Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Incremental Learning Framework for Indoor Scene Recognition

Aram Kawewong
Department of Computer Engineering
Chiang Mai University
239 Huaykaew, Chiang Mai 50200 Thailand
e-mail: aram@eng.cmu.ac.th

Rapeeporn Pimpup and Osamu Hasegawa
Imaging Science and Engineering Laboratory
Tokyo Institute of Technology
4259 Nagatsuta, Midori, Yokohama 226-8503 Japan
e-mails: {pimup.r.aa,hasegawa.o.aa}@m.titech.ac.jp

Abstract

This paper presents a novel framework for online incremental place recognition in an indoor environment. The framework addresses the scenario in which scene images are gradually obtained during long-term operation in the real-world indoor environment. Multiple users may interact with the classification system and confirm either current or past prediction results; the system then immediately updates itself to improve the classification system. This framework is based on the proposed n-value self-organizing and incremental neural network (n-SOINN), which has been derived by modifying the original SOINN to be appropriate for use in scene recognition. The evaluation was performed on the standard MIT 67-category indoor scene dataset and shows that the proposed framework achieves the same accuracy as that of the state-of-the-art offline method, while the computation time of the proposed framework is significantly faster and fully incremental update is allowed. Additionally, a small extra set of training samples is incrementally given to the system to simulate the incremental learning situation. The result shows that the proposed framework can leverage such additional samples and achieve the state-of-the-art result.

Indoor place recognition has been an important navigation problem in both computer vision and robotics. One of the challenges in indoor place recognition is the complexity of the scenes, which is inevitable. Some places might be well-structured in their spatial properties (e.g., concert hall), while others are better characterized by the objects they contain (e.g., video store). Also, because the characteristics of an indoor scene are significantly different from those of outdoor scenes, a specifically designed approach is needed.

Many approaches had been proposed to solve this problem (Lazebnik, Schmid, and Ponce 2006; Torralba et al. 2003), but they were not designed for a wide range of indoor place categories. This motivated the study by Quattoni and Torralba (2009) which can deal with 67 categories of indoor scenes. The dataset was reported as the largest indoor-scenes dataset at the time. In this method, an individual scene category is represented by a set of scene prototypes. Each prototype is defined by a constellation model that consists of a single root and a set of regions of interest (ROIs). The root is described by the GIST descriptor (Oliva and Torralba 2001). This descriptor globally captures the holistic shape information of the entire image and cannot be moved. However, the ROIs can move slightly in accordance with the different positions of objects in different scene categories. Each ROI is described by a spatial pyramid of visual words (Sivic and Zisserman, 2003). The visual vocabulary must be created *a priori* and the ROI positions must be annotated manually. The learning model is based on gradient-based optimization.

Jia-Li et al. (2010) approached the problem differently. The so-called ObjectBank method was proposed to recognize indoor scenes by considering the objects found in the scene. Unlike the method of Quattini and Torralba (2009), the system is aware of the objects contained in the scene, as it uses a set of pre-trained object detectors to process an image and obtain a histogram of occurrences of objects in the scene. This method relates the concept of transfer learning (Lampert et al. 2009; Farhadi et al. 2009).

Recently, Pandey and Lazebnik (2011) used the deformable part-based model (DPM) with latent SVM (LSVM) to solve the problem. Unlike the original part-based model (Felzenszwalb et al. 2010), this method can find regions of interest (ROI) in the scene automatically without the need for annotation and has achieved state-of-the-art results on the MIT 67-category scene dataset.

All methods described above are based on an offline framework where assumptions about the fixed set of training and class labels hold. Alternatively, we are interested in the scenario where the system is required to be used with a limited set of image samples of the places at the beginning. More image samples will gradually be obtained later. For example, consider a servant robot that is placed in an unfamiliar large building. While some new image samples of known categories might be obtained during its operation, a new place category might need to be added. This information is useful and should be used to update the robot in an online incremental manner. Consequently, an online incremental framework, that *i*) allows the robot to add new place categories whenever needed and *ii*) updates the classifier by a new input image obtained as feedback from the actual classification result, is needed. In summary, this paper proposes two contributions:

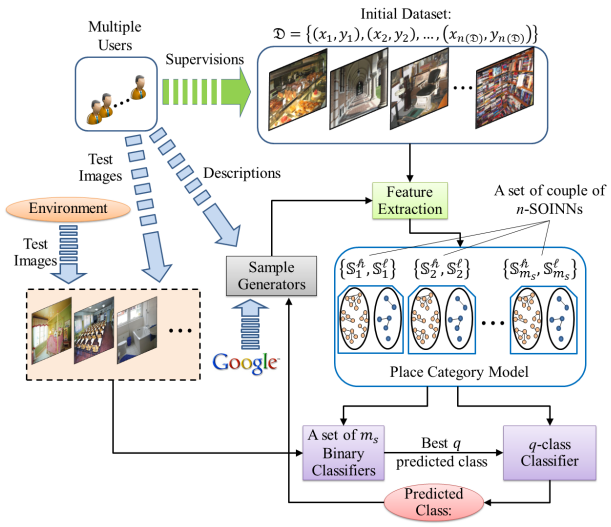

Figure 1: Proposed incremental framework for indoor scene recognition.

- A new online incremental framework for indoor scene recognition, which allows the machine to add new classes and update existing classes with new information at any time.
- A new method of incremental indoor place recognition, which fits the proposed framework well. The method is based on the proposed n-value self-organizing and incremental neural networks (n-SOINN) which has been modified from the original SOINN of Shen and Hasegawa (2006). It runs sufficiently fast for real-time update while retaining the same accuracy as those of offline state-of-the-art (SOA) methods.

The evaluation was performed on the MIT 67-category indoor scene datasets (Quattoni and Torralba 2009). The results are compared with the state-of-the-art results in many aspects including time and accuracy.

Incremental Place Recognition Framework

The proposed framework for incremental indoor place recognition is shown in Fig. 1. It focuses on the scenario where the results of the recognition are finally corrected by human experts and sent back to the system. Suppose a robot wanders in an unfamiliar environment and tries to classify a scene into an existing known category. After a prediction, it can question some people present in the scene about the category of the recent scene, or it can capture the scene and preserve the predicted label for confirmation with a human later. The policy of determination of predictions to be confirmed by a human can be created in various ways, randomly or starting with the most unreliable prediction. However, this study retrieves the prediction in a FIFO manner. The framework also supports the connection between the module and Google for additional data queries over the internet. For example, the predicted scene may be used in a query by the image search engine to obtain a set of similar images. This set can be shown to the user for selection of some more samples to update the category model. By the answer from human experts, the confirmed scene is fed back to the correct category as a new additional positive sample.

It should be clarified that the framework is especially appropriate for use in an environment with human presence. In particular, the robot lives with humans and can interact with them on any topic. The framework is not suitable for use in automated operation, where incremental human interaction is rare. Although the framework is appropriate for an environment with abundant human interactions, it also supports solo operation. The correction of the scenes predicted class is optional.

The framework runs in two main phases: initial learning and life-long learning. The former consists of learning on a dataset given in advance. In other words, the system learns the offline dataset in an incremental manner. This allows the system to learn sufficient information before progressing to real scene recognition. As shown in Fig. 1, the initial dataset $\mathbf{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n(\mathbf{D})}, y_{n(\mathbf{D})})\}$ is provided for initial learning. Our system learns these samples in an incremental manner.

In the second phase, the system continues the same process, but the input data is the test scene observed from the real environment (input comes from environment as shown in Fig. 1). This phase is supposed to be long-term. The system continues to classify input scenes into the classes with the highest probability score. For some (or all) classifications, the sample generator module tracks the scene and its predicted label. These tracks can be confirmed with a human and used to update the classifier. Note that human correction might be performed immediately by the user standing near the robot or later (sample generators pick up some past recognition and ask a human for the label). In either case, the feedback is useful and should be utilized efficiently.

The learning mechanism consists of two parts: *i*) category modeling and *ii*) classifier updating. The former aims to efficiently represent the category by using a limited set of training sample. The latter keeps updating the classifier (e.g., generating the new hyperplane) of the category by the representative samples obtained from the former part. The key success of this framework thus lies in the question; how to compress the input samples incrementally, keep them bounded for long-term running and use it to achieve good accuracy.

Proposed n-SOINN

The key to the framework is the clustering approach which compresses a large set of image feature vectors (i.e., 10,000 vectors) to a small set of vectors without causing a drop in accuracy. This clearly requires an incremental clustering mechanism.

Out of many existing methods of incremental clustering, we selected the SOINN (Shen and Hasegawa 2006) mainly because of its high computation speed. Conceptually, the clustering ability of incremental spectral clustering (ISC) is supposed to be better than that of SOINN, since

ISC minimizes intra-class distance and maximizes inter-class distance (Valgren and Lilienthal 2008). Nevertheless, this comes with high cost of computation as suggested by Tangruamsub et al. (2009). A self-organizing map (SOM) is similar to SOINN, but the former requires the number of nodes to be defined a priori (Kohonen 1990). MB-VDP (Gomes et al. 2008) is also an online clustering method, in which the number of clusters is determined by the data itself and the computing memory can be bounded. We have tested MB-VDP on the MIT 67-category indoor scene dataset and found that the computation time is far greater than that of SOINN. This could be because MB-VDP has an additional phase of data compression that needs to be iterated incrementally. Nevertheless, our future study includes an effort to update the MB-VDP to increase the computation speed so that it becomes applicable to our framework. We have also tried to use the standard SGD-SVM (Buttou and LeCun 2004) in our framework but the result was not good ($\sim$10% for testing set of Quattoni and Torralba (2009)). This could be because SGD-SVM is generally suitable for the problem of linear SVM whereas the most of SVMs used for indoor scene recognition are based on Radial-Based Function (RBF) kernel.

SOINN is an unsupervised clustering method, which can automatically determine the number of nodes and represent the topology of a multiclass dataset. Recently, some studies used it to cluster data in a labeled class in supervised learning (Kankuekul et al. 2012). Particularly, given a sequence of input vectors $(\mathbf{x}_1, ..., \mathbf{x}_t)$ obtained incrementally up to time t, where $\mathbf{x} \in \mathbb{R}^d$, starting from an empty set of nodes $\mathbb{A} = \emptyset$, first two input vectors are chosen as the two initial nodes of SOINN. Then, for every input vector $\mathbf{x}$, the first- and the second- nearest nodes are retrieved by the following equations:

$$s_1 = \text{argmin}_{c \in \mathbb{A}} dist\left(\mathbf{x}, \mathbf{w}_c\right) \quad (1)$$

$$s_2 = \text{argmin}_{c \in \mathbb{A} - \{s_1\}} dist\left(\mathbf{x}, \mathbf{w}_c\right) \quad (2)$$

where $\mathbf{w}_c$ is the d-dimensional weight vector of node c. Given d_1 and d_2 as the distances between $\mathbf{x}$ and the weight vectors of the nodes s_1 and s_2, respectively, if d_1 is greater than the threshold τ_{s_1} of the node s_1 or d_2 is greater than the threshold τ_{s_2} of the node s_2, a new node will be created and added to the set $\mathbb{A}$. The weight vector of the new node is $\mathbf{x}$. The threshold τ_c is the threshold value for each node $c \in \mathbb{A}$. Its value is initially set to infinity and gradually adapts according to all input vectors that are beaten by it in the comparison. On the other hand, if $d_1 < \tau_{s_1}$, then $\mathbf{x}$ is assigned to s_1. If the edge between s_1 and s_2 does not exist, it is created. The weight vector of the node $\mathbf{w}_{s_1}$ is updated by the learning rate $\epsilon(t)$

$$\mathbf{w}_{s_1} = \mathbf{w}_{s_1} + \epsilon(t)(\mathbf{x} - \mathbf{w}_{s_1}) \quad (3)$$

where t represents the number of input vectors over which the node s_1 has been a winner. The age of the new edge is set to zero. If an edge between s_1 and s_2 already exists, the age of this edge is incremented by one. Any edge whose age is greater than a predefined threshold age_{dead} is removed. This guarantees that any connection between nodes lasting longer than age_{dead} is cut. For every λ inputs, SOINN performs two actions: 1) checks for the node with the greatest number of accumulated wins (popular node) and separate it into two nodes, and 2) eliminates all isolated nodes. The first action expands the SOINN size at the popular nodes, while the second reduces the size by eliminating all non-popular nodes.

Within the context of the framework, we mention two disadvantages of the original SOINN. First the parameter λ is originally set for an unsupervised single SOINN. It is a global parameter that forces SOINN to perform garbage collection on non-popular nodes and extend the popular node for every constant period. This sometimes prevents the popular node from being separated into two nodes, even though the winning time is relatively high. The parameter models node popularity by considering the total number of inputs to SOINN rather than the winning time of each node. Therefore, to control the topology of SOINN, we introduce a new parameter n that forces any first winner node that wins more than n times to assign the win to the second winner node. If the second winner also has a winning time of more than n, a new node is generated. This parameter directly controls the output nodes of SOINN. In particular, setting $n = 0$ allows SOINN to preserve all features in their original values. In contrast, setting a high value of n makes the SOINN output only a small number of representative nodes while most input features are discarded. The latter case resembles the original SOINN. This SOINN having its topology controlled by the parameter n is referred to as n-SOINN.

Second, the basic Euclidean distance used in the original SOINN is for a single SOINN used for learning the topology of the entire multiclass dataset in an unsupervised manner. To use n-SOINN for supervised clustering, i.e. k SOINNs required for clustering k classes, the variance of data in each class could be different, which is considered by the Euclidean distance. In other words, it is tricky to compare the distance between nodes in different SOINNs. As a result, the distance function $dist$ between the input and the weight vector of n-SOINN is calculated by the standardized Euclidean distance (sEuclidean) instead of the normal Euclidean distance. The variance vector is calculated on the basis of all weight vectors of recent nodes existing in n-SOINN. This helps in balancing the priority of dimension which is from a different property of the scene. Note that n-SOINN is slightly different from other clustering methods such as k-means in the sense that raw data are not preserved by the system and a representative can be eliminated whenever it becomes non-popular.

Modeling Place Categories

In the initial phase of the framework, a dataset of m_s place categories is given as $\mathbf{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{n_s}, y_{n_s})\}$ where $\mathbf{x} \in \mathbb{R}^d$ is a d-dimensional image feature vector of the scene and y is a place category label, $y \in \mathcal{P} = \{p_1, ..., p_{m_s}\}$. A place category p_k is modeled by two n-SOINNs, one with a high (h) n-value $\mathbb{S}_k^h$ and another with a low (ℓ) n-value $\mathbb{S}_k^\ell$. Starting from an empty set of nodes, both n-SOINNs obtain pairs of an input vector and its label incrementally one by

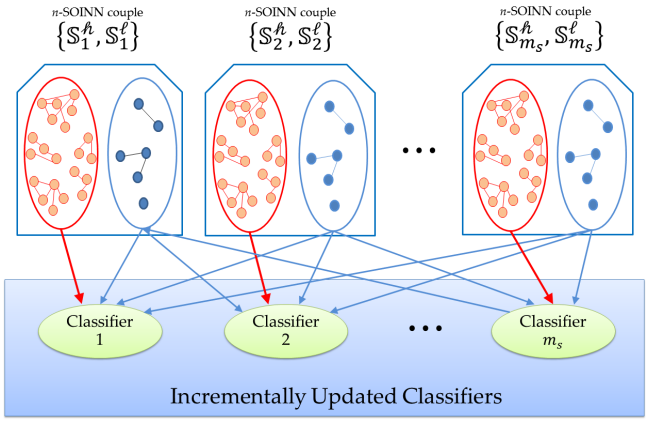

Figure 2: Connection between n-SOINNs and binary classifier of each individual place category.

one. If the input vector $\mathbf{x}_i$ is labeled as $y_i = p_k$, it is input to high value n-SOINN $\mathbb{S}_k^h$.

The n-values of the n-SOINNs are set differently in order to represent the category model in two different ways. As portrayed in Fig. 2, the place category is modeled by a set of representative nodes of the n-SOINNs. Since our framework needs a binary classifier for each category, positive and negative samples are needed. The positive samples are represented by a set of nodes in the n-SOINNs where n is low, whereas the negative samples are represented by a set of n-SOINNs with a high n-value. This is because the positive part is believed to capture accurate information about the category (low n-value, high density nodes), while the negative is believed to capture a more general non-accurate sample of all other disjoint classes (high n-value, low density nodes).

Modeling of place in this manner efficiently compresses the size of the nodes representing samples of each category. The output from this phase is used to obtain the classifier of each category as shown in Fig 2.

Learning and Recognition

An online incremental system must switch back and forth between learning and recognition all the time. The predicted result from recognition is sent to update the model and the classifier on user confirmation. The classifier used in this study is support vector machine (SVM) as its classification time is sufficiently fast. Note that although we use the SVM as the classifier, other methods should also be applicable to our framework since positive and negative data are all still available.

Learning the Classifier

Since n-SOINN can compress the number of nodes, immediate learning of the classifier becomes possible. We learn the SVM on the output nodes of n-SOINN to obtain the classifier as shown in Fig. 2.

For the category k, the positive samples are taken from the set of nodes $\mathbb{A}_k^\ell$ belonging to the n-SOINN $\mathbb{S}_k^\ell$. The negative samples are collected from the set $\mathbb{A}_j^h$ belonging to the n-SOINN $\mathbb{S}_j^h$ for all j where $0 < j < m_s, j \neq k$. To learn a new category, a given set of samples are input in $\mathbb{S}_{m+1}^\ell$ and $\mathbb{S}_{m+1}^h$. The next time recognition is requested, $m+1$ classifiers are created. Note that we select to perform the multiclass classification by using multiple binary SVMs scheme since it has been shown to be better than that of a single multiclass SVM (Duan and Keerthi 2005).

Recognizing the Scene

The recognition process in this framework consists of two steps. First, an input scene is given to all m_s classifiers according to m_s categories. This results in a set of pairs of elements formed from the predicted category z_j and the score θ_j from the j^{th} classifier.

$$\mathcal{Z} = \{(z_j, \theta_j) | \theta_j \geq \theta_{j+1}, 0 < j < m_s\}_{j=1}^{m_s} \quad (4)$$

The set is sorted by the score value. The first element is the predicted label with the highest score. However, this prediction is the binary classification. The discriminant function is based on the negative samples which have been compressed from the original set. It is possible that the first- and the second-best predictions might be unclear. As a result, we perform another multiclass classification among the q best-predicted labels. For the classification in this second step, the classifier is a single multiclass SVM trained by using the nodes from the n-SOINNs of each of the q best-predicted categories. The best-predicated label is the final answer of the recognition system. In particular, the input scene $\mathbf{x}$ is classified to the category p^*

$$p^* = \underset{p_i \in \{z_1, \ldots, z_q\}}{\arg\max} \; P(y = p_i | \mathbf{x}) \quad (5)$$

where $P(y = p_i|\mathbf{x})$ is the estimated probability of the predicted label from SVM. The probability is estimated by using the default method in LIBSVM (Chang and Lin 2011) which is implemented by the Platt scaling (Platt 2000) based on method of Lin et al. (2007).

Experimental Results

The evaluation of the proposed approach was performed on the MIT 67-category scene dataset. This dataset is considered as one of the most challenging sets for indoor scene recognition owing to its large number of categories.

The n-SOINN-SVM was run by using radial-based-function (RBF) with $C = 8$ and $\gamma = 0.0092$. These parameters were obtained by doing cross-validation on subset of trainings. For n-SOINN, we use n-value = 2 for high density n-SOINN and n-value = 100 for low density n-SOINN. The parameter q is set to 3. All testing was done on 1340 testing images of all 67 categories (same split with that of Quattoni and Torralba (2009)).

Exp. I: Comparison with Offline Baselines

The baselines used in this experiment are the offline methods using standard SVM with radial-biased function (RBF) with $C = 8$ and $\gamma = 0.001953$ according to that of used in CENTRIST (Wu and Rehg 2011). The image features used in this

	Method	Avg. positives per class	Avg. negatives per class	Update time for new sample (s)	Accuracy
Gist-Color	RBF-SVM-Bin	80	8280	132.89	32.69%
	RBF-SVM-Multi	5358		653.2925	31.42%
	n-SOINN-SVM	50.0448	350.6866	1.8297	29.33%
S-PACT	RBF-SVM-Bin	80	8280	107.0603	31.57%
	RBF-SVM-Multi	5358		477.4144	34.18%
	n-SOINN-SVM	64.3731	365.4627	2.1838	32.84%

Table 1: Comparison of accuracy (Bin = Binary, Multi = Multiclass)

Methods	Gist+ROI (Quattoni & Torralba 2009)	MM-Scene (Zhu et al. 2010)	CENTRIST (Wu & Rehg 2011)	ObjectBank (Li-Jia et al. 2010)	DPM (Pandey & Lazebnik 2011)	DPM+SP+GC (Pandey & Lazebnik 2011)	n-SOINN-SVM+GC (proposed)	n-SOINN-SVM+PACT (proposed)	n-SOINN-SVM+Comb. (proposed)	n-SOINN-SVM+Comb. (*Extended Learning*) (proposed)
Accuracy	26.5%	28.0%	36.9%	37.6%	30.4%	43.0%	29.3%	32.8%	41.3%	**45.0%**

Table 2: Comparison of accuracy with SOA methods

study are GistColor (Quattoni and Torralba 2009), S-PACT (Wu and Rehg 2011), and the combination between GistColor (GC) and S-PACT with slightly modifications. The dimension of the Gist-Color has been changed from 312 to 512, which resulted in better accuracy. For the S-PACT, unlike that of Wu and Rehg (2011), we used the same split as done by other state-of-the-arts (Zhu et al. 2010; Li-Jia et al. 2010; Pandey et al. 2011; Quattoni and Torralba 2009) so that the accuracy is slightly lower than that reported previously. These baselines were implemented by using two different schemes: set of 1-vs-all SVMs and K-class SVM. We did not use the Spatial Pyramid (SP) and Deformable-Part-based Model (DPM) because they are not appropriate for the framework. The SP requires an offline vocabulary. The DPM needs time for automatic part detection.

The result is shown in Table 1. n-SOINN-SVM achieves slightly lower accuracy (1-2%) than that of baselines. However, its computation time is significantly faster than the baselines. The SVM update time for any new incoming input image is about 2 s while the baselines are $>$ 100 s. The second and the third columns show, respectively, the number of positive and negative samples required for SVM training. Data size has been markedly compressed by n-SOINN with only a small drop in accuracy.

The accuracy was also compared to other SOA results in Table 2. To compete with these methods, we combine GC and S-PACT using Softmax transformation like that of Pandey and Lazebnik (2011). The method still slightly underperformed the method DPM+GistColor+SP (41.3% against 43%). However, it outperforms all other SOA methods in term of both time and accuracy. Note that the computation time for all SOA methods are supposed to be much higher than n-SOINN-SVM since they are based on standard SVM classification without any data compression.

Exp. II: Extended Learning

This experiment answers the question can the proposed method really utilize the feedback given by human experts to

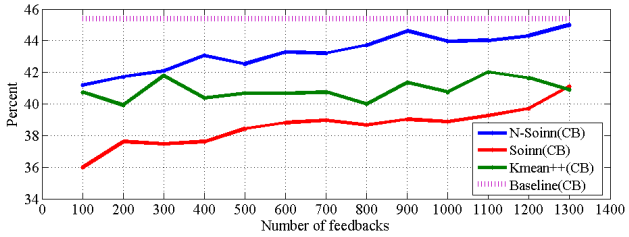

Figure 3: Increasing accuracy according to increasing number of feedbacks (CB: Combined Features)

outperform the SOA method?. Learning more images is not guaranteed to always increase the accuracy especially for the case of complex scenes. In this experiment, we ask two human experts to select image from Googles 20 images per class (totally 1340 images for 67 categories) and gradually give it to the system with correct label as if they are the feedback from real-world operation. These feedbacks are completely disjoint from the testing set. This experiment simulates the scenario where offline SOA methods cannot perform additional learning because their training consumes too long time. They cannot learn these additional image samples. At this point, it becomes interesting for us to see if the n-SOINN-SVM can leverage these feedbacks to outperform the SOA result.

By the result, randomly adding 1340 image samples, one by one, to the system, the proposed method gradually increased its accuracy to 45% as shown by Fig. 3. We also run the baseline using K-Class RBF-SVM on the full dataset (MIT-67 + Additional 1340 images) to obtain the upper boundary of how well these feedbacks can be utilized. Interestingly, n-SOINN-SVM obtained the same accuracy at 45%. This implies that the method can efficiently utilize the feedback. This result is also shown in Table 2 for comparison with other offline SOA. It should be emphasized that this does not show that our method outperform the SOA. Alternatively, it proves that the proposed framework is useful as it can successfully use its fast learning to learn on additional feedbacks and obtain the state-of-the-art result. Fig. 4 shows the graph of time required by each method to learn on feedbacks. By our framework, only one SVM is updated whenever new input is obtained. This saves amount of time for updating the whole m_s classifiers. However, we set to activate updating on all SVMs for every 100 input samples which results in a spike in the graph. This graph shows that n-SOINN-SVM is sufficiently fast for real-time update. The offline method, even with the update of only single SVM per one feedback, consumes too much time for use in an incremental manner.

Exp. III: Comparison with Incremental Baselines

This experiment was conducted to see if n-SOINN outperforms other clustering methods. For example, RBF-SVM-Bin in Table 1 can be used with k-means to reduce the size of positives and negatives. Thus, three different clustering methods were used to compare with n-SOINN. Each

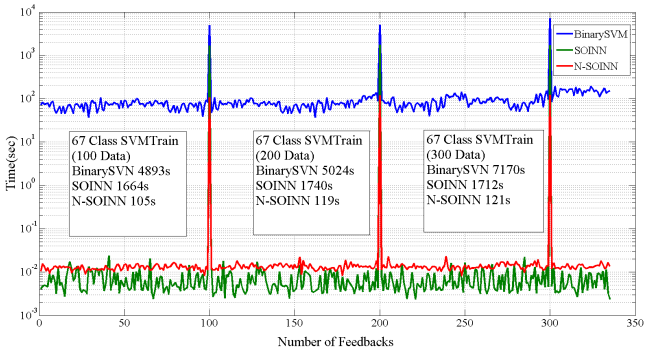

Figure 4: Computation time for incrementally update of every input feedback. The graph is obtained by using S-PACT feature.

	Method	Avg. positives per class	Avg. negatives per class	Update time for new sample (s)	Accuracy
Gist Color	RBF-SVM-*k*-means	60	330	2.7196	27.01%
	RBF-SVM-random	57.6716	396	1.6277	29.03%
	Original-SOINN-SVM	53.3543	2843	2801.4	31.19%
	n-SOINN-SVM	61.0448	373.3433	2.4204	**31.49%**
S-PACT	RBF-SVM-*k*-means	80.56	528	2.3689	31.87%
	RBF-SVM-random	82.597	584.76	1.805	31.72%
	Original-SOINN-SVM	61.4853	3589	3535.85	32.16%
	n-SOINN-SVM	80.6269	529.79	4.3876	**33.73%**

Table 3: Comparison of performance among different clustering methods

of them was used with the same classification scheme (1-vs-all SVM) and same parameters as that used in Exp. I. The first method is to use k-means++ of Arthur et al. (2007) with $k = k_1, k_2$ on positive and negatives respectively before using them to create SVM. The second method is to randomly select k_1 samples out of all positive samples and select k_2 samples out of all negative samples. The third method is the use of original SOINN to incrementally obtain positives over time and the representative nodes are used to create SVM as shown in Fig. 2. The number k_1 and k_2 were set equal to the number of resulting nodes in n-SOINN, which is different for each category. For original SOINN, we let the number of nodes grow automatically.

Table 3 shows the comparison between n-SOINN-SVM and the 1-vs-all scheme RBF-SVM by using different data compression methods. By the result, it is clear that n-SOINN can compress the training samples of both positive and negative samples more efficiently than that of k-means or random selection. Especially for the negative samples, n-SOINN selects nodes from the low density nodes of low n-value n-SOINN. This significantly reduces the size of negative samples that is usually the main factor for high cost of SVM update. For SOINN, the performance was not very different but the computation time of SOINN was much higher.

It is noteworthy that this experiment was tested on different dataset from that of the previous experiment so that the result of n-SOINN-SVM-GC and n-SOINN-SVM-PACT are different from those of Table 1. The dataset used hereby is the original MIT-67 Category dataset plus an extra set of

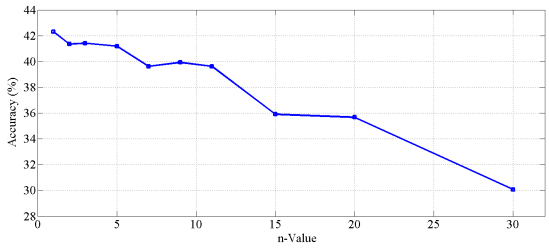

Figure 5: Effect of changes of n-value for low (ℓ) n-SOINN

images from internet (same as Exp. II). This is because all of the baselines in this experiment possess the ability of incremental learning. They can learn the extra dataset and leverage it to increase the accuracy. Therefore, we evaluated the accuracy by using all dataset to see if the proposed n-SOINN can efficiently utilize the feedbacks.

We also analyzed the effect of the change of n-value for low n-SOINNs as shown in Fig. 5. The graph shows that the increase of the n-value of low n-SOINN causes some drop in accuracy but the computation time can be speeded up. It should be noticed that the change of small n-value (i.e., $n < 6$) does not significantly decrease the accuracy while the learning time can be speeded up. For example, by changing the n-value to $n = 5$ causes 0.15% drop in accuracy but the update time of the classifier can be reduced by 0.04 s. This is useful for the life-long learning in large scale where the system must be updated frequently. That is, the graph suggests that n-value of the low n-SOINN, which represents the positive samples, can be changed between 0 to 5. For high value n-SOINN (used for representing negative samples), we found that higher n-value offers better accuracy. This is favorable since the great n-value results in the very compact size of network.

Conclusion

We have proposed an incremental framework for learning on indoor scene categories. The framework supports the incremental interactions with humans which provide feedbacks to the system for extended learning at any time. The proposed n-SOINN-SVM performs fast incremental learning which make it fit to the framework. It offers the high accuracy on par with that of SOA method while being capable to learn additional feedback from human experts. By leveraging feedbacks, the method outperform all SOA results and shows that incremental learning framework is essential since learning on more feedback can yield the better accuracy than that of well-trained offline system. Although the method has been proposed for scene image, it can also be applied to other computer vision applications using more appropriate image features.

Acknowledgement

This research was supported by Japan Science and Technology Agency (JST) CREST project and by TRF-CHE research grant for new scholar of Thailand Research Fund (TRF).

References

Arthur, D., and Vassilvitskii, S. 2007. k-means++: the advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA),* 1027–1035.

Buttou, L., and LeCun, Y. 2004. Large scale online learning. In *Proceedings of Advances in Neural Information Processing Systems (NIPS),* 217–224.

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology,* 2(3): 27.

Duan, K., and Keerthi, S. S. 2005. Which is the best multiclass SVM method? An empirical study. In *Proceedings of the 6th International Workshop on Multiple Classifier Systems,* 278–285.

Farhardi, A., Endres, I., Hoiem, D., and Forsyth, D. 2009. Describing objects by their attributes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 1778–1785.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. 2010. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI),* 23(9): 1627–1645.

Gomes, R., Welling, M., and Perona, P. 2008. Incremental learning of nonparametric Bayesian mixture models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 1–8.

Kankuekul, P., Kawewong, A., Tangruamsub, S., and Hasegawa, O. 2012. Online incremental attribute-based zero-shot learning. In *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 3657–3664.

Kohonen, T. 1990. The self-organizing map. *Proceedings of the IEEE,* 78(9): 1464–1480.

Lampert, C. H., Nickisch, H., and Harmeling, S. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 951–958.

Lazebnik, S., Schmid, C., and Ponce, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2169–2178.

Lin, H.-T., Lin, C.-J., and Weng, R. C. 2007. A note on Platts probabilistic outputs for support vector machines. *Machine Learning,* 68: 267–276.

Li-Jia, L., Hao, S., and Fei-Fei, L. 2010. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *Proceedings of Advances in Neural Information Processing Systems (NIPS),* 1378–1386.

Oliva, A., and Torralba, A. 2001. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision (IJCV),* 42(3): 145–175.

Pandey, M., and Lazebnik, S. 2011. Scene recognition and weakly supervised object localization with deformable part-based models. In *Proceedings of IEEE Interntaional Conference on Computer Vision (ICCV),* 1307–1314.

Platt, J. C. 2000. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classifiers,* MIT Press.

Quattoni, A., and Torralba, A. 2009. Recognizing indoor scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 413–420.

Shen, F., and Hasegawa, O. 2006. An incremental network for on-line unsupervised classification and topology learning. *Neural Networks,* 19(1): 90–106.

Sivic, J., and Zisserman, A. 2003. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of IEEE International Conference on Computer Vision (ICCV),* 1470–1477.

Tangruamsub, S., Tsuboyama, M., Kawewong, A., and Hasegawa, O. 2009. Mobile robot vision-based navigation using self-organizing and incremental neural networks. In *Proceedings of International Joint Conference on Neural Networks (IJCNN),* 3094-3101.

Torralba, A., Murphy, K., Freeman, W., and Rubin, M. 2003. Context-based vision system for place and object recognition. In *Proceedings of IEEE International Conference on Computer Vision (ICCV),* 273–280.

Valgren, C., and Lilienthal, A. 2008. Incremental spectral clustering and seasons: Appearance-based localization in outdoor environments. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA),* 1856–1861.

Wu, J., and Rehg, J. M. 2011. CENTRIST: a visual descriptor for scene categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI),* 33(8): 1489–1501.

Zhu, J., Li-Jia, L., Fei-Fei, L., and Xing, E. P. 2010. Large margin learning of upstream scene understanding models. In *Proceedings of Advances in Neural Information Processing Systems (NIPS),* 2586–2594.