Location Recognition Combining In/Outdoor Classification and Boosted Classifiers

Bing Liao, Ko-Tung Lin, Yu-Hui Chen, Yi-Husan Tsai

I. Motivation and Objective:

Image Location recognition is to recognize the location of a given image. A possible application for this technique is utilizing a large-scale image database to recognize one's current location by simply taking a photo of the sight around him, such as a street corner or a storefront. This technique is also widely applied in computer vision area, especially for locating mobile robotics.

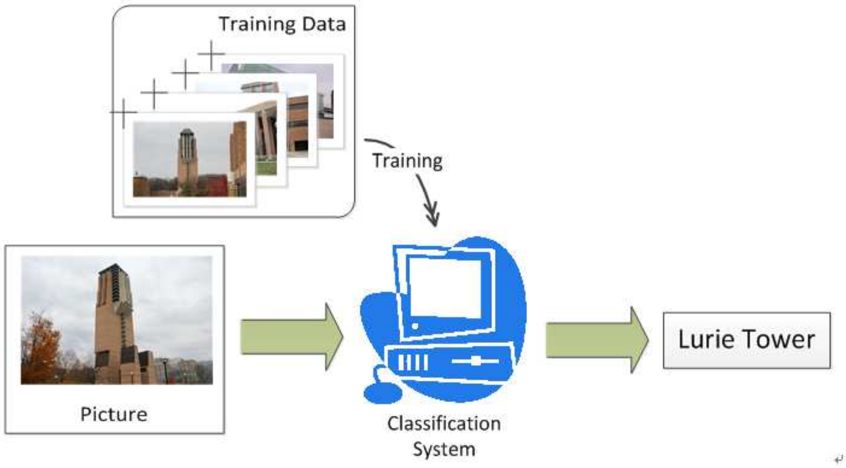


Figure 1. classification system concept

In this project, our goal is to develop a robust location recognition system focusing on University of Michigan North Campus. As shown in figure 1, given a picture of an indoor/outdoor place (ex. Lurie Tower), the classification system could identify the user's current location. Mathematically, our project could be restated as the following:

Given an input image x, which can be represented by its feature space $\mathcal{F}$, a well-trained location recognition system $f: \mathcal{F} \rightarrow \{1,2,\ldots,L\}$, where $\{1,2,\ldots,L\}$ are the labels of locations, can predict the location of this image by $f(x)$ (Figure 1).

II. Background and Related Work:

Location recognition is a fundamental task for autonomous robots. Several different approaches have been proposed in related topics. A broad class of techniques, "metric SLAM" (SLAM, Simultaneous Localization and Mapping), approach the problem by jointly maintaining an estimate of the pose of the robot and a set of map landmarks using an Extended Kalman Filter or particle filter [10]. However, it is common for these techniques to fail when a robot revisits a previously seen location after conducting a long traverse through unexplored terrain, which is known as the "loop closure problem" [9].

The appearance-based navigation has been developed as a solution to the problem of loop closure detection. For indoor location recognition, [8] has introduced a SVM-based visual classifier fusion method. They extracted Scale-invariant Feature Transform (SIFT) [13]/color SIFT, MPEG-7 features and some additional features, and apply several C-SVM classifiers to realize the feature-wise classification. After that, they fused the outcomes of feature-wise detectors based on their geometric mean. To realize multi-class classification, they used “one-against-all” SVM classifier. However, in [11] Hsu and Lin provided more multi-class SVM methods, such as “one-against-one” and Directed Acyclic Graph SVM (DAGSVM), which might show better performances on classification.

For outdoor location recognition, [9] outlines a complete probabilistic framework for this task, which is applicable even in visually repetitive environments where many locations may appear identical. In this paper, the image was processed with the Speeded-Up Robust Feature (SURF) [14] feature detector. Cummins and Newman developed a probabilistic model on top of the Bag-of-Words (BoW) representation which allowed them to define appearance-based navigation as recursive Bayesian filtering problem. Moreover, they introduced a noisy detector model to account for the unreliable nature of visual feature detection, which outperforms the standard term frequency-inverse document frequency (tf-idf) ranking.

To achieve better recognition performance, the indoor/outdoor scene classification can be introduced before performing location recognition. Scene classification is a challenging task which contains uncertain variability such as illumination and scale conditions or occlusions. Existing approaches can be categorized in three groups in the literature. The first method is mainly developed on content-based images [1, 2], using low-level features (color, texture, etc.) extracted from the whole image. This approach focused on dealing with binary classification such as indoor/outdoor or city/landscape. Another approach is focusing on using intermediate representation before classifying scenes [3, 4]. Although this strategy can be applied to a large number of scenes categories, it requires tedious works on labeling images with respect to local and global properties. The third approach is based on BoW models [5, 6, 7], which can incorporates classifiers in discriminative fashion or hierarchical Bayesian models. An advantage compared to two previous methods is that BoW model can be well integrated to catch meaningful scene components by extracting local features, and can also apply to a wide range of scene categories.

III. Proposed Innovative Method and Hierarchy:

Figure 2 presents the structure of our recognition system. We realize the location recognition (classification) by two stages: given a testing image, the first stage classifies it to either the indoor or outdoor class; in the second stage, we extract several features (histograms, SIFT[13], SURF[14]) and implement multiple classifiers to realize feature-wise classification, then apply boosting algorithm[17] to fuse the outcomes of these classifiers to achieve the final classification result.

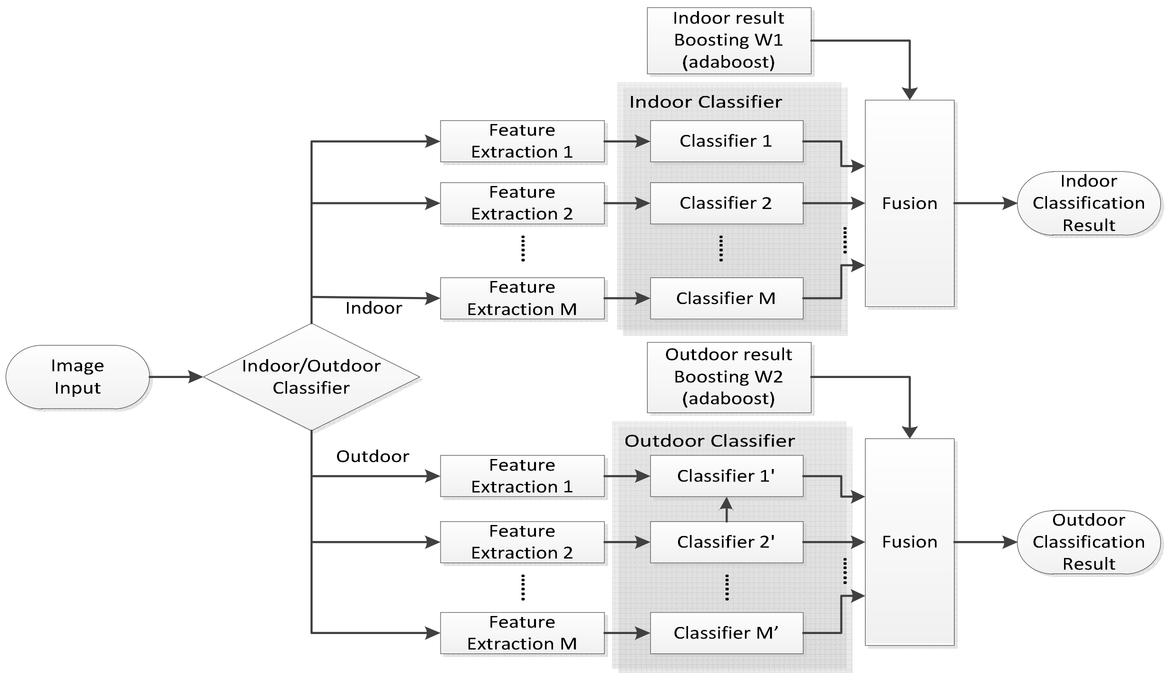


Figure 2. system structure

Our innovations can be summarized as the following:

1. *Two stage classification strategy:*

Our motivation to realize the classification by two stages is due to the fact that the same feature extracted from an indoor image and an outdoor image show different performances in classification. For example, SURF is good at catching local details in an image. Therefore, it can capture most of the characters of the objects in the indoor images and provides accurate classification. However, when extracting SURF from outdoor images, it catches many details of tree leaves or grass (noise), so the classification result will be affected by these noises.

2. *Combining different classification methods by Boosting algorithm*

For feature-wise classifiers, SVM classifiers are widely used in location/object recognition. We apply the “one-against-one” SVM to realize the multi-class SVM, which gives better performance than the “one-against-all” SVM [11]. Moreover, we bring the concept of BoW into Machine Learning and implement/modify the Nearest Neighbor (NN) and Naïve Bayes classifiers on top of it. (Details will be given in Methodology Chapter)

We select boosting algorithm to fuse the feature-wise classifiers and therefore generate the final classifier. Considering the fact that those feature-wise classifiers show different classification performance, it is instinctive to give them different boosting weights based on their importance (correctly classification rate). After we train their boosting weights, we hold a weighted vote to determine the final classification result. Related work [8] has stated a fusion method by computing the geometric mean of SVM outputs. However, the geometric mean is meaningful only if we apply on the same classification method (ex. all SVMs), while Boosting algorithm can apply on different classification methods. We will further prove that Boosting outperforms the geometric mean method.

IV. Methodologies used and developed:

A. Classifiers for image recognition:

Multi-class SVM:

Since SVM was originally designed for binary classification, in our project we need to apply some multi-class SVM methods to realize multi-class classification. Based on [11], we choose the "one-against-one" SVM classifier. For L classes, this method uses L*(L-1)/2 SVM classifiers, where each of them is trained on data from two classes. For training data from the ith and the jth classes, we solve the following binary classification problem:

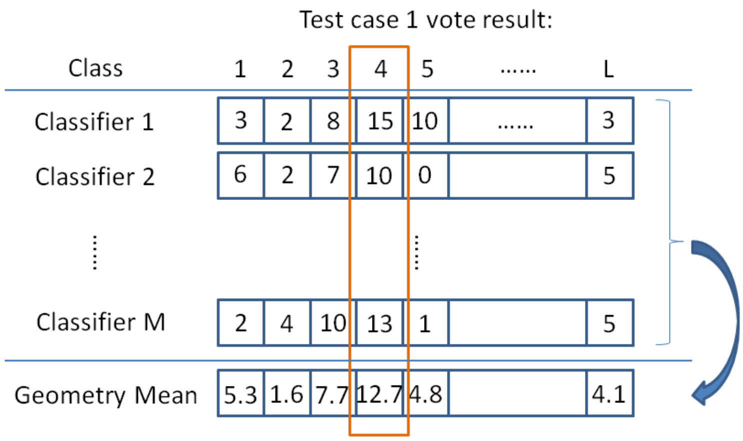
$$\min_{w^{ij}b^{ij}\xi^{ij}} \quad \frac{1}{2}\left(w^{ij}\right)^T w^{ij} + C\sum_{t=1}^{n}\xi_t^{ij}\left(w^{ij}\right)^T \qquad --(1)$$

$$s.t. \quad \left(w^{ij}\right)^T \Phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, if\ y_t = i;$$

$$\left(w^{ij}\right)^T \Phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, if\ y_t = j; \xi_t^{ij} \geq 0.$$

Each classifier checks $sign\{\left(\mathrm{w}^{\mathrm{i}}\right)^{\mathrm{T}}\Phi(\mathrm{x}) + \mathrm{b}^{\mathrm{i}}\}$ to vote for either class i or class j. Then we predict testing data x will be assigned to the class with the largest vote.

For each "one-against-one" SVM classifier, the confidence value of testing data for each class is set by the vote result for that class (as shown in figure 3). We then can further compute the geometric mean (to be compared with our own method) of those confidence values from different classifiers for each class. The final classification result is chosen to be the class with the largest geometry mean.



>> SVM-GM classification result for Test case 1 is 4

Figure 3. SVM geometry mean

Bag-of-word (BoW):

BoW model is a dictionary-based method which was first used to represent documents by considering each document as a "bag", which contains many words from the dictionary (codebook). By using the similar idea for image representation, BoW has been widely used in computer vision field, especially for object categorization. Therefore, the image can be considered as a document, and features extracted from the image are "words".

To represent the image as the BoW model, a widely-used method is to extract the SIFT and/or SURF as the image detector/descriptor. After extracting interesting points (features) from the training images, we convert feature vectors into "words" (codewords). One popular conversion method is performing mean-shift clustering [12] over all the feature vectors. Therefore, each "word" is defined as the centroid of the cluster. After applying mean-shift clustering algorithm over the training set, a codebook containing "words" can be learned and generated for the future testing process.

The Mean-shift clustering algorithm seeks a mode or local maximum of density of a given distribution. Contradicted with K-means, it is a non-parametric clustering technique which does not require prior knowledge of the number and shape of clusters. Given n data points x_i, i=1,…,n on $\mathbb{R}^d$, some kernel function g and a bandwidth (window radius) parameter h, define the window centroid at time t as x^t, then the mean shift can be expressed as:

$$m_h(x^t) = \frac{\sum_{i=1}^{n} x_i g(\frac{\|x^t - x_i\|^2}{h})}{\sum_{i=1}^{n} g(\frac{\|x^t - x_i\|^2}{h})} - x^t \qquad --(2)$$

For simplicity, we can apply the flat kernel, where $\mathrm{g(x)} = \mathbf{1}\{\|x\| \leq 1\}$. After initialization, the mean shift procedure is obtained by iteratively doing:

1) Compute the mean shift vector $m_h(x^t)$;
2) Shift the translation window centroid by $m_h(x^t)$, i.e., $x^{t+1} = x^t + m_h(x^t)$;

until it converges (i.e. $m_h(x^{t+1}) \approx m_h(x^t)$). After all data points are clustered, we collect the centroids of clusters to be the "words" and therefore construct the BoW model.

For those "words", since they may appear in multiple classes for multiple times, we treat them with different importance. Here, we compute the term frequency-inverse document frequency (tf-idf) weight for each "word". For any word w and class l, the term frequency $tf(w,l)$ is defined as the number of times that w appears in class l; the inverse document frequency is defined as $(w) = log \frac{L}{|\{l:w\in l\}|}$, where L is the total number of classes and $|\{l: w \in l\}|$ represents the number of classes in which w appears. Then,

$$tf\text{-}idf(w,l) = tf(w,l) \times idf(w).$$

To classify testing images based on the codebook learned from BoW model, methods can be divided into two categories. One is discriminative model such as SVM or Nearest Neighbor, and another is generative model such as Naïve Bayes or hierarchical Bayesian models.

The first approach we choose to classify images is the Nearest Neighbor (NN) algorithm. By extracting features using SIFT descriptor, a codebook is learned and each training image can be represented as a histogram of different "words". Given a testing image, we can also extract SIFT features and compute the histogram based on the "words" from the training stage. Since we may have a huge number of "words" and the dimension of the histogram will be large, we use k-d tree [21] data structure for efficiently searching the nearest neighbor over all histograms in training images. Therefore, the testing image will be classified as the class which its nearest neighbor belongs to.

We modify the Naïve Bayes Classifier by applying tf-idf weight as our second approach. Let w_1, w_2, … w_K denote the "words" in the codebook, then for each class l, we define $n_{kl} = tf(w_k, l)$. Then, the estimation of Pr(w = w_k|l) is:

$$\hat{g}_l(w_k) = \frac{n_{kl} \times idf(w_k) + 1}{\{\sum_{i=1}^{K} n_{il} \times idf(w_i)\} + K}$$

For a testing image, suppose it's extracted m SURF vectors, then for each feature vector s_i, i=1,…,m, we can find its nearest "word" w_j, $j \in \{1, \dots, K\}$, where K is the number of words in the codebook. If the distance between s_i and w_j is less than some threshold, then we say s_i is a valid feature vector and consider it as word w_j, i.e. $\hat{s_i} = w_j$. Therefore, the Naïve Bayes classifier is: $\hat{f}(x) = \arg\max_{l} \hat{\pi_l} \prod_{\substack{i=1 \\ valid}}^{n} \hat{g}_l(\hat{s_i})$, where $\hat{\pi_l} = \frac{|number\ of\ training\ images\ in\ class\ l|}{|number\ of\ training\ images|}$.

B. Classification fusion by AdaBoosting:

The AdaBoost[17] is an iterative procedure that combines many weak classifiers to approximate the Bayes classifier $C^*(x)$. In our case, the weak classifiers are the SVMs and BoW-based classifiers as described before. The goal for using Boosting is to learn a set of weights for combining the classification results from these weak classifiers and therefore achieve higher recognition accuracy.

In this project, the images will be classified into 17 classes. Therefore, we apply the modified multi-class AdaBoost algorithm with Stagewise Additive Modeling using a Multi-class Exponential loss function (SAMME) [18]. Given the same setup as that of AdaBoost and define M as the number of weak classifiers, SAMME proceeds as follows:

Algorithm:

1. Initialize the observation weights $w_i = 1/N, i = 1,2, \dots, N$
2. For m=1 to M:
   (a) Fit a classifier $T^{(m)}(x)$ to the training data using weights w_i.
   (b) Compute: $err^{(m)} = \sum_{i=1}^{N} \mathbf{1}\{l_i \neq T^{(m)}(x_i)\} / \sum_{i=1}^{N} w_i$.
   (c) Compute: $\alpha^{(m)} = \log\{(1 - err^{(m)}) / err^{(m)}\} + \log(L - 1)$
   (d) Update: $w_{i+1} \leftarrow w_i \exp\left(\alpha^{(m)} \mathbf{1}\{l_i \neq T^{(m)}(x_i)\}\right), i = 1, \dots, N$
   (e) Re-normalize w_{i+1}

Notice that this algorithm is very similar to multi-class AdaBoost with only the difference in 2(c). For original multi-class AdaBoost, we have $\alpha^{(m)} = \log \frac{1-\mathrm{err}^{(m)}}{\mathrm{err}^{(m)}}$, which means that each classifier is required to have an accuracy better than 1/2 for $\alpha^{(m)}$ to be positive. Now with modified formula 2(c), we only need the accuracy of each weak classifier be better than random guessing which is much easier to be achieved.

Unlike most weak classifiers that usually be used in AdaBoost, our classifiers here have to be "trained" before it can classify the data to get $T^{(m)}(x_i)$ needed in this algorithm. To solve this problem, we use 3-fold cross validation. The N training data are divided into 3 sets: A, B, C. Then we successively consider one of these sets as the hold-out set and train the classifier by using the other two.

Once we find the weight for each classifier, denoted as $\{\alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(M)}\}$, the output of the fusion classifier is determined by:

$$f(x) = \arg\max_{l} \sum_{m=1}^{M} \alpha^{(m)} \mathbf{1}\{l = T^{(m)}(x)\}$$

V. Experiments:

A. Experiment setting

We have collected 414 pictures of north campus and labeled them by 17 different locations. (shown in table-1). The first ten locations are outdoors while the others are indoors. Each location has about twenty five pictures taken from different angles and different distances.

Outdoor									
1	2	3	4	5	6	7	8	9	10
Duderstadt	Stamp	Lurie Tower	CSE	EECS	GG Brown	Nuclear	IOE	FXB	Pierpont

Indoor						
11	12	13	14	15	16	17
Book shelf	Computers	Information Desk	Mujo	Piano Lounge	U go's	Quizno

Table-1. Indoor Outdoor location label

The system is expected to be trained by 323 images and evaluated by the rest 91 testing data. The training set is further separated into 3 sets of size 107, 110, 106 to be used in 3-fold cross validation. Each testing image is assigned as one of the seventeen classes and compared with the ground truth. The result is evaluated by the following formula: $\mathrm{Accuracy} = \frac{N_{correct}}{N_{total}}$, where $N_{correct}$ is the number of correctly classified images and N_{total} is the total number of testing images.

For weak classifiers, we choose twelve different features for SVM and SIFT/SURF for two BoW-based classifiers (see table-2). These features are extracted by the tools [19]. For SVMs, we use the tools LibSVM [20] with linear kernel, and set parameter C (in formula (1)) equal to 1. In BoW implementation, we choose the bandwidth parameter h = 0.5 (in formula (2)) for mean-shift algorithm and obtain a codebook of 1683 "words" for SURF and 961 "words" for SIFT.

SVM1	RGB histogram	SVM8	SIFT
SVM2	Opponent histogram	SVM9	Hue SIFT
SVM3	Hue histogram	SVM10	HSV SIFT
SVM4	Nrg histogram	SVM11	Opponent SIFT
SVM5	Transformed Color Histogram	SVM12	RG SIFT
SVM6	Colormoments	BOW (Naive)	SURF
SVM7	Colormoment invariants	BOW(NN)	SIFT

Table-2. Weak classifiers and features

B. Results Comparison

One stage v.s. Two stages

The results of indoor/outdoor classification are shown in table-3. As you can see that very high accuracy can be obtained in indoor/outdoor classification by most of the classifiers we have. We simply chose the BoW (NN) with the highest accuracy (%100) to be the classifier for our first stage in the system.

For two stages strategy, all these classifiers are trained by either indoor or outdoor training images. Once the testing data is labeled as indoor/outdoor from the first stage, these classifiers can further classify the testing data within those indoor or outdoor labels. For example, if the testing data is classified as "indoor", then we use the classifiers which are trained by only indoor training data and classify it into one of the seven indoor locations.

SVM1	%95.6	SVM8	%97.8
SVM3	%97.8	SVM11	%95.6
SVM6	%98.9	BOW(NN)	%100

Table-3 First stage accuracy

Table-4 compares the accuracies of each weak classifier between one-stage and two-stage strategy. We also included the results of "One-Vector(OV)" and "Geometric Mean(GM)". OV method means that we concatenated all features into one long vector and then classify by using single SVM. GM is combining twelve SVM classifiers by calculating the geometric mean of their confidence values ([8] applies this method). We can see the accuracy improved in every classifier by introducing our two-stage strategy.

	One stage	Two stages		One stage	Two stages
SVM1	%65.93	%68.13	SVM7	%32.97	%38.46
SVM2	%70.33	%70.33	SVM8	%61.54	%61.54
SVM3	%58.24	%59.34	SVM9	%69.23	%71.43
SVM4	%61.54	%62.64	SVM10	%83.52	%84.62
SVM5	%51.65	%53.85	SVM11	%82.42	%82.42
SVM6	%67.03	%69.23	SVM12	%64.84	%67.03
BOW (Naive)[9]	%70.33	%74.73	BOW(NN)	%82.42	%85.71
SVM-OV	%85.71	%86.81	SVM-GM[8]	%86.81	%89.01

Table-4 One-stage v.s. Two-stage strategy

Uniform Weights v.s. AdaBoosting

From previous results we can see that some of the features/weak classifiers show bad performance in location recognition. Given these classifiers, we want to figure out an effective way to combine them and thus further improve the accuracy upon existing results.

Here we show the classification accuracies using "uniform-weighted" and "AdaBoosting-weighted" fusion methods. The classifier's weights for the former method are all equal to one, while these weights for the latter method are learned by AdaBoosting algorithm as introduced in section IV.

	SVM only	SVM + BOW
Uniform Weights	%84.62	%89.01
AdaBoosting Weights	%89.01	%90.11

Table-5 Uniform weights v.s. AdaBoosting

Note that "SVM only" in table-5 stands for the fusion over only SVM classifiers, while "SVM+BoW" fuses both SVM classifiers and BoW-based classifiers. As you can see in this table, by using the AdaBoosting weights instead of uniform weights, the classifier can achieve higher classification accuracy.

To sum up, we list the results by comparing the existed methods with our method in table-6. We can show that the proposed "Indoor/Ourdoor two-stage strategy with boosted weighting on different classifiers" method can significantly outperform other approaches.

	Accuracy
BOW (Naive)[9]	%70.33
BOW(NN)	%82.42
SVM-One Vector	%85.71
SVM-Geometric Mean [8]	%86.81
Our Method	**%90.11**

Table-6 Comparison between different proposed methods

VI. Conclusions

In this project, we propose the two-stage strategy and apply AdaBoosting to combine different classification methods, including SVM, BoW (Naïve Bayes) and BoW (Nearest Neighbor). We have proved that our method can improve the classification accuracy of location recognition system compared with other existed methods. However, due to the fact that we collected only 414 North Campus images in our data set, more general and comprehensive data set need to be collected before we can confidently give more conclusive result.

One possible extension for our system is that we can introduce multiple “topics” in the first stage. For now our first stage only labels the given image as either indoor or outdoor. However, if we can bring in more “topics” such as school, shop or park, our recognition system can then be applied in much broader situations.

During the project, we have learned how to utilize these machine learning algorithms and how to choose these algorithms according to different purposes. The most important thing is that most algorithms need to be modified before applying them on real applications. For example, the performance of Naïve Bayes classifier is really bad before we introduced tf-idf weighting. As a result, we should have well-understanding of these machine learning algorithms and have the ability to modify them according to different applications we face.

VII. Project Members’ Contributions and Efforts:

Bing Liao: SURF extraction, mean-shift clustering, BoW Naïve Based classifier implementation, tf-idf, data collecting, labeling and organizing data

Ko-Tung Lin: Histograms feature extraction, Multi-class SVMs, data collecting

Yu-Hui Chen: AdaBoosting algorithm implementation, cross-validation, data collecting, labeling and organizing data

Yi-Husan Tsai: BoW Nearest Neighbor classifier implementation, SIFT extraction, data collecting

References:

[1] M. Szummer and R.W. Picard, “Indoor-outdoor image classification”, In *CAIVD Workshop, ICCV*, Jan. 1998

[2] A. Vailaya, M. Figueiredo, A. Jain, and H.J. Zhang, “Image classification for content-based indexing”, *IEEE Trans. on Image Processing*, 10(1):117–130, Jan. 2001

[3] A. Oliva and A. Torralba, “Modeling the shape of the scene: a holistic representation of the spatial envelope”, *Int. Journal of Computer Vision.*, 42, 2001

[4] J. Vogel and B. Schiele, “A semantic typicality measure for natural scene categorization”, In *DAGM’04 Annual Pattern Recognition Symposium*, Tuebingen, Germany, 2004

[5] Li. Fei-Fei and P. Perona, “A Bayesian Hierarchical Model for Learning Natural Scene Categories”, *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2005

[6] Bosch, A. and Zisserman, A. and Munoz, X., “Scene Classification using a Hybrid Generative/Discriminative Approach”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008

[7] Yang, Jun and Jiang, Yu-Gang and Hauptmann, Alexander G. and Ngo, Chong-Wah, “Evaluating Bag-of-Visual-Words Representations in Scene Classification”, Proceedings of *ACM SIGMM Workshop on Multimedia Information Retrieval*, 2007

[8] Sjöberg, M.; Koskela, M.; Viitaniemi, V.; Laaksonen, J.; , “Indoor location recognition using fusion of SVM-based visual classifiers”, *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on* , vol., no., pp.343-348,Aug.29,2010-Sept.,1,2010

[9] Cummins, M. and Newman, P., “FAB-MAP: Appearance-Based Place Recognition and Mapping using a Learning Visual Vocabulary Madel”, *International Conference on Machine Learning (ICML),* Haifa, Israel, 2010

[10] Durrant-Whyte, H. and Bailey, T. “Simultaneous localization and mapping: Part I.”, *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006

[11] Hsu, C. and Lin, C., “A Comparison of Methods for Multiclass Support Vector Machines”, *IEEE Transactions on Neural Networks*, Vol. 13, No. 2, March 2002

[12] K. Fukunaga and L. D. Hostetler (January 1975). “The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition”, *IEEE Transactions on Information Theory* (IEEE) 21 (1): 32–40

[13] Lowe, D., “Distinctive image features from scale invariant keypoints”, IJCV 60, 91-110, 2004

[14] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp. 346--359, 2008

[15] Lowe, David G., “Object recognition from local scale-invariant features”, Proceedings of the *International Conference on Computer Vision* (ICCV), Vol.2, pp. 1150–1157, 1999

[16] Ryuji Funayama, Hiromichi Yanagihara, Luc Van Gool, Tinne Tuytelaars, Herbert Bay, "ROBUST INTEREST POINT DETECTOR AND DESCRIPTOR", published 2009-09-24

[17] Freund, Y. & Schapire, R., "A decision theoretic generalization of on-line learning and an application to boosting", *Journal of Computer and System Sciences* 55(1), 119 – 139. 1997

[18] Zhu, J., Rosset, S., Zou, H., & Hastie, T.. Multi-class AdaBoost.Distribution, 0-20.,2006

[19] Koen van de Sande: http://koen.me/research/colordescriptors/

[20] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM : a library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[21] J. L. Bentley, "Multidimensional binary search trees used for associative searching", *Communications of the ACM*, 18(9):509-517, 1975.