g_{LD^+}) log(y', x', g_{LD^+})\right) \quad (2)$$

where the first term denotes the label information of the current model, and the second term is the sum of the expected entropy on the unlabeled data bin $\boldsymbol{B_i}$ with the model $\boldsymbol{g_{LD^+}}$. After applying the collaborative sampling, we can determine the pseudo labeled set $\boldsymbol{\Delta_i} = \{\boldsymbol{x_1}, \dots, \boldsymbol{x_k}\}$ for the bin dataset repeatedly applying Eq. (2). However, it still requires a heavy computation overhead to build the model for each data sample in $\boldsymbol{B_i}$. Thus, Eq. (2) is approximated by building the model for the selected samples of the pseudo labeled set $\boldsymbol{\Delta_i}$ as follows:

$$\Delta_i = argmin_{\{\boldsymbol{x_1}, \dots, \boldsymbol{x_k}\} \in B_i} \sum_{\boldsymbol{x_i} \in B_i, y \in C} P(y|\boldsymbol{x}; g_{LD}) \times \left(-\sum_{\boldsymbol{x'} \in B_i, y' \in C} P\left(y'\middle|\boldsymbol{x'}; g_{LD^{+\Delta_i}}\right) log(y', \boldsymbol{x'}, g_{LD^{+\Delta_i}})\right), \quad (3)$$

where the first term denotes the label information of the current model for the selected samples of the pseudo labeled set $\boldsymbol{\Delta_i}$, and the second term is the sum of the expected entropy on the unlabeled data $\boldsymbol{B_i}$ with the weight model $\boldsymbol{g_{LD^{+\Delta_t}}}$. If a selected sample $\{\boldsymbol{x_1}, \dots, \boldsymbol{x_k}\}$ with labeled $\{\boldsymbol{y_1}, \dots, \boldsymbol{y_k}\}$, and added to LD, denoted by $\boldsymbol{LD^{+\Delta_t}} = \boldsymbol{LD} \cup \{(\boldsymbol{x_1}, \boldsymbol{y_1}) \dots, (\boldsymbol{x_k}, \boldsymbol{y_k})\}$. One can notice that we build the model once, instead of building models for whole unlabeled data sample using Eq. (1). The EER forward learning process is used for the reselection algorithm of $\boldsymbol{\Delta_i}$ for retraining of the bin which is failed by the CNN detector. The reselected samples added to the current labeled dataset, and the combined dataset is used to retain the CNN model for the bin-based SSL step.

### 4.2 EER rollback learning process

The objective is to investigate the most uncertain labeled samples that disturb the current model, and select new samples replacing the most uncertain samples or relabeling the recently pseudo labeled samples. We employ the EER estimation for the rollback learning to minimize the expected entropy over pseudo labeled data samples. Considering the computation time, we don't consider the whole unlabeled dataset but only inspect the most recent pseudo labeled dataset. In the rollback process, we find the candidate samples of removal or relabeling from the recently pseudo labeled sample(s). The rollback learning process conducts the certification of the label(s) of the rollback sample(s) by relabeling it or reselecting from its neighborhood. The rollback learning is divided into two types: 1) the removal process and 2) the relabeling process.

Removal process: The removal rollback process is to undo the bad effect of the data sample by removing it from the current pseudo dataset added to the labeled dataset since the rollback sample(s) are suspected to disturb the EER model. The model using the previously labeled dataset is used to update the model since it is expected more reliable than the last labeled dataset. During the removal of rollback learning, the influence of the rollback samples is removed from the pseudo dataset of the model. The removal rollback process discards the unreliable samples from the pseudo labeled dataset, and the unreliable samples are detected. The rollback removal process is formulated as follows:

$$\boldsymbol{x}^{\dagger} = argmin_{x \in LD} \sum_{y \in C} P\left(y \middle| \boldsymbol{x}; g_{LD\backslash(\boldsymbol{x},y^{\dagger})}\right) \times \tag{4}$$

$$\left(-\sum_{\boldsymbol{x}' \in UL, y' \in C} P\left(y' \middle| \boldsymbol{x}'; g_{LD\backslash(\boldsymbol{x},y^{\dagger})}\right) \log\left(y', \boldsymbol{x}', g_{LD\backslash(\boldsymbol{x},y^{\dagger})}\right)\right), \tag{5}$$

where $\boldsymbol{LD}\backslash(\boldsymbol{x}, \boldsymbol{y}^{\dagger})$ denotes the labeled dataset after removing $(\boldsymbol{x}, \boldsymbol{y}^{\dagger})$. Since Eq. (4) requires a heavy computation time not computable in practice, rollback samples, $\boldsymbol{R}_{\Delta}^{\boldsymbol{remov}}$ are chosen from the pseudo labeled data samples of the current step, instead of those from the whole unlabeled dataset. The training dataset rolls back from $\boldsymbol{LD}$ to $\boldsymbol{LD}\backslash\boldsymbol{R}_{\Delta}^{\boldsymbol{remov}}$, and replace them new samples from the neighbors in the current feature space. We are looking for a pool of labeled samples to remove that minimizes the entropy over the unlabeled dataset by the removal, and other pseudo labeled samples will be reselected in the reselection process. The rollback samples for the removal will be selected from only the bin of the last pseudo labeled samples using the classification model as follows:

$$R_{\Delta}^{remov} = argmin_{\{x_1, \ldots, x_r\} \in \Delta_t} \sum_{x_i \in R_{\Delta}^{remov}, y \in C} P\left(y \middle| x; g_{LD\backslash R_{\Delta}^{remov}}\right) \times \left(-\sum_{x' \in \Delta_t, y' \in C} P\left(y' \middle| x'; g_{LD\backslash R_{\Delta}^{remov}}\right) log\left(y', x', g_{LD\backslash R_{\Delta}^{remov}}\right)\right), \tag{6}$$

where $\boldsymbol{R}_{\Delta}^{\boldsymbol{remov}}$ denotes rollback samples to be removed for the reselection process. If the selected rollback samples are $\{\boldsymbol{x_1}, \ldots, \boldsymbol{x_r}\}$ which were pseudo labeled by $\{\boldsymbol{y_1}, \ldots, \boldsymbol{y_r}\}$, respectively, and removed from LD, denoted by the set difference $\boldsymbol{LD}\backslash\boldsymbol{R}_{\Delta} = \boldsymbol{LD}\backslash \{(\boldsymbol{x_1}, \boldsymbol{y_1}) \ldots, (\boldsymbol{x_r}, \boldsymbol{y_r})\}$.

Relabeling process: The selected relabeling samples in $\boldsymbol{R}_{\Delta}$ are updated or added to LD. If a label of the rollback sample is changed after the relabeling rollback learning process, the forward learning label is replaced with the new one. In this way, the forward labeling error is corrected. If a new label is the same with the forward learning one, the rollback sample will be treated as a new sample. It will be added to LD. Similar ideas of boosting are discussed in [4], whereby the mislabeled sample candidate will be focused more. The relabeling rollback learning for one sample is performed by following the formulation similarly to [4] as follows:

$$x^{\dagger} = argmin_{x \in LD} \frac{1}{Z} \sum_{y_i \in C, i \neq i^{\dagger}} P\left(y_i \middle| x; g_{LD\backslash(x,y_i^{\dagger})}\right) \times \left(-\sum_{x' \in UL, y' \in C} P\left(y' \middle| x'; g_{LD|(x,y_i)}\right) log\left(y', x', g_{LD|(x,y_i)}\right)\right) \tag{7}$$

where $\boldsymbol{LD}|(\boldsymbol{x}, \boldsymbol{y_i})$ denotes the pseudo labeled $\boldsymbol{x}$ assigned by $\boldsymbol{y_i}$. Z is a normalization coefficient calculated by

$$Z = \sum_{i \neq i^{\dagger},\ y_i \in C} P\left(y_i \middle| x; g_{LD\backslash(x,y_i^{\dagger})}\right) = 1 - \left(y_i{}^{\dagger} \middle| x; g_{LD\backslash(x,y_i^{\dagger})}\right)$$

Considering the computation overhead of calculating the model for each relabeled candidates, we formalize the relabeling rollback learning process in terms of a pool of relabeled candidates in $\boldsymbol{\Delta_i}$ as follows.

$$R_{\Delta}^{relab} = argmin_{\{x_1,\dots,x_r\}\in\Delta_i} \frac{1}{Z} \sum_{y_i\in C, i\notin R_{\Delta}^{relab}} P\left(y_i \middle| x; g_{LD\setminus R_{\Delta}^{relab}}\right) \times \left(-\sum_{x'\in\Delta_i, y'\in C} P\left(y' \middle| x'; g_{LD|(x,y_i)}\right) log(y', x', g_{LD|(x,y_i)})\right), \quad (8)$$

where $LD|(x, y_i)$ denotes the pseudo labeled $\boldsymbol{x}$ assigned by $y_i$. Z is a normalization coefficient calculated by

$$Z = \sum_{i\notin R_{\Delta}^{relab},\ y_i\in C} P\left(y_i \middle| x; g_{LD\setminus R_{\Delta}^{relab}}\right) = 1 - P\left(y_i{}^{\dagger} \middle| x; g_{LD\setminus R_{\Delta}^{relab}}\right), \text{ where } y_i{}^{\dagger} \in R_{\Delta}^{relab}. \quad (9)$$

### 4.3 EER-ASSL Algorithm

The major tasks of the proposed EER-ASSL consist of the collaborative sampling-based AL and the rollback bin-based SSL. In the AL process, a batch of data samples is collected from an input data stream, processed by the collaborative sampling algorithm for the informative samples with minimum redundancy, is partitioned into bins. In the rollback SSL, the EER based rollback learning and the bin-based SSL are combined for rapid adaptive learning. The limited labeled samples are used to initialize the CNN model and EER model. The models are trained in the bin-based the incremental SSL scheme. If a performance criterion is violated in learning the CNN model, the EER method is activated for rapid rollback learning. The volume of the reliable labeled dataset $LD$ is increased by adding the pseudo labeled data samples. The enlarged LD is used to build the next CNN and EER models. The process is repeated until convergence. One can notice that the EER rollback model provides a rapid short- term adaptation, and a confident and the CNN detector model an incremental long-term performance improvement, respectively.

Let $D_{div}$ denote the samples mined from the current batch after the collaborative sampling. The detailed discussion can be found in [22]. We focus on the rollback bin-based SSL algorithm here. Let $D_{\Delta}$ denote the confidential batch dataset for the bin-based SSL, and it will be used as the container of the confidential batch dataset. If the cardinality of $D_{\Delta}$ becomes confidence parameter $\gamma$, the confidence sample selection process is stopped. $D_{\Delta}$ is initialized with a sample that satisfies $x_{top} = argmax_{x\in D_{div}} f(x)$, $x_{top} \in D_{div}$. The confidential sampling strategy chooses a sample from $D_{div}$ and adds to $D_{\Delta}$ according to the distance metric of the current deep feature space using $x_{top} = argmax_{x\in D_{div}} \{max_{x_i, x_j \in D_{\Delta}} d(x_i, x_j)\}$, where $d(x_i, x_j)$ is Euclidian distance between two samples $x_i$ $and$ $x_j$ in the deep feature space. The CNN is retrained using the bin sequence from the confidential samples in $D_{\Delta}$. The confidential samples are partitioned into the bins, and stored in bin pool denoted by $\boldsymbol{B}(= \{B_j\}_{j\in\boldsymbol{B}})$ or $= \{B_{0,\dots}, B_{j,\dots} B_J\}$.

In each rollback bin-based SSL step, the confidence scores are assigned to the pseudo samples by the current CNN detector. The labeled data $D_0$ is used to initialize CNN detector model $f_0$ and EER model $g_0$ in the beginning, respectively. $Acc_0$ is calculated by $f^0$using the validation data. For each bin, we build the CNN models $\{f_0^{B_j}\}_{j=1}^{J}$ using $D_0 \cup B_j$, respectively. Let $Acc_1$ indicate the maximum accuracy among the scores of the bins calculated by $\{f_0^{B_j}\}_{j=1}^{J}$,

i.e., $Acc_1 = \max_{B_j}\{Acc_0^{B_j}\}$. If the performance improved, i.e., $Acc_1 \geq Acc_0$, we move to the next step by updating, $D_1 = D_0 \cup B^*$ and $f_1 = f_0^{B^*}$. At time step i, for each bin, build the CNN models $\{f_i^{B_j}\}_{j=1}^{J}$ using $D_i \cup B_j$, respectively, and $Acc_{i+1} = \max_{B_j}\{Acc_i^{B_j}\}$. The cases are divided into three: Case 1) $Acc_{i+1} \geq Acc_i$, Case 2) $Acc_i - \tau < Acc_{i+1}\ Acc_i$, and Case 3) $Acc_{i+1} \leq Acc_i - \tau$, where $\tau$ is a tolerance threshold for an exploration potential.

Case 1: we get the best bin for the next step and update $D_{i+1} = D_i \cup B^*$ and $f_{i+1} = f_i^{B^*}$; $B_i = B^*$; $\boldsymbol{B} = \boldsymbol{B} \setminus B_i$ . Note that the bin pool $\boldsymbol{B}$ is reduced by removing the selected bin.

Case 2: we conduct the following sub-steps: 1) find the removal samples from $\Delta_i$ using the rollback learning process using Eq. (7), 2) find the relabeling samples, and assign them new labels from $\Delta_i$ using the rollback learning process based on Eq. (5), and 3) update $\Delta_i$ by reselection using the EER forward learning process based on Eq. (3).

The above the forward-rollback learning processes are repeated, until the condition of $Acc_{i+1} \geq Acc_i$ or $Acc_{i+1} \leq Acc_i - \tau$ or a time limit. If the condition $Acc_{i+1} \geq Acc_i$ is satisfied, we update $D_{i+1} = D_i \cup \Delta_j$, $f_{i+1} = f_i^{\Delta_j}$, $g_{i+1} = g_i^{\Delta_j}$, and $\boldsymbol{B} = \boldsymbol{B} \setminus B_i$ .

Case 3: oracle labels incorrectly labeled data in $B^*$, and update $f_{i+1} = f_i, g_{i+1} = g_i, D_{i+1}$.

The rollback process of Case 2 can reduce significantly the oracle labeling steps. $(D_i \cup \Delta_i)$ is used to build a training data set $D_{i+1}$, which is used for training $f_{i+1}$ and $g_{i+1}$ at time t. The process is repeated until convergence. Finally, the rollback bin-based SSL produces the two models f and g, and enlarged labeled dataset LD. The combination the EER based rollback learning and the bin-based SSL allows obtaining a rapid adaptive object detector, even from the noisy streaming samples under a dynamically changing environment. The rollback bin-based SSL algorithm is summarized in Algorithm 1.

**Algorithm** 1. Rollback bin-based SSL
**Input**: bin pool $\boldsymbol{B}$
**Output**: CNN model *f*, EER model *g*, and labeled dataset *LD*.

**Repeat until** $\boldsymbol{B} \neq \phi$

1. For each bin, $B_i \in \boldsymbol{B}$, build $f_{i+1}$ using $D_i \cup B_i$, and calculate $Acc_i^{B_i}$.

2. $Acc_{i+1} = \max_{B_j}\{Acc_i^{B_j}\}$.

3. If $Acc_{i+1} \geq Acc_i$,

$B^* = \operatorname{argmax}_{B_j}\{Acc_i^{B_j}\}$, $D_{i+1} = D_i \cup B^*$;

$f_{i+1} = f_i^{B^*}$ ; $B_i = B^*$, and $\boldsymbol{B} = \boldsymbol{B} \setminus B_i$ .

4. Else if $Acc_i - \tau < Acc_{i+1} < Acc_i$,

While $Acc_i - \tau < Acc_{i+1} < Acc_i$,

4.1 Remove the samples from $\Delta_i$ , i.e., the removing rollback process using Eq. (5).

4.2 Relabel the samples in $\Delta_i$ i.e., the relabeling rollback

process using Eq. (7).

4.3 Reselect the samples from $B_i$ using the forward learning process using Eq. (3).

4.4. If $Acc_{i+1} \geq Acc_i$, $D_{i+1} = D_i \cup \Delta_j$, $f_{i+1} = f_i^{\Delta_j}$, $g_{i+1} = g_i^{\Delta_j}$, and $\boldsymbol{B} = \boldsymbol{B} \backslash B_i$ . $i$++

Else if $Acc_{i+1} < Acc_i - \tau$ or time limit, oracle labels incorrectly labeled data in $B^*$.

$f_{i+1} = f_i, g_{i+1} = g_i, D_{i+1}$, i++.

**Return** $\{f = f_{i+1}, g = g_{i+1}, LD = D_{i+1}\}$

## 5. Experiments

Extensive experiments are conducted using the benchmark datasets, such as PASCAL VOC as well as a local dataset, and the performances are compared with state-of-the-art detectors technology such as Faster RCNN, SSD300, and YOLOv2. The experimental implementations are conducted using a single server with a single NVIDIA TITAN X with cuDNN [10] and Tensorflow [36]. We used the experiment settings as the Darknet-19 CNN model [30] with the base detector is YOLOv2, which is the state-of-the-art object detector.

### 5.1. Benchmark Datasets

PASCAL VOC dataset: Famous PASCAL VOC benchmark has two versions: Pascal VOC 2007 and 2012 [8]. Pascal VOC 2007 consists 20 classes with the total of 9963 images (train/validation/test) with 24,640 annotated objects. Pascal VOC 2012 has 20 classes with 11,530 images (train/validation/test) containing 27,450 annotated objects. The YOLOv2 model was trained using the PASCAL VOC 2007 trainval dataset and the PASCAL VOC 2012 trainval dataset. Pascal VOC 2007 dataset has four super classes: person, animal, vehicle, indoor. Our experiments focused on the object classes in the indoor environment, i.e., bottle, chair, dining table, potted plant, sofa, and tv monitor.

Local dataset: The dataset of 450 chair images, 450 potted plant images 450 ticket gate images and 450 table images are selected in the local areas. We use the input image resolution of 416 × 416 pixels. We used 12 chair images as the initial labeled data sample. The 98 images are selected randomly, and used for the validation dataset. The remaining 340 images of each class were used for an unlabeled dataset for the experiments. When trained using the PASCAL VOC dataset, the local dataset produces very poor detection results with YOLOv2, even it is state-of-the-art object detection technology [10].

### 5.2. Experiment parameter settings

Our object detection method used the evaluation of the PASCAL VOC challenge [36, 37]. This is applied where an average precision is computed by averaging the precision over a set of evenly spaced recall levels.

$$AveP = \frac{1}{11}\sum_{r\in\{0,0.1,\ldots 1.0\}} Pinterp^{(r)} \quad (10)$$

Here, $Pinterp^{(r)}$ is an interpolated precision that takes the maximum precision over all recall greater than r. We use the Intersection over Union (IoU) in order to calculate the overlap between two boundaries that ground truth and prediction. We mentioned in (**Fig. 7**), the red bounding box represents ground truth, the black bounding box indicates EER-ASSL, and the yellow bounding box represents the YOLOv2 VOC model. In our experiment we consider IoU threshold value predefine to be 0.5. (IOU=> 0.5). If the performance of prediction is over the threshold value we consider as correct otherwise it is considered incorrect. In order to get the best performance we apply the gradient based optimization method Adam and the stochastic gradient descent (SGD). For both of these optimizers we select the learning rate 0.001. However, the SGD optimizer is much slower in our experiment compared to Adam with 500 epochs as shown in (**Fig. 3**).

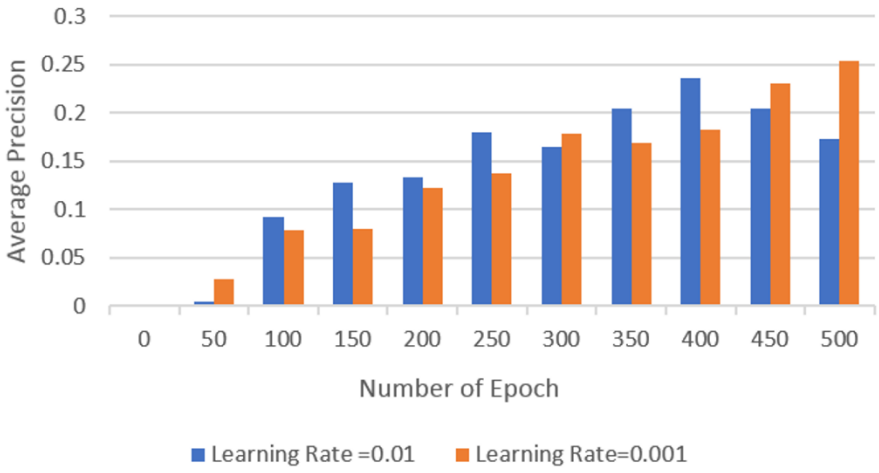


**Fig. 3.** Experiment result with different learning rate using Adam and SGD optimizer

**Fig. 3** represents the experiment results with various learning rates while training the local dataset using Adam and SGD optimizer. We select Adam optimizer with the learning rate 0.01 and 0.001 due to the fact that the learning rate 0.001 has a higher and more stable performance. Our training process is divided into several steps such as phase 1 and phase 2 [21]. Both SGD and Adam optimizer were used for training the different number of bins. The total number of the bins is 20 for two phases. Our experiment shows Adam optimizer to have a faster (time) convergence than the SGD optimizer with a higher AP. For this reason we selected Adam optimizer with the 0.001 learning rate. We divided the entire labeled dataset into two phases. In phase1, the parameter combination is as follows: uncertainty, diversity, and confidence [0.8, 0.8, 0.8]. Based on the performance, we change the parameter combination to either [0.8, 0.6, 0.8] or [0.8, 0.8, 0.6].

### 5.3 Effect of EER-ASSL

The performance of the proposed EER-ASSL, IASSL, and simple SSL are compared in **Fig. 4**. The collaborative sampling parameters are set by [0.8, 0.6, 0.8] for uncertainty, diversity, and confidence for all experiments. Adam optimizer was used with the learning rate 0.001.

One can notice that the EER-ASSL has much improved performance over the incremental ASSL. The simple SSL performance rarely improved.

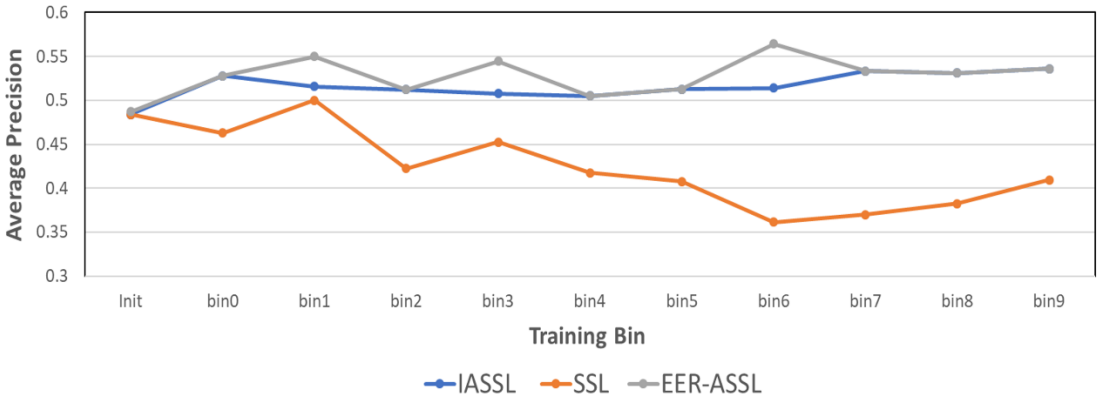


**Fig. 4.** The information flow of the rollback based ASSL

## 5.4 Testing on noisy local images

**Fig. 5.** Noisy image samples with labeling result

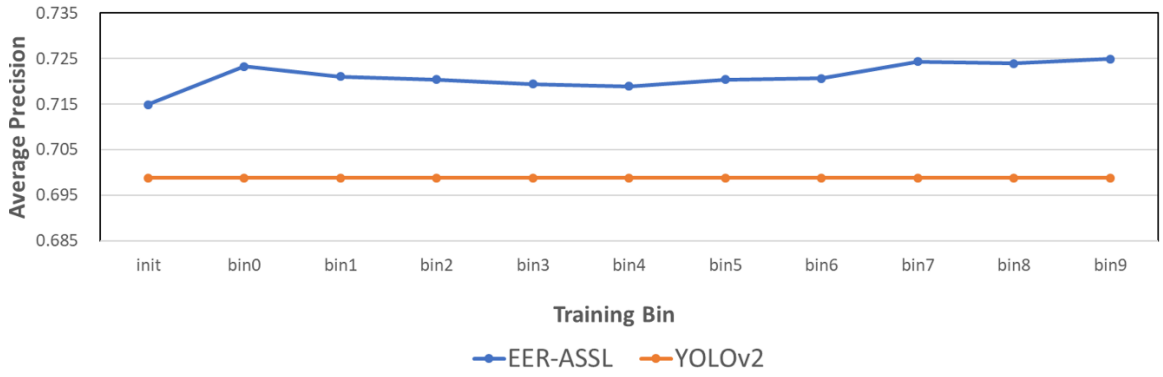


**Fig. 6.** The performance comparisons measured by the average precision (AP) of EER-ASSL using noisy local images

Using local noisy images (**Fig. 6**), we compared EER-ASSL, IASSL [21], and Iterative Cross learning (ICL) that shows much improvement on noisy images [22]. ICL is tested by image classification performance, while EER- ASSL is evaluated by object detection performance, where both the bounding box positions and the class labels are noisy. The learning rate of Adam optimizer is set 0.001. The experiment results are shown in **Fig. 3**. In the beginning, EER-ASSL shows low performance (AP) affected by the noisy data, but after the 4th bin of the first phase, it outperforms the other methods. The experiment results are reflected in **Fig. 7**. One can notice that EER-ASSL demonstrates outstanding performance under diverse illumination changes.

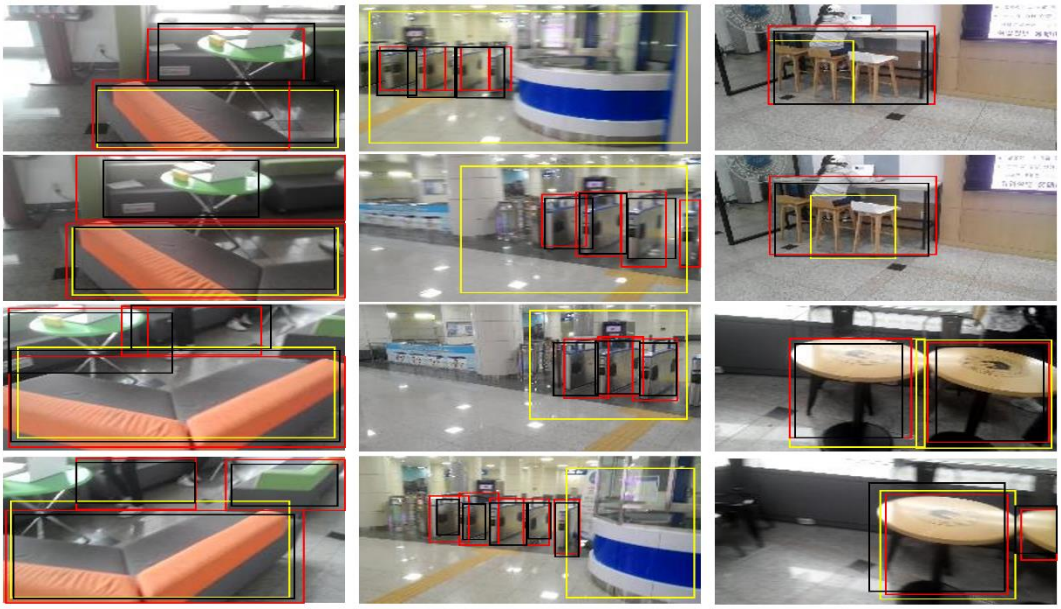

**Fig. 7.** The detection result from IASSL on local datasets of chairs, sofas, and tables. In all cases, the red bounding box represents ground truth, the black bounding box indicates EER-ASSL, and the yellow bounding box represents the YOLOv2 VOC model

### 5.5. Comparison with state-of-the-art technology

EER-ASSL is compared with several state-of-the-art object detectors. Each of four objects has 100 labeled and 300 unlabeled data samples, where local chair, sofa, and table images were mixed with the PASCAL VOC test data for fair evaluation. The detectors were trained with the same benchmark dataset and local dataset for a fair evaluation. **Table 1** shows the comparison results where each column indicates the composition ratio for both benchmark and local data.

**Table 1.** Performance comparison of state-of-the-art object detectors and EER-ASSL in terms of the mAP measure

| Method | (0,100)* | (10,90) | (25,75) | (50,50) | (75,25) | (90,10) |
|---|---|---|---|---|---|---|
| Faster RCNN | 67.7 | 67.9 | 68.1 | 68.3 | 68.9 | 69.3 |
| SSD300 | 68.4 | 68.8 | 69.1 | 69.3 | 69.8 | 70.3 |
| YOLOv2 | 70.1 | 70.4 | 70.6 | 71.1 | 71.5 | 71.5 |
| EER-ASSL | 71.9 | 71.7 | 70.0 | 69.9 | 68.6 | 67.6 |

*In the first row (a, b) indicates the composition ratio for all data, in which "a" represents VOC 2007 test data, and "b" represents the ratio of local data

YOLOv2 trains the network using the ImageNet [7, 39, 40] 1000 class classification dataset and then modifies the network in order to perform the detection. This jointly training classification and detection data is much larger than the local dataset which is limited in number to only 100 images for each class such as a sofa or a ticket gate. As a result, decreasing local training data has not much effect on the YOLOv2 model in our experiment. On the other hand, our EER-ASSL model already adapted the local data and if the composition data ratio is over 50 percent then the EER-ASSL outperforms other state-of-the-art methods that can be seen in the additional columns (10,90) and (25,75).

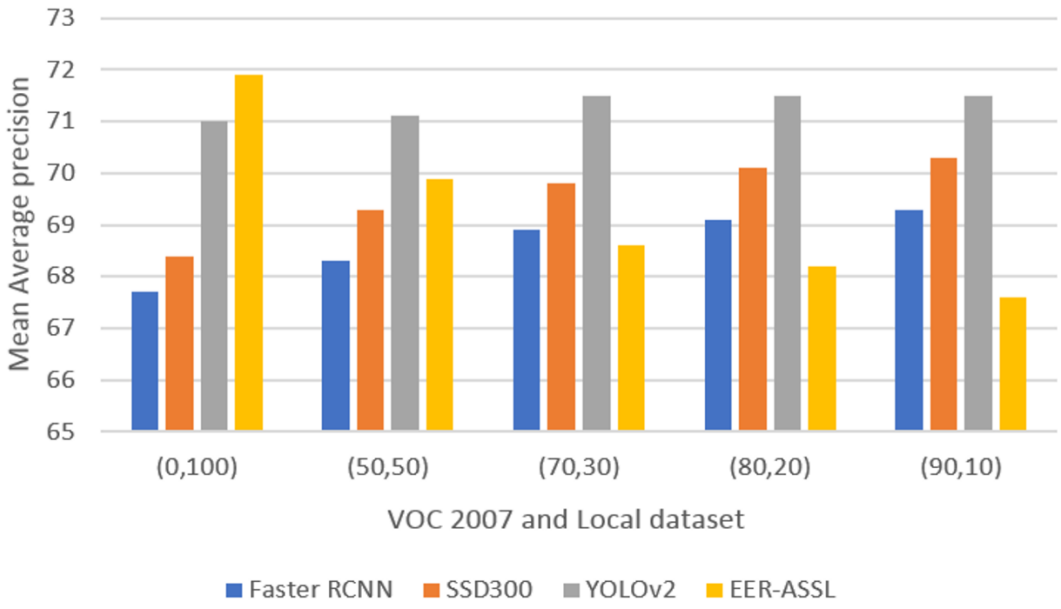


**Fig. 8.** Performance comparison of state-of-the-art object detectors and EER-ASSL in terms of mAP measure on benchmark datasets

The comparison of EER-ASSL, in terms of mAP, with state-of-the-art object detectors such as Faster RCNN, SSD 300, and YOLOv2 are shown in **Table 2**. The experiments were conducted similar to previous works considering incremental learning with Fast RCNN and

Faster RCNN [11, 36]. Similarly, we use our baseline detector as YOLOv2 using both local and benchmark dataset. Both PASCAL VOC 2007 test data and our local data are employed. The collaborative sampling parameters are set 0.8, 0.6, and 0.8 for uncertainty, diversity, and confidence, respectively. EER-ASSL shows great improvement from the other object detectors.

**Table 2.** EER-ASSL performance in terms of mAP measure on local dataset

| Method (local data) | mAP | Batch size | Trained-on | Speed (fps) | #Boxes | Input resolution |
|---|---|---|---|---|---|---|
| Faster RCNN | 67.7 | 1 | 07+12 | 5 | ~6000 | ~1000×600 |
| SSD300 | 68.4 | 8 | 07+12 | 59 | 8732 | 300×300 |
| YOLOv2 | 71.0 | 1 | 07+12 | 67 | 845 | 416×416 |
| Ours (EER-ASSL) | 71.9 | 1 | 07+12+local | 42 | 845 | 416×416 |

**Table 2** summarizes the mean average precision of the state-of-the-art methods on local test datasets. Our proposed EER-ASSL method shows improvement in performance in higher mAP results with new objects such as a sofa or a ticket gate in a similar environment. As shown in **Table 2**, our EER-ASSL method's adaptive property significantly improves the detection performance with a faster computational speed on the local dataset and its environment.

## 6. Concluding remarks

This paper presents EER-ASSL combining the ERR-based rollback learning and the bin-based SSL for a CNN object detector in the presence of noisy data distributions. The ensemble of ERR-based prediction model and CNN detector model achieves higher accuracy and requires less human effort, compared with state-of-the-art detectors. The EER learning method supports a rapid short-term myopic adaptation, and the CNN models an incremental long-term performance improvement. The future research direction is to build an adaptive and improved deep learning architecture by cooperating with the fast feed forward networks and the extreme learning machines to find ways to achieve a more flexible and fast adaptive learning sequence in noisy data distributions.

## Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A1B03935440). The GPUs used in this research were generously donated by NVIDIA.

## References

[1] I. Dimitrovski, D. Kocev, I. Kitanovski, S. Loskovska, and S. Džeroski, "Improved medical image modality classification using a combination of visual and textual features," *Computerized Medical Imaging and Graphics*, vol. 39, pp. 14-26, Jan. 2015. Article (CrossRef Link)

[2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no 4, pp. 541-551, 1989. Article (CrossRef Link)

[3] L. Tamas, R. Frohlich, and Z. Kato, "Relative pose estimation and fusion of omnidirectional and Lidar cameras," *Lecture Notes in Computer Science*, vol. 8926, pp. 640–651, Mar. 2015. Article (CrossRef Link)
[4] X. Zhang, S. Wang, and X. Yun, "Bidirectional Active Learning: A Two-Way Exploration Into Unlabeled and Labeled Data Set," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, Dec. 2015. Article (CrossRef Link)
[5] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," *arXiv Prepr. arXiv*, 2013. Article (CrossRef Link)
[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proc. of the 25th International Conference on Neural Information Processing Systems*, vol. 1, pp. 1097-1105. Article (CrossRef Link)
[7] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot Object Detection via Feature Reweighting," 2018. Article (CrossRef Link)
[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Bernstein, A. C. Berg, and L. FeiFei, "ImageNet Large Scale Visual Recognition Challenge," *International Jouranl of Computer Vision*, vol. 115, pp. 211-252, 2015. Article (CrossRef Link)
[9] M. Everingham and J. Winn, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Development Kit," *Challenge*, vol. 2007, pp. 1-23, 2007. Article (CrossRef Link)
[10] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," 2016. Article (CrossRef Link)
[11] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "cuDNN: Efficient Primitives for Deep Learning," pp. 1-9, 2014. Article (CrossRef Link)
[12] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental Learning of Object Detectors without Catastrophic Forgetting," in *Proc. of IEEE International Conferenc of Computer Vision*, 2017. Article (CrossRef Link)
[13] Z. Lu, X. Wu, and J. C. Bongard, "Active learning through adaptive heterogeneous ensembling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 2, pp. 368-381, 2015. Article (CrossRef Link)
[14] B. Settles, "Active Learning Literature Survey," Computer Science Report, University of Wisconsin, USA, 2009.
[15] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, p. 1070, 2018. Article (CrossRef Link)
[16] A. Sorokin and D. Forsyth, "Utility data annotation with Amazon Mechanical Turk," in *Proc. of 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1-8, 2018. Article (CrossRef Link)
[17] K. K. Singh and Y. J. Lee, "You reap what you sow: Using Videos to Generate High Precision Object Proposals for Weakly-supervised Object Detection," in *Proc. of IEEE Conference Computer Visions*, pp. 2219-2228, 2019 .
[18] Y. Yang and M. Loog, "Active Learning Using Uncertainty Information," *arXiv:1702.08540*, Feb. 2017. Aricle (CrossRef Link)
[19] S. Pang and X. Yang, "Deep Convolutional Extreme Learning Machine and Its Application," *Computational Intelligence and Neuroscience*, vol. 2016. Article (CrossRef Link)
[20] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto, "Unsupervised Moving Object Detection via Contextual Information Separation," *Computer Vision Foundation*, pp. 879-888, 2019.
[21] Article (CrossRef Link)
[22] Z. Chen, K. Wang, X. Wang, P. Peng, E. Izquierdo, and L. Lin, "Deep Co-Space: Sample Mining Across Feature Transformation for Semi-Supervised Learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2667-2678, Oct. 2018. Article (CrossRef Link)
[23] P. K. Rhee, E. Erdenee, S. D. Kyun, M. U. Ahmed, and S. Jin, "Active and semi-supervised

learning for object detection with imperfect data," *Cognitive Systems Research*, vol. 45, pp. 109-123, 2017. Article (CrossRef Link)
[24] X. Zhu, Semi-Supervised Learning Literature Survey, 2008. Article (CrossRef Link)
[25] D. K. Shin, M. U. Ahmed, and P. K. Rhee, "Incremental Deep Learning for Robust Object Detection in Unknown Cluttered Environments," *IEEE Access*, vol. 6, pp. 61748-61760, 2018. Article (CrossRef Link)
[26] J. Yuan, W. Zhang, H. S. Tai, and S. McMains, "Iterative cross learning on noisy labels," in *Proc. of 2018 IEEE Winter Conference on Applications of Computer Vision*, pp. 757-765, 2018. Article (CrossRef Link)
[27] X. Zhu, J. Lafferty, and Z. Ghahramani,"Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions," *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003. Article (CrossRef Link)
[28] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," in *Proc. of the 17th International Conference on Machine Learning*, pp. 999-1006, 2000. Article (CrossRef Link)
[29] D. Cohn, Z. Ghahramani, and M. I. Jordan, "Active Learning with Statistical Models," *Journal of Artificial Intelligence Research*, vol. 4, 1996. Article (CrossRef Link)
[30] K. Chaloner and I. Verdinelli, "Bayesian experimental design: A review," *Statistical Science*," vol. 10, pp. 237-304, 1995. Article (CrossRef Link)
[31] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904-1916, 2015. Article (CrossRef Link)
[32] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Proc. of European Conferece on Computer Visions*, vol. 9905, pp. 21-37, 2016. Article (CrossRef Link)
[33] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2015. Article (CrossRef Link)
[34] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, "Object Detection with Discriminatively Trained Part Based Models," *IEEE Transations on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1-20, 2009. Article (CrossRef Link)
[35] I. Muslea, S. N. Minton, and C. A. Knoblock, "Active learning with strong and weak views: A case study on wrapper induction," IJCAI Int. Jt. Conf. Artif. Intell., pp. 415-420, 2003. Article (CrossRef Link)
[36] J. Kwon and K. M. Lee, "Tracking of a Non-Rigid Object via Patch-based Dynamic Appearance Modeling and Adaptive Basin Hopping Monte Carlo Sampling," in *Proc. of 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1208-1215, 2009. Article (CrossRef Link)
[37] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, and C. Citro, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," ***arXiv:1603.04467*** **,** 2016. Article (CrossRef Link)
[38] C. Mitash, K. E. Bekris, and A. Boularias, "A Self-Supervised Learning System for Object Detection using Physics Simulation and Multi-view Pose Estimation," in *Proc. of International Conference on Intelligent Robots and Systems*, pp. 545-551, 2017. Article (CrossRef Link)
[39] G. Salton and M. J. Mcgill, Introduction to Modern Information Retrieval, New York, USA: McGraw-Hill Inc, 1986.
[40] M. Everingham , L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp 303-338. Article (CrossRef Link)
[41] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," pp. 1-15, 2017. Article (CrossRef Link)
[42] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," 2016. Article (CrossRef Link)

**Minhaz Uddin Ahmed** received his B.S. and M.S. degrees in Computer Science from the National University, Bangladesh, in 2006 and 2010. He obtained Ph.D in 2019 from the Inha University, south Korea. His research interests include Human Action Recognition, Facial Expression Recognition, Machine Learning and Deep Learning

**Yeong Hyeon Kim** received his B.S. degree in Computer Engineering from Inha University, Incheon, Korea in 2018. He completed his Masters' degree from Inha University in 2020. Currently he is pursuing PhD in computer engineering. His research interests include Object Detection, Tracking and Localization, Deep Learning, Machine Learning, and computer vision.

**Phill Kyu Rhee** received his B.S. degree in Electrical Engineering from the Seoul National University, Seoul, Korea, in 1982, an M.S. degree in Computer Science from the East Texas State University, Commerce, Texas, in 1986, and a Ph.D. degree in Computer Science from the University of Louisiana, Lafayette, Louisiana, in 1990. From 1982 to 1985, he worked as a research scientist in the Systems Engineering Research Institute, Seoul, South Korea. In 1991, he joined the Electronic and Telecommunication Research Institute, Seoul, South Korea, as a senior research staff member. From 1992 to 2001, he was an associate professor in the Department of Computer Science and Information Engineering of Inha University, Incheon, South Korea, and since 2001, he has been a professor in the same university and department. His current research interests are pattern recognition, machine intelligence, and autonomic cloud computing. Dr. Rhee is a member of the IEEE Computer Society and the Korea Information Science Society (KISS).