Self-supervised Online Metric Learning with Low Rank Constraint for Scene Categorization

Yang Cong, *Member, IEEE*, Ji Liu, Junsong Yuan, *Member, IEEE* and Jiebo Luo, *Fellow, IEEE*

***Abstract*—Conventional visual recognition systems usually train an image classifier in a bath mode with all training data provided in advance. However, in many practical applications, only a small amount of training samples are available in the beginning and many more would come sequentially during online recognition. Because the image data characteristics could change over time, it is important for the classifier to adapt to the new data incrementally. In this paper, we present an online metric learning method to address the online scene recognition problem via adaptive similarity measurement. Given a number of labeled data followed by a sequential input of unseen testing samples, the similarity metric is learned to maximize the margin of the distance among different classes of samples. By considering the low rank constraint, our online metric learning model not only can provide competitive performance compared with the state-of-the-art methods, but also guarantees to converge. A bi-linear graph is also defined to model the pair-wise similarity, and an unseen sample is labeled depending on the graph-based label propagation, while the model can also self-update using the more confident new samples. With the ability of online learning, our methodology can well handle the large-scale streaming video data with the ability of incremental self-updating. We evaluate our model to online scene categorization and experiments on various benchmark datasets and comparisons with state-of-the-art methods demonstrate the effectiveness and efficiency of our algorithm.**

***Index Terms*—low rank, online learning, metric learning, semi-supervised learning, scene categorization** [1]

I. INTRODUCTION

NOWADAYS, machine learning technology plays a central role in many practical systems with visual cognitive ability. Usually, the machine learning model is trained offline with labeled data, which is not updated during the online procedure, e.g. the computer vision system for scene categorization in our case. Unfortunately, for an online practical vision system, the performance of the machine learning model may deteriorate over time as the new incoming data may deviate from the initial training data. In order to handle such a issue, the model needs to be re-trained offline again in the batch mode using both existing and new data, which will be time-consuming.

Y. Cong is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, China, 110016 and the School of EEE, Nanyang Technological University, Singapore, 639798 e-mail: congyang81@gmail.com

J. Liu is with the Department of Computer Science, University of Wisconsin-Madison, USA e-mail: ji-liu@cs.wisc.edu

J. Yuan is with the School of EEE, Nanyang Technological University, Singapore, 639798 e-mail: jsyuan@ntu.edu.sg

J. Luo is with the Department of Computer Science, University of Rochester, Rochester, NY 14627 USA e-mail: jiebo.luo@gmail.com

[1]This work was supported in part by Natural Science Foundation of China (61105013), Nanyang Assistant Professorship (M4080134), and NTU CoE Seed Grant (M4081039).

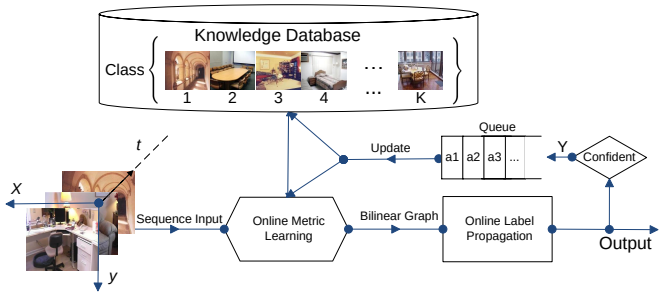


Fig. 1. Illustration of Online Metric Learning Procedure: We first collect labeled data and train an initial model. Then, with video data arriving sequentially, after extracting the features, online metric learning and label propagation are used to make a prediction. The confident samples are inserted into the training set queue to online update the model incrementally.

Moreover, if the size of the training dataset is too large, it is difficult for the batch training model to handle all the data in one iteration.

To overcome these problems, online models that learn from one or a group of instances each time [1]–[6] provide an efficient alternative to offline re-training by incrementally updating the classifier upon the new arrivals and establishing a decision boundary that adapts to the ever-changing data. In this paper, we focus on an adaptive similarity learner by representing the model in a matrix form, similar to metric learning, collaborative filtering, and multi-task learning. The intention of the online metric learning model is to learn a Positive Semi-definite (PSD) matrix $W \in \mathbb{R}^{d\times d}$, such that $p_1^T W p_2 \geq p_1^T W p_3$ for all p_1, p_2, $p_3 \in \mathbb{R}^d$; if p_1, p_2 are more similar and p_1, p_3 are less similar. For classification, p_1, p_2 should be from the same class and p_3 is from a different one. Essentially, the supervised online metric learner is designed to distinguish feature points with max margin as well. If all data with dimension d lie in a low dimension subspace r ($r < d$), the metric matrix with the rank less than r can distinguish any two samples if the data is distinguishable. Ideally, for data without any noise, many metric matrices with rank larger than r can distinguish it. However, training data always contains noise in practice, thus the metric matrix with a high rank would cause over-fitting and is sensitive to the noise and therefore not robust.

It is well known that the low rank property is often satisfied in practical data. We thus consider the low rank constraint in our metric learning model and learn a low dimensional representation of the data in a discriminative way, where low rank matrix models can therefore scale to handle substantially many more features and classes than with full-rank dense

matrices. For classification based on our online metric learning model, we define a bi-linear graph model to predict the label of a new incoming testing sample and fuse the information of both labeled and unlabeled data in the fashion of semi-supervised learning. Then a unified framework is designed to online self-update the models, which are used to handle online scene categorization, as shown in Fig. 1. The main contributions of our paper are as follows:

i. By considering the low rank property of the data distribution, we propose a novel online metric learning model with the low rank constraint to overcome over-fitting.
ii. We define a bi-linear graph to measure the similarity of pair-wise nodes. Different from traditional similarity graphs, such as full graph, $k-$NN and $\epsilon-$NN graphs, our bi-linear graph can maintain its accuracy for label propagation without tuning any parameters.
iii. We propose a general framework for online self-supervised learning by combining online metric learning with semi-supervised label propagation. In comparison to supervised learning with batch training, our algorithm can self-update the model incrementally and incorporate useful information from both labeled and unlabeled samples.

The organization of the rest of our paper is as follows. In Sec. II, we review the related work. In Sec. III, we present an overview of our online learning framework. Next, we propose our online metric learning model and online label propagation model in Sec. IV and Sec. V, respectively. Sec. VI reports our experimental results and comparisons with state-of-the-art methods. Finally, Sec. VII concludes the paper.

II. Related Work

Classifying scenes into categories, such as kitchen, office, coast and forests, is a challenging task due to the scale, illumination, content variations of the scenes and the ambiguities among similar types of scenes. For scene categorizations, there are mainly two key issues: image representation and similarity measurement.

For **image representation**, there are many scene descriptors, such as various Histogram-based features [7], [8], SIFT [9], Gist [10], Bag of Words (BOW) and spatial pyramid matching [11], kernel codebook [12], [13], Principal Component Analysis of Census Transform histograms (CENTRIST) [14], [15] and the combination of CENTRIST and color cues [16]. Designing an effective scene representation is beyond the scope of this paper and we chose to adopt CENTRIST here.

For **similarity measurement**, most traditional methods for scene recognition focus on supervised learning with batch training, such as [17], [13], [18], [19], [20], [21], [22], [23], [24], which cannot handle online processing, and would break down if the size of dataset is too large. Online algorithms have received much attention in the last decade, as they learn from one instance or sample at a time. For online supervised learning methods, Cauwenberghs et al. [4] propose a solution to the problem of training Support Vector Machines (SVMs) with a large amount of data; Utgoff et al. [25] introduce incremental decision tree classifiers that can be updated and retrained using new unseen data instances. Several methods have been proposed to extend the popular AdaBoost algorithm to the online scenario, for example complex background and appearance models [26], and visual tracking [1], [27], [28]. Moreover, there are also many practical industrial applications using online learning, e.g. [29] designs online image classifiers to handle CD imprint inspection in industrial surface inspection; [30] presents an online machine vision system for anomaly detection in sheet-metal forming processes; [31] models user preferences using online learning and also [3] combines supervised and semi-supervised online boosting trees. Learning a measurement of similarity between pairs of objects is a fundamental problem in machine learning. A large margin nearest neighbor method (LMNN) [32] is proposed to learn a Mahalanobis distance to have the k-nearest neighbors of a given sample belong to the same class while separating different-class samples by a large margin. LEGO [33], Online learning of a Mahalanobis distance using a Log-Det regularization per instance loss, is guaranteed to yield a positive semidefinite matrix. In [34], a metric learning by collapsing classes (MCML) is designed to learn a Mahalanobis distance such that same-class samples are mapped to the same point, formulated as a convex problem. Chechik et al. [2], [35], [36] design an Online Algorithm for Scalable Image Similarity learning (OASIS), for learning pairwise similarity that is fast and scales linearly with the number of objects and the number of non-zero features. However OASIS may suffer from over-fitting.

III. The flowchart of Our Algorithm

We propose an online learning framework, which uses metric learning to measure the similarity and adopts semi-supervised learning to label the testing samples. The flowchart is shown in Fig. 2, and consists of two phases: batch initial training phase and online prediction phase. During the batch initial training, each image is assigned a label and useful features are extracted and stored as feature vectors along with their labels. We then perform batch training to obtain an initial metric learner with the low rank constraint, i.e. the matrix W for similarity measurement. During the online training phase, features are also extracted from each sequentially incoming image, and depending on whether the data has a label or not, the proposed supervised and semi-supervised classifiers will be used to self-update the metric learner W. For an unlabeled sample, we measure the similarity between it and each of the initial training samples and propagate the label using our bi-linear graph accordingly. Next, those samples with high confidence scores are also used to update W. All the labeled samples are used for updating, where the updating procedure is similar to the batch initial training. Such a process iterates during online processing. The online learning phase will stop if the prediction performance reaches a desired level. Generally, there are mainly two key technical issues, online metric learning and label propagation, which are discussed below.

A. Notations

Let A be a symmetric matrix in $\mathbb{R}^{d\times d}$ and its eigenvalue decomposition is $A = U\Lambda U^T$, where $(\cdot)^T$ is the

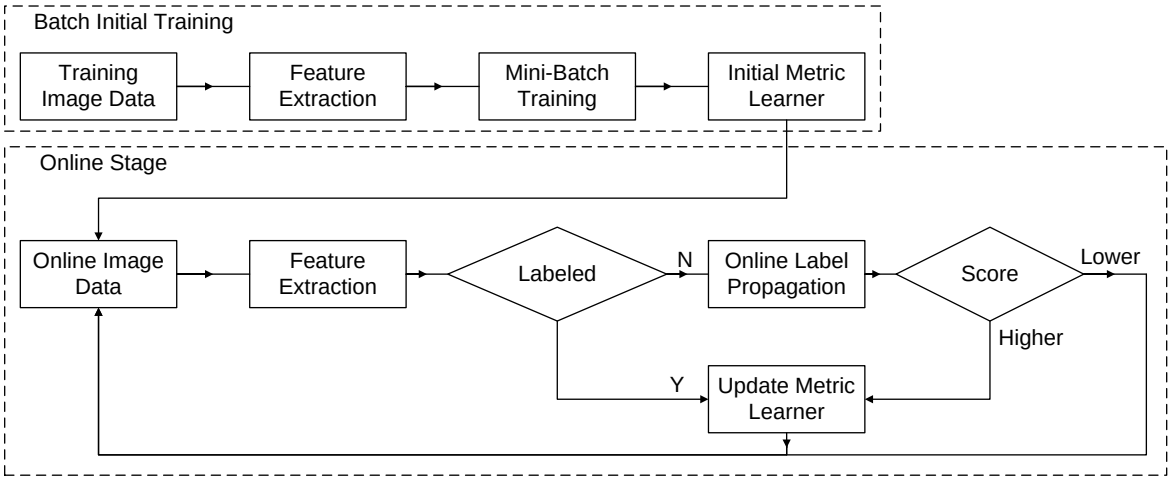


Fig. 2. The work flow of the proposed online learning algorithm.

transpose transformation, $U^TU = I$ and Λ is a diagonal matrix, i.e., $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_d)$. Denote $max(0, z)$ as $(z)^+$ or z^+. Let $\Lambda^+ = diag(\lambda_1^+, \lambda_2^+, ..., \lambda_d^+)$ and $\mathcal{D}_\tau(\Lambda) = diag((\lambda_1 - \tau)^+, (\lambda_2 - \tau)^+, ..., (\lambda_d - \tau)^+)$. Define $A^+ = U\Lambda^+U^T$and $A^- = A - A^+$. The shrinkage operation of A is $\mathcal{D}_\tau(A) = U\mathcal{D}_\tau(\Lambda)U^T$. Let $\mathcal{T}_\tau(\Lambda) = diag(min(\lambda_1, \tau), min(\lambda_2, \tau), ..., min(\lambda_d, \tau))$. Define the truncate operation $\mathcal{T}_\tau(A) = U\mathcal{T}_\tau(\Lambda)U^T$. We can easily verify that $A = \mathcal{T}_\tau(A) + \mathcal{D}_\tau(A)$.

IV. Online Metric Learning with Low rank constraint (OMLLR)

The goal of Online Metric Learning (OML) is to learn a similarity function $s_W(p_i, p_j)$ parameterized by matrix W for similarity measurement, which is a bi-linear form [2], [35] as:

$$s_W(p_i, p_j) \equiv p_i^T W p_j, \tag{1}$$

where $p_i, p_j \in \mathbb{R}^d$ are the feature vectors and $W \in \mathbb{R}^{d\times d}$. s_W assigns higher scores to more similar pairs of feature vectors and vice versa. For robustness, a soft margin is given as

$$s_W(p_i, \hat{p}_i) > s_W(p_i, \bar{p}_i) + 1, \ \forall p_i, \hat{p}_i, \bar{p}_i \in P. \tag{2}$$

Here $\hat{p}_i \in P$ is more similar to $p_i \in P$ than $\bar{p}_i \in P$. In our case, p_i and $\hat{p}_i$ belong to the same class; while p_i, $\bar{p}_i$ are from different classes. The hinge loss function $l_W(\cdot, \cdot, \cdot)$ is used to measure the cost:

$$l_W(p_i, \hat{p}_i, \bar{p}_i) = \max\big(0, 1 - s_W(p_i, \hat{p}_i) + s_W(p_i, \bar{p}_i)\big). \tag{3}$$

For the Online Algorithm for Scalable Image Similarity learning (OASIS) in [2], [35], the Passive-Aggressive algorithm is used to minimize the global loss l_W. First of all, W is initialized to an identity matrix $W^0 = I_{d\times d}$. Then, the algorithm iteratively draws a random triplet $(p_i, \hat{p}_i, \bar{p}_i)$, and solves the following convex problem with a soft margin:

$$\begin{aligned} W^i &= \arg\min_W \frac{1}{2}\|W - W^{i-1}\|_F^2 + \mu\xi. \\ &\text{s.t. } l_W(p_i, \hat{p}_i, \bar{p}_i) \le \xi \text{ and } \xi \ge 0 \end{aligned} \tag{4}$$

where $\|\cdot\|_F$ is the Frobenius norm (point wise L_2 norm) and μ is the tuning parameter. The classical online metric learning model OASIS can be optimized in an efficient way, however the learning process is not robust due to the overfitting problem. This is because the metric matrix W is not unique and could be with redundant freedom degree if the data points lie in a low dimensional subspace of $\mathbb{R}^d$.

To prove it, Theorem 1 shows that for the data in a subspace with dimension $r < d$, a metric matrix with rank at most r can determine the similarity measure.

Theorem 1: For any matrix $X \in \mathbb{R}^{n\times d}$ with rank r and any Positive Semi-definite (PSD) matrix $W \in \mathbb{R}^{d\times d}$, there exists a PSD matrix $Q \in \mathbb{R}^{d\times d}$ with $rank(Q) \le r$ such that

$$X^TWX = X^TQX.$$

In practice, each column of X is a data point $p_t \in \mathbb{R}^d$, and we have $X_i^TWX_j = X_i^TQX_j$. It means that for pair-wise similarity measurement of X_i and X_j, the metric matrix W is not unique.

If we construct the data matrix X from $\{p_t, \hat{p}_t, \bar{p}_t \mid \text{all } t\}$ (each column of X is a data point $p_t \in \mathbb{R}^d$) with a metric W, then we can always find a metric Q whose rank is at most r such that $X_i^TWX_j = X_i^TQX_j$.

Consider the training data with K classes $P_1, \cdots, P_K$ and let $P = \cup_{i=1}^K P_i$. Define the hinge loss function as $l(W, t) = max(0, 1 - p_t^TW\hat{p}_t + p_t^TW\bar{p}_t)$ like Eq. (3) where $\hat{p}_t, \bar{p}_t, p_t \in \mathbb{R}^d$, $W \in \mathbb{R}^{d\times d}$, and t is a random index, which is usually sampled uniformly from a index set T that includes K classes.

In order to estimate the metric matrix with a low rank property, a natural idea is to solve the following optimization problem:

$$\begin{aligned} \min_W :&\ \ f(W) := \mathbb{E}_t[l(W, t)] + \gamma rank(W) \\ s.t. :&\ \ W \succeq 0. \end{aligned} \tag{5}$$

Unfortunately, the optimization problem in Eq. (5) is non-convex and NP-hard. A conventional way is to use the trace norm $\|\ \|_*$ to approximate the rank function $rank(W)$, which makes the problem tractable:

$$\begin{aligned} \min_{W}: \ & f(W) := \mathbb{E}_t[l(W,t)] + \gamma\|W\|_* \\ s.t.: \ & W \succeq 0. \end{aligned} \tag{6}$$

If t follows the uniform distribution over the index set Φ, then $\mathbb{E}_t[l(W,t)] = \frac{1}{|\Phi|}\sum_{t\in\Phi} l(W,t)$. If one can evaluate the subdifferential of $\mathbb{E}[l(W,t)]$ at each step, then the proximal operation can be applied to solve the problem in Eq. (6):

$$\begin{aligned} W^{i+1} = arg\min_{W}: \ & \frac{1}{2}\|W - W^i + \alpha^i \partial \mathbb{E}_t[l(W^i,t)]\|^2 + \\ & \alpha^i\gamma\|W\|_* \\ s.t.: \ & W \succeq 0 \end{aligned}$$

Define the proximal operation as

$$prox_{P,\Omega}(x) = arg\min_{y\in\Omega} \ \frac{1}{2}\|y - x\|_F^2 + P(y). \tag{7}$$

In our case, $P(W) = \alpha^i\gamma\|W\|_*$ and $\Omega = \{W \mid W \succeq 0\}$. Then we have $W^{i+1} = prox_{P,\Omega}(W^i - \alpha^i\partial\mathbb{E}_t[l(W^i,t)])$.

The gradient of $\mathbb{E}_t[l(W^i,t)]$ is not computable sometimes, e.g., some data samples $\hat{p}_t, \bar{p}_t, p_t$ are unavailable in the i^{th} iteration, or it is too expensive to evaluate $\partial\mathbb{E}_t[l(W^i,t)]$ due to the large-scale training data. In order to handle this issue, the stochastic algorithm uses $\partial l(W^i,t)$ to approximate $\partial\mathbb{E}_t[l(W^i,t)]$ where t is randomly generated at each iteration, due to $\partial\mathbb{E}_t[l(W^i,t)] = \mathbb{E}_t[\partial l(W^i,t)]$. Thus, in the stochastic algorithm, the basic updating rule in each iteration is $W^{i+1} = prox_{P,\Omega}(W^i - \alpha^i\partial l(W^i,t))$. We can summarize the algorithm in Algorithm.1.

Algorithm 1 Online Metric Learning with Low Rank

Input: γ, α $\hat{p}_t$, $\bar{p}_t$, p_t for all t
Output: W

1: Initialize $i = 0$ and $W^0 = I \in \mathbb{R}^{d\times d}$
2: Repeat the following steps
3: Generate t from its distribution
4: $W^{i+1} = prox_{P(W),\Omega}\left(W^i - \alpha^i\partial l(W^i,t)\right)$
5: $i = i + 1$

Step 4 is the key step in this algorithm. First, one can verify that

$$\partial l(W,t) = \begin{cases} (\bar{p}_t - \hat{p}_t)p_t^T, & l(W,t) > 0; \\ \left[0, \ (\bar{p}_t - \hat{p}_t)p_t^T\right], & l(W,t) = 0; \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Note that $\partial l(W,t) = 0$ is a range when $l(W,t) = 0$. In this case, $\partial l(W,t)$ can take any value in this range. Theorem 2 introduces the closed form of $W^{i+1} = prox_{P,\Omega}(W^i - \alpha^i\partial l(W^i,t))$:

Theorem 2: Let $P(W) = \|W\|_*$ and $\Omega = \{W \mid W \succeq 0\}$. We have

$$prox_{\gamma P,\Omega}(C) = \mathcal{D}_\gamma\left(\frac{1}{2}(C + C^T)\right). \tag{9}$$

The remaining issue is how to choose the step size α^i. A conventional way is to let $\alpha^i = \Omega(1/\sqrt{i})$, which can lead to the optimal convergence rate as $\mathbb{E}[f(\overline{W}) - f(W^*)] \leq O\left(\frac{1}{\sqrt{|\Phi|}}\right)$, where W^* is the optimal solution and

$$\overline{W} = \frac{1}{|\Phi|}\sum_{i=1}^{\Phi} W_i \tag{10}$$

V. Online Label Propagation

Depending on the similarity measured by OMLLR above, we adopt the graph-based semi-supervised learning (also called label propagation) to make a more accurate prediction, which associates the information of both the labeled data and unlabeled data. For similarity graph, we define a new bi-linear graph using OMLLR:

Definition 1: Bi-linear Graph: Assume the similarity of pairwise points $\forall\ i,\ j,\ \ 1 \leq i, j \leq n,\ i \neq j$ is defined as

$$S_{i,j} \ = \ \max(0,\ S_w(i,j)) \ = \ \max(0,\ p_i^T W p_j). \tag{11}$$

For $p_i \in P,\ i \in [1,\ldots,N]$, we obtain a matrix $\{S_{ij},\ 1 \leq i,j \leq N\}$, where its symmetric version is $S_{i,j} = (S_{i,j} + S_{j,i})/2$.

In comparison with other traditional graph models, e.g. $k-$NN or $\epsilon-$NN graph, which are either sensitive to tuning parameters (e.g. σ) or instable to define a suitable graph structure without enough prior knowledge (e.g. k or ϵ), our bi-linear graph can maintain the accuracy without tuning parameters or prior knowledge of the topology graph.

A. Online Label Propagation

To predict the label of the new data, we define G = (V, E), where V denotes $n = n_l + n_u$ feature vectors (n_l labeled and n_u unlabeled); and E contains the edges of every pair of nodes measuring the pairwise similarity. Suppose we have $\Psi = \{1, 2, \ldots, \mathrm{K}\}$ classes. Let $F = \binom{F_l}{F_u} \in \mathbb{R}^{(n_l+n_u)\times K}$, where $F_l = [f_1, f_2, \ldots, f_{n_l}]^T \in \mathbb{R}^{n_l\times K}$ denotes the label matrix of the labeled data, and $F_u = [f_1, f_2, \ldots, f_{n_u}]^T \in \mathbb{R}^{n_u\times K}$ is the label matrix of unlabeled data needed to be predicted. In order to facilitate the calculation, we first normalize the similarity matrix S as,

$$P_{ij} = P(i \to j) = \frac{S_{ij}}{\sum_{k=1}^{n} S_{ik}}. \tag{12}$$

The matrix $P \in \mathbb{R}^{n\times n}$ can be split into labeled and unlabeled sub-matrices,

$$P = \begin{bmatrix} P_{ll} & P_{lu} \\ P_{ul} & P_{uu} \end{bmatrix}, \tag{13}$$

where $P_{ll} \in \mathbb{R}^{n_l\times n_l}$, $P_{lu} \in \mathbb{R}^{n_l\times n_u}$, $P_{ul} \in \mathbb{R}^{n_u\times n_l}$ and $P_{uu} \in \mathbb{R}^{n_u\times n_u}$. For label propagation, we have

$$F_u^{t+1} \leftarrow P_{uu}F_u^t + P_{ul}F_l. \tag{14}$$

When t approaches infinity, we have

$$F_u = \lim_{t\to\infty}(P_{uu})^t F_u^0 + (\sum_{i=1}^{t} P_{uu}^{(i-1)})P_{ul}F_l, \tag{15}$$

where F_u^0 is the initial value of F_u. Since P is a row normalized matrix, the sum of each row of the sub-matrix

Algorithm 2 Testing & Online Learning

Input: Query sample q, similar matrix W, training set $P = \{p_i\}$, threshold T_ξ
Output: $W^\star$, $P^\star$

Generate Bi-linear Graph S
$c_q^* = \arg\max_{c\in\Psi} E_c(q)$
if $E(c_q)/E(\overline{c}_q) > T_\xi$ **then**
    Insert $q \Rightarrow$ queue Q
end if
if Full (Q) **then**
    Update (Q)
    Insert $Q \Rightarrow P$ and clear Q
end if
return $W^\star = W$, $P^\star = P$

Function Update (Q)
Set $i = 1$
while $i <$ ITER-MAX $\bigcap$ $\|W^i - W^{i-1}\|_F < T_w$ **do**
    Get sample $q_i \in Q$, $q_i^+ \in c_{q_i}$ and $q_i^- \in \overline{c}_{q_i}$
    Update W by Algorithm.1
    $i = i + 1$
end while

$(P_{uu})^n$ approaches to zero. As a result, the first item of Eq. (15) converges to zero, $(P_{uu})^n F_u^0 \rightarrow 0$. Furthermore, the second item of Eq. (14) can be written as

$$F_u = (I - P_{uu})^{-1} P_{ul} F_l.$$

For online predicting the label of the sequentially input sample, we have $n_u = 1$, thus $P_{uu} \in \mathbb{R}^{1\times 1}$ is a fixed real number and $(I - P_{uu})^{-1}$ is a constant if P_{uu} is not equal to 1, so

$$F_u \propto P_{ul} F_l. \tag{16}$$

Eq. (16) is also consistent with the energy function we defined:

$$\mathrm{E}_c(x_i) = \sum_{j=1}^{n} \delta_c(j) S_{i,j},\ \delta_c(i) = \begin{cases} 1, & i \in c \\ 0, & \text{otherwise,} \end{cases} \tag{17}$$

where $c \in \{1, \ldots, \mathrm{K}\}$; x_i denotes the query sample; $S_{x,j}, j = \{1, \ldots, n\}$ is the bi-linear graph; and $\delta_c(i)$ is an indicate function. $E_c(x)$ is the energy function, which measures the cost of x belonging to class c. Thus, given x, the optimal solution $c^\star$ is the one maximizing the energy $E_c(x)$, as

$$c_x^\star = \arg\max_c E_c(x). \tag{18}$$

B. Updating

We design an adaptive model updating scheme depending on the property of each testing sample. As shown in Fig. 2 and Algorithm. 2, all the labeled testing samples are used to update the model; and for the unlabeled testing sample, it will be used to update the model if it satisfies the following criterion:

$$E_{c^\star}(q) > T_\xi \times E_{\overline{c}}(q), \quad \forall \overline{c},\ \overline{c} \neq c^\star. \tag{19}$$

In this paper $T_\xi = 1.2$. All samples used to update are pushed into a queue Q and when Q is full, the matrix W of the model will be iteratively updated using both the labeled data and unlabeled points with high confidence scores together. By tuning the length L of the Q, we can select to update the online model incrementally ($L = 1$) or with mini-batch training ($L > 1$).

VI. Experiments

In this section, we perform several experiments and comparisons to validate the proposed approach. Experiments are conducted on three types of dataset:

- Synthesized data, which is randomly generated for a fair comparison.
- Scene categorization dataset, including the 8-class sports image dataset, and the Visual Place Categorization (VPC) 09 video dataset, which is captured in the same fashion as a real online system. We extract the CENTRIST feature [15] from each image (or frame). The CENTRIST is in total 1302-d with the spatial-pyramid structure and we only use the first level of 42-d in this paper.
- Image classification dataset, i.e. Caltech 256. For image representation, we adopt the same feature used in [36] for a fair comparison, which is a spare representation based on the framework of local descriptors by combing the color histogram and texture histogram with the feature dimension as 1000.

A. Evaluation Criterion

We compare our method Online Metric Learning via Low Rank (OMLLR) with the state-of-the-art methods including both online learning methods and batch training methods. The accuracy is defined by Eq. (20):

$$Acc = \frac{\sharp\{\text{correct categorizations samples}\}}{\sharp\{\text{total number of samples}\}} \tag{20}$$

For batch training methods, we can achieve a definitive accuracy as defined in Eq. (20); and for online learning methods, as the model is updated incrementally, the accuracy will fluctuate with the iterations. Therefore, we adopt the model with the highest accuracy for comparison.

$$\begin{aligned} \text{OASIS:}\quad & W_{OASIS} = \arg\max_{W_j} Acc(j),\ j \in \{1, \ldots, N\} \\ \text{LMNN:}\quad & W_{LMNN} = \arg\max_{W_j} Acc(j),\ j \in \{1, \ldots, N\} \end{aligned} \tag{21}$$

where j is the index of iteration from 1 to N, and W_j is the matrix generated in each iteration.

For our OMLLR, which is designed to achieve the expectation of the model with the highest accuracy, we adopt two criteria for comparison, as in Eq. (22):

$$\begin{aligned} \text{Ours1:}\quad & \overline{W} = \frac{\sum_{i=1}^{N} \alpha_i W_i}{\sum_{i=1}^{N} \alpha_i} \\ \text{Ours2:}\quad & W_{max} = \arg\max_{W_i} Acc(i),\ i \in \{1, \ldots, N\} \end{aligned} \tag{22}$$

where i is the iteration index from 1 to N, $\alpha_i = 1\ \sqrt{i}$ and W_i is the matrix generated by each iteration. The weighted W, "Ours1", is the expectation of the model W, which guarantees to convergence in theory; and "Ours2" is the same as Eq. (21), i.e. the model W with the highest accuracy Acc.

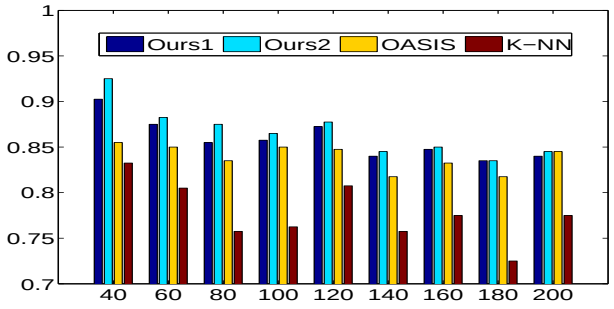


Fig. 3. Comparison of the accuracy between our methods and the state-of-the-art methods when varying the rank and fixing the feature dimension (dim=200), where the y-axis is the accuracy and the x-axis denotes the rank.

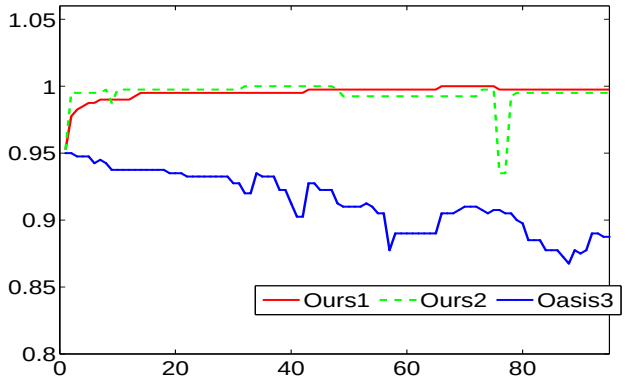


Fig. 4. An example of the simulation result, where the x-axis is the number of iterations (10k per step) and the y-axis is the accuracy.

B. Synthesized Data

We first use the synthesized data to evaluate the performance of our online metric learning model, OMLLR. The synthesized data have two classes, i.e. positive and negative. In order to generate low rank synthesized data, each class of data is first sampled from a low-dimensional multivariate normal distribution of full rank, where the mean and covariance matrices are randomly generated by a uniform distribution; then we embed them into a high dimensional feature space by random projection. The size of both training and testing samples is 1000, so we have totally 2000 samples. For a fair comparison, the Gaussian noisy is added into the synthesized data as well.

In Tab. I, we record results of comparing our method "Ours1" (the weighted W) and "Ours2" (the best result of W) with the state-of-the-art methods (OASIS [2], [35] and K-NN), where the "rank" varies from 5 to 100, and the corresponding feature dimension "dim" is twice of the rank. We can see that the accuracy of "Ours1" is lower than that of the "Ours2", but outperforms the classical online learning method (OASIS) and the benchmark batch training method, K-NN. In Fig. 3, we fix the feature dimension "dim" to 200, and vary the rank from 40 to 200 (with the interval of 20). The results are similar to those in Tab. I, i.e. the accuracy of the benchmark method K-NN is the worst one, and our methods both "Ours1" and "Ours2" outperform the classical OASIS. Another interesting point is that, when the feature dimension is fixed, the lower the data rank, the greater the gap of the accuracy between ours and OASIS, which justifies the effectiveness of the low rank constraint in our method. Therefore, we can conclude that our methodology can still work well for high dimensional real data having low rank property. Fig. 4 shows an example of online learning, our methodology not only outperforms OASIS, but also converges after only several iterations, e.g. "Ours1".

TABLE I
THE RESULTS OF COMPARISONS BY VARYING THE DIMENSION AND RANK OF FEATURES, WHERE OURS1 IS THE RESULT OF OUR METHOD USING WEIGHTED W; OURS2 IS THE RESULT OF W WITH THE HIGHEST ACCURACY; OASIS IS THE RESULT OF CLASSICAL ONLINE METRIC LEARNING [2], [35]. THE K-NN (K NEAREST NEIGHBOR METHOD) IS ADOPTED AS A BENCHMARK HERE.

Method	dim=10	20	40	60	100	200
	rank=5	10	20	30	50	100
Ours1	81.75	99.50	98.75	89.25	87.25	85.75
Ours2	**86.25**	**99.50**	**99.00**	**89.50**	**90.25**	**86.50**
Ours1 ($\gamma = 0$)	83.5	96.50	96.50	86.50	87.25	84.75
OASIS	82.25	97.00	96.00	85.50	85.50	85.00
K-NN	75.25	97.75	95.75	83.25	80.75	76.25

C. Sport 8 Dataset

The dataset in [17] contains images of eight sports, badminton, bocce, croquet, polo, rock climbing, rowing, sailing, and snowboarding as shown in Fig. 5. The number of images in each category ranges from 137 to 250. We randomly sample 50 images from each category for the initial training of W, and leave the remaining images for testing. The confusion matrix is shown in Fig. 6, where scores are from 64% to 91% with an average accuracy of 75.2%. In Tab. II, we compare our algorithm with the state-of-the-art methods, where our accuracy is higher than that of Li et al. [17], Cheng's [37] using an L1-graph based semi-supervised learning, and OASIS [2], [35], although a bit lower than Wu et al. [15]. However, Wu's [15] uses more training samples and higher feature dimension 1302 with an RBF kernel, while ours uses much fewer training samples and lower feature dimension with incremental updating. Fig. 7 shows an example of online learning. Both our methods for both "Ours1" and "Ours2" outperform OASIS in every iteration and converge as well.

TABLE II
THE ACCURACY OF THE SPORT 8 DATASET.

Method	Training Type	Accuracy (%)
Li [17]	Batch	73.4
Wu [15]	Batch	78.2
Cheng [37]	Semi-supervised+Batch	73.2
OASIS [2], [35]	Online	69.40
Ours1	Online	75.06
Ours2	Online	77.03

D. Visual Place Categorization (VPC) 09 Dataset

The Visual Place Categorization (VPC) 09 dataset [39] is captured using a rolling tripod plus a camera to mimic a robot, which is working in the same fashion as an online system. The VPC dataset was collected from 6 home environments, including 12 different scenarios (bathroom, bedroom, closet, dining-room, exercise-room, family-room, kitchen, living-room, media-room, workspace and transition).

TABLE III
THE COMPARISON OF THE AVERAGE ACCURACY OF OUR OMLLR AND THE STATE-OF-THE-ART METHODS USING VPC 09 DATASET.

Filter	Train	Methods	Home1	Home2	Home3	Home4	Home5	Home6	Bed	Bath	Kitchen	Living	Dining	Avg
No	Online	Ours1	42.36	21.53	37.53	40.43	32.22	38.28	44.27	57.83	17.75	41.60	15.50	35.39
		Ours2	54.50	31.12	42.89	54.99	41.95	51.13	44.09	66.76	26.22	50.63	42.77	**46.09**
		OASIS [2], [35]	25.33	21.32	21.99	20.57	24.84	39.18	25.92	6.02	3.47	82.28	10.00	25.54
		LMNN [38]	39.41	28.75	36.79	39.06	30.74	34.88	41.44	51.23	26.02	38.21	17.80	34.94
No	Batch	IROS [14]	44.77	33.33	40.68	43.28	41.10	48.07	48.13	65.71	46.56	29.18	19.78	41.87
		1-NN	41.83	27.48	33.96	38.66	30.85	29.70	40.69	46.38	26.92	40.92	13.81	33.75
		5-NN	41.18	28.23	34.33	39.82	31.62	31.56	39.21	46.32	28.78	44.94	13.04	34.46
Yes		Ours1	46.03	21.66	38.59	41.95	33.05	41.29	41.12	63.04	18.52	50.06	12.74	37.10
		Ours2	59.65	31.97	44.88	60.48	43.99	57.10	43.33	72.60	31.79	58.30	42.37	**49.68**
		IROS [14]	44.58	35.89	40.96	49.93	46.91	55.46	64.89	74.77	48.24	20.59	19.61	45.62

Fig. 5. Sample images from the Sport 8 datasets, including badminton, bocce, croquet, polo, rock climbing, rowing, sailing, and snowboarding.

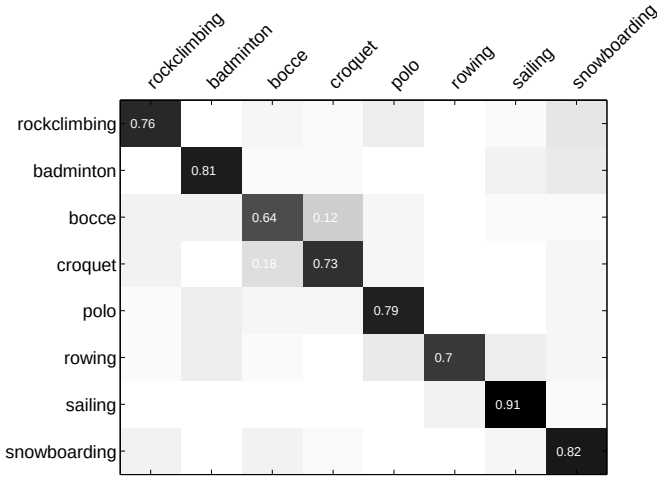


Fig. 6. Confusion matrix for the Sport 8 dataset, where the label of each row is the ground truth and the label of each column is the predicted category. The average accuracy is 77.03%, and random chance is 12.5%. For a better view, please check the electronic version.

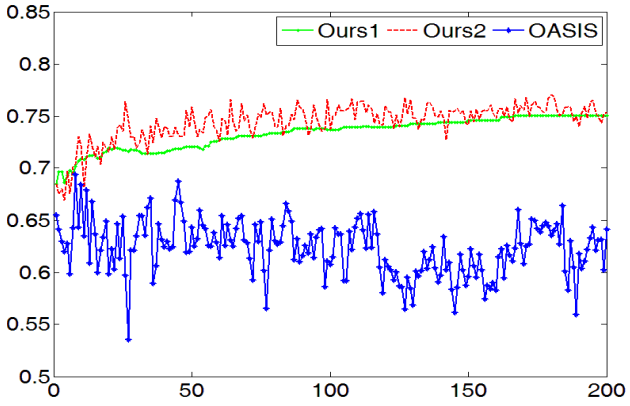


Fig. 7. The comparison of our OMLLR with OASIS using the Sport 8 dataset, where the x-axis is the number of iteration (10k per step) and the y-axis is the accuracy.

The VPC dataset was compressed in JPEG (95% quality) images with the resolution of each image 1280×720.

We compare our online method (OMLLR) with the state-of-the-art methods, including two online methods (OASIS [2], [35] and LMNN [38]) and batch training methods (K-Nearest Neighbor, 1-NN and 5-NN, and Wu's method [14]). The experiments are setup as recommended by [14], so we also adopt 5 categories for comparison in our paper, i.e. bedroom, bathroom, kitchen, living-room and dining-room. A leave one out cross validation strategy is adopted to evaluate our algorithm. The proposed method was repeated 6 times. In each run, one home was reserved for testing and all other 5 homes were combined to form a training set. The overall accuracy of our VPC system is the average of the 6 individual homes.

We first compare our method with OASIS [2], [35] as shown in Fig. 8, where all the online learning models are run for 3 million iterations, and each subfigure corresponds to home 1 to home 6. In 3 million iterations of Fig. 8, the accuracy of our online model fluctuates in each iteration and the accuracy of both "Ours1" and "Ours2" outperforms OASIS in all the cases. Although the accuracy of the expectation of the model "Ours1" is not better than the best one "Ours2", it is always better than those of the other iterations, especially for Home3. Moreover, "Ours1" guarantees to converge, and the property of convergence is critical for an online algorithm in practice.

The comparisons are shown in Tab. III, where for the average accuracy of 6 categories, both "Ours1" and "Ours2" outperform other online learning methods, e.g. OASIS [2], [35], LMNN [38] and also K-NN based batch training methods (1-NN and 5-NN); and for IROS [14] using the batch training model, the accuracy of "Ours2" is better than that of IROS. In general, the accuracy of online learning models is always worse than the that of batch training methods, but the performance of our OMLLR is acceptable.

For the issue of frame-level scene classification, the label of consecutive frames has high correlation. In [14], Wu et al. use a temporal smooth to improve the accuracy of the coarse result, and for us, we only adopt a simple median filter for frame-level temporal smooth with filter width as 5 frames.

TABLE IV
CATEGORIZATION ACCURACY (OURS1) OF ALL HOMES AND CATEGORIES WHEN THE BAYESIAN FILTERING IS NOT USED.

	bed	bath	kitchen	living	dining	average
home1	28.03	83.51	12.24	**95.12**	79.34	59.65
home2	28.60	**81.15**	9.92	27.44	12.72	31.97
home3	50.67	**89.89**	29.82	15.34	38.69	44.88
home4	23.21	56.60	79.37	**92.78**	50.46	60.48
home5	**81.79**	57.51	14.06	37.59	29.00	43.99
home6	47.71	66.96	45.32	**81.53**	44.00	57.10
average	43.33	**72.60**	31.79	58.30	42.37	49.68

TABLE V
CATEGORIZATION ACCURACY (OURS2) OF ALL HOMES AND CATEGORIES WHEN THE BAYESIAN FILTERING IS NOT USED.

	bed	bath	kitchen	living	dining	average
home1	30.97	75.08	13.49	**80.49**	72.46	54.50
home2	31.44	**71.51**	8.70	21.32	22.61	31.12
home3	48.36	**87.93**	25.79	15.02	37.33	42.89
home4	27.86	48.23	65.02	**85.39**	48.43	54.99
home5	**77.91**	55.36	12.40	33.58	30.49	41.95
home6	48.00	62.45	31.91	**67.97**	45.33	51.13
average	44.09	**66.76**	26.22	50.63	42.77	46.09

After the operation of temporal smooth filter, the accuracy of both online learning and batch training improves, and "Ours2" is still better than IROS. Tab. IV and Tab. V are the specific results by "Ours1" and "Ours2". As the testing and training samples are from different scenes [14], e.g. to test Home 1, the training samples include images from Home 2 to 6. Most of the results are lower than 50%, and both "Ours1" and "Ours2" outperform other methods. The frame-level results of scene categorization for VPC 09 are shown in Fig. 10. The images of the left column are examples of each home; and each figure of the right column is the frame-level result, where the x-axis is the frame index and the y-axis is the 5 class labels (bed, bath, kitchen, living and dining correspond to label 1, 2, 3, 5 and 6 with label 4 absent), and the red and blue line correspond to our predicted result and the ground truth, respectively. So the more overlapping of red and blue lines, the higher the accuracy of our model.

E. Caltech 256

We also test our OMLLR using the Caltech 256 dataset [40], which consists of 30607 images from 257 categories and is evaluated by humans in order to ensure image quality and relevance. Following [36], we also tested on subsets of classes from Caltech 256, i.e.

- 10 classes: bear, skyscraper, billiards, yo-yo, minotaur, roulette-wheel, hamburger, laptop-101, hummingbird, blimp.
- 20 classes: airplanes-101, mars, homer-simpson, hourglass, waterfall, helicopter-101, mountain-bike starfish-101, teapot, pyramid, refrigirator, cowboy-hat, giraffe, joy-stick, crab-101, birdbath, fighter-jet, tuning-fork, iguana, dog.
- 50 classes: car-side-101, tower-pisa, hibiscus, saturn, menorah-101, rainbow, cartman, chandelier-101, backpack, grapes, laptop-101, telephone-box, binoculars, helicopter-101, paper-shredder, eiffel-tower, top-hat, tomato, star-fish-101, hot-air-balloon, tweezer, picnic-table, elk, kangaroo-101, mattress, toaster, electric-guitar-101, bathtub, gorilla, jesus-christ, cormorant, mandolin, light-house, cake, tricycle, speed-boat, computer-mouse, superman, chimp, pram, friedegg, fighter-jet, unicorn, greyhound, grasshopper, goose, iguana, drinking-straw, snake, hotdog.

For each set, images from each class are split into a training set of 40 images and a test set of 25 images. A cross-validation procedure is also adopted to select the values of hyper parameters. For our OMLLR, the regularization parameter γ in Eq. (5) is in the set of $\gamma \in \{0.1, 0.01, 0.001, 0.001\}$.

For evaluation, a standard ranking precision measures based on nearest neighbors is also used. For each query image in the test set, all other training images are ranked according to their similarity to the query image. The number of same-class images among the top k images (the k nearest neighbors, e.g 1, 10, 50) is computed. When averaged across test images (either within or across classes), this yields a measure known as precision-at-top-k, providing a precision curve as a function of the rank k. We also calculate the mean average precision (mAP), a widely used criterion in the information retrieval community, where the precision-at-top-k is first calculated for each test image and averaged over all positions k that have a positive sample.

Our method, OMLLR is compared with the state-of-the-art online metric learning methods, including OASIS [2], [35], [36], LMNN [32], LEGO [33], MCML [34] and Euclidean (the standard Euclidean distance in feature space). The statistic result is proposed in Tab. VI, where our OMLLR is the result of the expectation of the model W, i.e. "Ours1", and OMLLR($\gamma = 0$ is used for justify the efficiency of low rank constraint, please check Sec. VI-F for details. Our OMLLR outperforms all state-of-the-arts for the full range of k. Another interesting thing is that our performance gain is decreased with the increase of the class number, i.e. from 10 classes to 50 classes. This is because for a fixed training steps (35k iterations), the more the number of classes, the lower the probability of different samples meet each other, which will destroy the performance. Fig. 9 demonstrates the precision curve for retrieval, and the performance of our method is better than others for all cases.

F. Comparisons

i Evaluating the effectiveness of low rank constraint:
To justify the effectiveness of low rank constraint, we can eliminate the impact of low rank constraint by setting the value of γ in Eq. (5) to 0, which is similar to the model definition of Eq. (4) as OASIS. The results are shown in Tab. VI using Caltech 256 dataset, the performance of our OMLLR is the best one; in comparison, ours with γ as 0 decreases accordingly and is similar to other models without low rank constraint, such as OASIS, MCML, LEGO and LMNN. This result again justifies the effectiveness of low rank constraint.

ii Comparing the influence of varying the initial training data size:

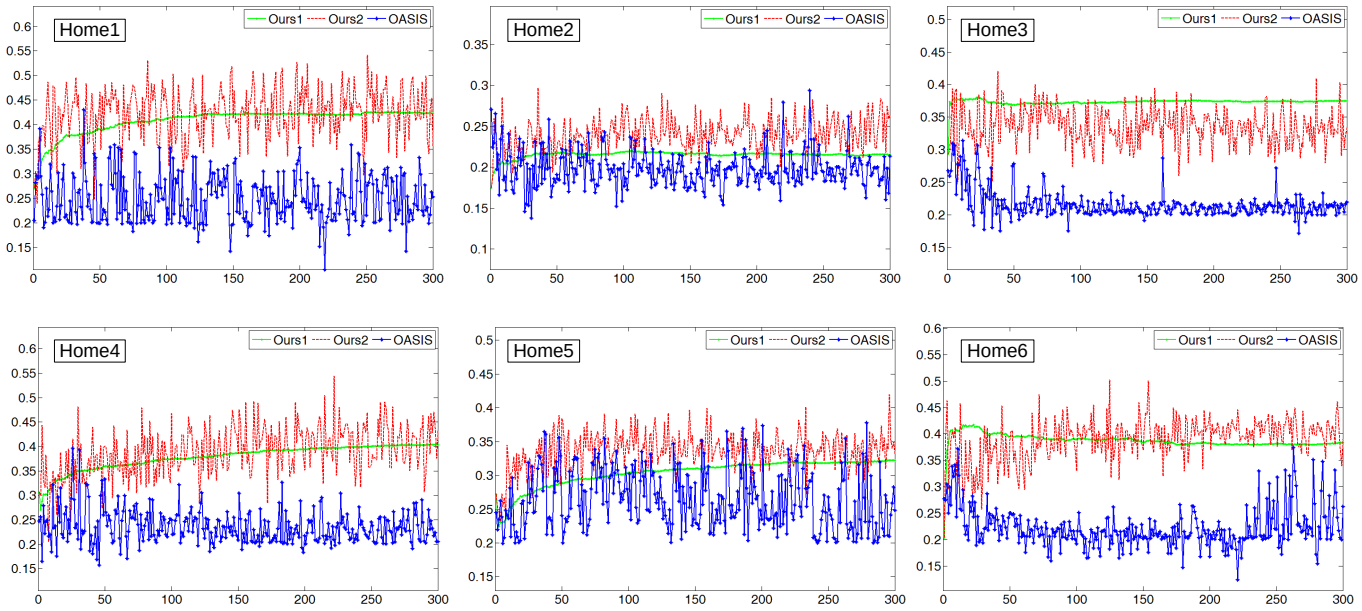
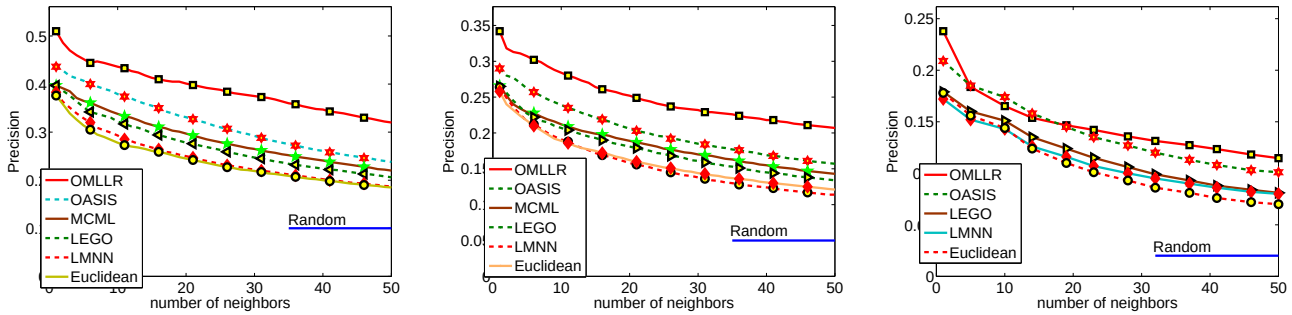


Fig. 8. The comparison of the accuracy between our OMLLR and OASIS [2], [35] for home1-6. In each figure, the x-axis corresponds to the iteration steps (10k for each) and the y-axis is the current accuracy, where the accuracy of "Ours1", "Ours2" and OASIS is denoted by sold green line, dash red line and dash blue line, respectively.



(a) 10 Classes (b) 20 Classes (c) 50 Classes

Fig. 9. Comparison of the performance of OMLLR, OASIS, LMNN, MCML, LEGO and the Euclidean metric in feature space. Each curve shows the precision at top k as a function of k neighbors. The results are averaged across 5 train/test partitions (40 training images, 25 test images), error bars are standard error of the means, black dashed line denotes chance performance. (A) 10 classes. (B) 20 classes. (C) 50 classes.

TABLE VI

AVERAGE PRECISION AND PRECISION AT TOP 1, 10, AND 50 OF ALL COMPARED METHODS. VALUES ARE AVERAGES OVER 5-FOLD CROSS-VALIDATIONS; $\pm$ VALUES ARE THE STANDARD DEVIATION ACROSS THE 5 FOLDS. A '$*$' DENOTES CASES WHERE A METHOD TAKES MORE THAN 5 DAYS TO CONVERGE. OMLLR($\gamma = 0$) MEANS IT DOES NOT CONSIDER THE LOW RANK CONSTRAINT.

10 classes	OMLLR Matlab	OMLLR($\gamma=0$) Matlab	OASIS Matlab	MCML Matlab+C	LEGO Matlab	LMNN Matlab+C	Euclidean -
Mean avg prec	**41** $\pm$ 1.6	34 $\pm$ 1.6	33 $\pm$ 1.6	29 $\pm$ 1.7	27 $\pm$ 0.8	24 $\pm$ 1.6	23$\pm$1.9
Top 1 prec.	**51** $\pm$ 2.8	44 $\pm$ 3.2	43 $\pm$ 4.0	39 $\pm$ 5.1	39 $\pm$ 4.8	38 $\pm$ 5.4	37 $\pm$ 4.1
Top 10 prec.	**45** $\pm$ 2.2	39 $\pm$ 2.6	38 $\pm$ 1.3	33 $\pm$ 1.8	32 $\pm$ 1.2	29 $\pm$ 2.1	27 $\pm$ 1.5
Top 50 prec.	**34** $\pm$ 1.0	26 $\pm$ 1.5	23 $\pm$ 1.5	22 $\pm$ 1.3	20 $\pm$ 0.5	18 $\pm$ 1.5	18 $\pm$ 0.7

20 classes	OMLLR	OMLLR($\gamma=0$)	OASIS	MCML	LEGO	LMNN	Euclidean
Mean avg prec	**23** $\pm$ 1.3	21 $\pm$ 1.3	21 $\pm$ 1.4	17 $\pm$ 1.2	16 $\pm$ 1.2	14 $\pm$ 0.6	14 $\pm$ 0.7
Top 1 prec.	**33** $\pm$ 1.7	29 $\pm$ 1.8	29 $\pm$ 2.6	26 $\pm$ 2.3	26 $\pm$ 2.7	26 $\pm$ 3.0	25 $\pm$ 2.6
Top 10 prec.	**26** $\pm$ 1.6	23 $\pm$ 1.7	24 $\pm$ 1.9	21 $\pm$ 1.5	20 $\pm$ 1.4	19 $\pm$ 1.0	18 $\pm$ 1.0
Top 50 prec.	**20** $\pm$ 1.0	17 $\pm$ 0.6	15 $\pm$ 0.4	14 $\pm$ 0.5	13 $\pm$ 0.6	11 $\pm$ 0.2	12 $\pm$ 0.2

50 classes	OMLLR	OMLLR($\gamma=0$)	OASIS	MCML	LEGO	LMNN	Euclidean
Mean avg prec	**14** $\pm$ 0.3	13 $\pm$ 0.4	12 $\pm$ 0.4	$*$	9 $\pm$ 0.4	8 $\pm$ 0.4	9 $\pm$ 0.4
Top 1 prec.	**22** $\pm$ 1.4	18 $\pm$ 1.5	21 $\pm$ 1.6	$*$	18 $\pm$ 0.7	18 $\pm$ 1.3	17 $\pm$ 0.9
Top 10 prec.	**17** $\pm$ 0.3	15 $\pm$ 0.4	16 $\pm$ 0.4	$*$	13 $\pm$ 0.6	12 $\pm$ 0.5	13 $\pm$ 0.4
Top 50 prec.	**12** $\pm$ 0.4	11 $\pm$ 0.3	10 $\pm$ 0.3	$*$	8 $\pm$ 0.3	7 $\pm$ 0.2	8 $\pm$ 0.3

TABLE VII
COMPARING THE INFLUENCE OF VARIOUS TRAINING DATA SIZE. THE FIRST ROW INDICATES THE TRAINING DATASIZE VARYING FROM 100 TO 1000.

	100	250	500	750	1000
Ours1	84.35	88.05	89.76	90.10	90.11
Ours2	86.35	88.44	89.76	90.11	90.12

TABLE IX
THE COMPARISON OF TIME CONSUMPTION, WHEN THE FEATURE DIMENSION INCREASES FROM 40 TO 200 IN THE TOP ROW. THE TIME CONSUMPTION FOR 1000 ITERATIONS IS RECORDED ACCORDINGLY.

Method	40	80	120	160	200
Ours	0.434	1.511	3.885	7.656	12.151
OASIS [2], [35]	0.092	0.096	0.107	0.114	0.130

We adopt the synthesized data to analyze the influence of the size of the initial training data, which varies from 100 to 1000, as shown in Tab. VII. We can find that by increasing the size of training data from 100 to 750, the accuracy of our model OMLLR for both Ours1 and Ours2 is improved significantly; and from the case of 750 to 1000, as the data size is large enough, the performance of our model is not changed. For other practical applications, a larger amount of training data is helpful to improve the performance of online learning model. However, it needs more iterations and consumes more computation time. Therefore, users should balance the size of training data and computational cost.

iii Comparing Bi-linear Graph with different similarity graphs:
To prove the effectiveness of our Bi-linear graph, we compare our Bi-linear graph with the classical graphs (such as $k-$NN, $\epsilon-$NN) in Tab. VIII using Sport 8 dataset. We can see that our proposed method not only outperforms other graphs, but also does not need to tune any parameters about the graph, where the traditional similarity graphs are parameter sensitive and their performances are not robust without a suitable selection of the parameters, e.g. $\sigma = 20$ or $\epsilon = 25$.

iv Comparing the time Consumption:
For comparing the time consumption of our OMLLR with the state-of-the-art methods, we test them using both the synthesized data and real data, i.e. Caltech 256, where our OMLLR is fully implemented in Matlab.
For the synthesized data, Tab. IX shows the comparison of time consumption between our OMLLR and the classical model, OASIS [2], [35]. With the feature dimension increases from 40 to 200, the time consumption is recorded every 1000 iterations. The comparison of time consumption for Caltech 256 dataset is shown in Tab. X, where our OMLLR is slower than OASIS, comparable with LEGO and LMNN, but much faster than MCML. Even though our OMLLR is more time consuming than OASIS, the performance of our OMLLR is better than other online metric learning methods, as shown in Tab. VI. This is because we adopt the SVD transformation for model optimization. All the experiments are performed on the computer with 4G RAM, Pentium IV 2.6GHz CPU.

VII. CONCLUSIONS

Most state-of-the-art scene recognition technologies relay on offline training in a batch model, thus may not be suitable for online scene recognition, which is still a challenging problem for computer vision. As the online image data characteristics may change over time, in this paper, we present an incremental metric learning framework for self-supervised online scene classification. Given a number of labeled data to initialize the similarity metric followed by a sequential input stream of unseen testing samples, the similarity metric is updated by maximizing the margin between different classes of samples with a low-rank constraint. The pair-wise similarity is measured by our new bi-linear graph for online label propagation to the new data. Next, by retaining the new images that are confidently labeled, the scene recognition model is further updated. Experiments on various benchmark datasets and comparisons with other state-of-the-art methods demonstrate the effectiveness and efficiency of our algorithm. Besides online scene recognition, our proposed online learning framework that can also be applied to other applications, such as object detection [41], object tracking [42], and image retrieval [2].

REFERENCES

[1] H. Grabner and H. Bischof, "On-line boosting and vision," in *CVPR*, vol. 1, 2006, pp. 260–267.
[2] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "An online algorithm for large scale image similarity learning," *NIPS*, vol. 21, pp. 306–314, 2009.
[3] F. Wang, C. Yuan, X. Xu, and P. van Beek, "Supervised and semi-supervised online boosting tree for industrial machine vision application," in *Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data*. ACM, 2011, pp. 43–51.
[4] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," *NIPS*, pp. 409–415, 2001.
[5] B. Liu, S. Mahadevan, and J. Liu, "Regularized off-policy td-learning," in *NIPS*, 2012.
[6] Y. Cong, J. Yuan, and Y. Tang, "Object tracking via online metric learning," in *ICIP*, 2012, pp. 417–420.
[7] K. van de Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.
[8] M. Szummer and R. Picard, "Indoor-outdoor image classification," in *IEEE International Workshop on Content-Based Access of Image and Video Database*, 1998, pp. 42–51.
[9] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
[10] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.
[11] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, vol. 2, 2006.
[12] J. van Gemert, J. Geusebroek, C. Veenman, and A. Smeulders, "Kernel codebooks for scene categorization," *ECCV*, pp. 696–709, 2008.
[13] A. Quattoni and A. Torralba, "Recognizing indoor scenes," *CVPR*, 2009.
[14] J. Wu, H. Christensen, and J. Rehg, "Visual place categorization: Problem, dataset, and algorithm," in *IROS*, 2009.
[15] J. Wu and J. Rehg, "CENTRIST: A visual descriptor for scene categorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1489–1501, 2011.
[16] Y. Cong, J. Yuan, and J. Luo, "Towards scalable summarization of consumer videos via sparse dictionary selection," *Multimedia, IEEE Transactions on*, vol. 14, no. 1, pp. 66–75, 2012.
[17] L. Li and L. Fei-Fei, "What, where and who? Classifying events by scene and object recognition," in *ICCV*, vol. 2, no. 4, 2007, p. 8.

TABLE VIII
COMPARING THE BI-LINEAR GRAPH WITH THE OTHER CLASSICAL SIMILARITY GRAPHS (FULL GRAPH, K-NN GRAPH AND $\epsilon-$NN GRAPH) UNDER VARIOUS PARAMETERS.

		K-NN Graph				$\epsilon-$NN Graph				
σ	Full Graph	K=10	K=30	K=50	K=100	$\epsilon = 25$	$\epsilon = 100$	$\epsilon = 900$	$\epsilon = 2500$	Bi-linear Graph
$\sigma = 5$	65.1	63.7	71.7	72.8	71.5	20.7	62.1	65.1	65.1	**77.03**
$\sigma = 10$	56.5	64.0	68.6	69.1	65.8	20.7	60.4	56.5	56.5	
$\sigma = 20$	51.2	62.9	65.7	65.2	60.4	20.7	60.4	51.2	51.2	

TABLE X
RUNTIME (MINUTES) OF ALL COMPARED METHODS (AROUND 35K TRAINING STEPS).

	OMLLR Matlab	OASIS Matlab	OASIS Matlab+C	MCML Matlab+C	LEGO Matlab	LMNN Matlab+C	fastLMNN Matlab+C
10 classes	342 ± 31	42 ± 15	0.12 ± 0.03	1835 ± 210	143 ± 44	337 ± 169	247 ± 209
20 classes	550 ± 43	45 ± 8	0.15 ± 0.02	7425 ± 106	533 ± 49	631 ± 40	365 ± 62
50 classes	731 ± 71	25 ± 2	1.6 ± 0.04	$*$	711 ± 28	960 ± 80	2109 ± 67

[18] D. Walther, E. Caddigan, L. Fei-Fei, and D. Beck, "Natural scene categories revealed in distributed patterns of activity in the human brain," *Journal of Neuroscience*, vol. 29, no. 34, p. 10573, 2009.

[19] L. Li, R. Socher, and L. Fei-Fei, "Towards total scene understanding: Classification, annotation and segmentation in an automatic framework," in *CVPR*, 2009, pp. 2036–2043.

[20] J. Liu and M. Shah, "Scene modeling using co-clustering," in *ICCV*, 2007.

[21] P. Quelhas, F. Monay, J. Odobez, D. Gatica-Perez, and T. Tuytelaars, "A thousand words in a scene," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 9, pp. 1575–1589, 2007.

[22] A. Bosch, A. Zisserman, M. Pujol *et al.*, "Scene classification using a hybrid generative/discriminative approach," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, 2008.

[23] J. Kivinen, E. Sudderth, and M. Jordan, "Learning multiscale representations of natural scenes using Dirichlet processes," in *ICCV*, 2007.

[24] J. Vogel and B. Schiele, "Semantic modeling of natural scenes for content-based image retrieval," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 133–157, 2007.

[25] P. Utgoff, N. Berkman, and J. Clouse, "Decision tree induction based on efficient tree restructuring," *Machine Learning*, vol. 29, no. 1, pp. 5–44, 1997.

[26] S. Avidan, "Ensemble tracking," in *CVPR*, vol. 2, 2005, pp. 494–501.

[27] X. Liu and T. Yu, "Gradient feature selection for online boosting," in *ICCV*, 2007, pp. 1–8.

[28] N. Oza and S. Russell, "Online bagging and boosting," in *Artificial Intelligence and Statistics*, 2001.

[29] E. Lughofer, "On-line evolving image classifiers and their application to surface inspection," *Image and Vision Computing*, vol. 28, no. 7, pp. 1065–1079, 2010.

[30] F. Gayubo, J. Gonzalez, E. de la Fuente, F. Miguel, and J. Peran, "On-line machine vision system for detect split defects in sheet-metal forming processes," in *ICPR*, vol. 1, 2006, pp. 723–726.

[31] O. Camoglu, T. Yu, L. Bertelli, D. Vu, V. Muralidharan, and S. Gokturk, "An efficient fashion-driven learning approach to model user preferences in on-line shopping scenarios," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2010, pp. 28–34.

[32] K. Weinberger and L. Saul, "Distance metric learning for large margin nearest neighbor classification," *The Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[33] P. Jain, B. Kulis, I. Dhillon, and K. Grauman, "Online metric learning and fast similarity search," *NIPS*, pp. 761–768, 2008.

[34] A. Globerson and S. Roweis., "Metric learning by collapsing classes," *NIPS*, 2006.

[35] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *The Journal of Machine Learning Research*, vol. 7, p. 585, 2006.

[36] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *The Journal of Machine Learning Research*, vol. 11, pp. 1109–1135, 2010.

[37] H. Cheng, Z. Liu, and J. Yang, "Sparsity Induced Similarity Measure for Label Propagation," in *ICCV*, 2009.

[38] K.Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," *NIPS*, 2006.

[39] "http://categorizingplaces.com/dataset.html."

[40] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[41] N. Jacobson, Y. Freund, and T. Nguyen, "An online learning approach to occlusion boundary detection," *Image Processing, IEEE Transactions on*, no. 99, pp. 1–1, 2010.

[42] Y. Wu, J. Cheng, J. Wang, H. Lu, J. Wang, H. Ling, E. Blasch, and L. Bai, "Real-time probabilistic covariance tracking with efficient model update," *Image Processing, IEEE Transactions on*, vol. 21, no. 5, pp. 2824–2837, 2012.

APPENDIX

Proof of Theorem 1

Proof: Since W is a PSD matrix, it can be decomposed as $W = UU^T$ where $U \in \mathbb{R}^{d\times d}$. Consider the following equation $X^TV = X^TU$ with respect to V. Define $B \in \mathbb{R}^{d\times(d-r)}$ with linear dependent columns $B_{.i}$'s in the null space of X^T. One can obtain the solution as $V = U + BZ$ where $Z \in \mathbb{R}^{(d-r)\times d}$. Split U and B into two parts $U = \begin{pmatrix} U1 \\ U2 \end{pmatrix}$ and $B = \begin{pmatrix} B1 \\ B2 \end{pmatrix}$ where $U1 \in \mathbb{R}^{(d-r)\times d}$, $U2 \in \mathbb{R}^{r\times d}$, $B1 \in \mathbb{R}^{(d-r)\times(d-r)}$, and $B2 \in \mathbb{R}^{r\times r}$. Define $Z = -B1^{-1}U1$. One verifies that $V = \begin{pmatrix} 0 \\ U2 - B2B1^{-1}U1 \end{pmatrix}$ and its rank is at most r. Since $X^TU = X^TV$, we obtain $X^TWX = X^TQX$ and the rank of Q is r by letting $Q = VV^T$. ■

Proof of Theorem 2

Proof: Decompose C into the symmetric space and the skew symmetric space, i.e., $C = C_y + C_k$ where $C_y = \frac{1}{2}(C + C^T)$ and $C_k = \frac{1}{2}(C - C^T)$. Note that $\langle C_y, C_k\rangle = 0$. Consider $W \succeq 0$ (W must be symmetric) in the following

$$\begin{aligned}\|W - C\|_F^2 =& \|W - C_y - C_k\|_F^2 \\ =& \|W - C_y\|_F^2 + \|C_k\|_F^2 + 2\langle W - C_y, C_k\rangle \\ =& \|W - C_y\|_F^2 + \|C_k\|_F^2.\end{aligned} \tag{23}$$

Thus, we obtain $prox_{\gamma P,\Omega}(C) = prox_{\gamma P,\Omega}(C_y)$.

$$\begin{aligned}
&\min_{W\succeq 0} \ \frac{1}{2}\|W - C_y\|_F^2 + \gamma\|W\|_* \\
=&\min_{W\succeq 0} \ \frac{1}{2}\|W - C_y\|_F^2 + \max_{\|Z\|\leq\gamma, Z\in S\mathbb{R}^{d\times d}} \langle W, Z\rangle \\
=&\max_{\|Z\|\leq\gamma, Z\in S\mathbb{R}^{d\times d}} \min_{W\succeq 0} \ \frac{1}{2}\|W - C_y\|_F^2 + \langle W, Z\rangle \\
=&\max_{\|Z\|\leq\gamma, Z\in S\mathbb{R}^{d\times d}} \min_{W\succeq 0} \ \frac{1}{2}\|W - C_y + Z\|_F^2 + \langle C_y, Z\rangle - \frac{1}{2}\|Z\|_F^2 \\
=&\max_{\|Z\|\leq\gamma, Z\in S\mathbb{R}^{d\times d}} \ \frac{1}{2}\|(C_y - Z)^-\|_F^2 + \langle C_y, Z\rangle - \frac{1}{2}\|Z\|_F^2
\end{aligned} \tag{24}$$

The first equality uses the dual form of the trace norm of a PSD matrix, where $S\mathbb{R}$ denotes the symmetric space. The second equality is due to Von Neumann theorem. The last equality uses the result that the projection from a symmetric matrix X onto the SDP cone is X^+, which also implies that $W = (C_y - Z)^+$.

It follows that

$$\begin{aligned}
&\max_{\|Z\|\leq\gamma, Z\in S\mathbb{R}^{d\times d}} \ \frac{1}{2}\|(C_y + Z)^-\|_F^2 + \langle C_y, Z\rangle - \frac{1}{2}\|Z\|_F^2 \\
=&\max_{\|Z\|\leq\gamma, Z\in S\mathbb{R}^{d\times d}} \ \frac{1}{2}\|(C_y - Z)^-\|_F^2 - \frac{1}{2}\|C_y - Z\|_F^2 + \frac{1}{2}\|C_y\|_F^2 \\
=&\max_{\|Z\|\leq\gamma, Z\in S\mathbb{R}^{d\times d}} \ -\frac{1}{2}\|(C_y - Z)^+\|_F^2 + \frac{1}{2}\|C_y\|_F^2
\end{aligned} \tag{25}$$

From the last formulation, we obtain the optimal $Z^* = \mathcal{T}_\gamma(C_y)$ and the optimal $W^* = (C_y - Z^*)^+ = (C_y - \mathcal{T}_\gamma(C_y))^+ = \mathcal{D}_\gamma(C_y)^+ = D_\gamma(C_y)$. It completes our proof. ■

Yang Cong (S'09-M'11) received the B.Sc. degree from Northeast University in 2004, and the Ph.D. degree from State Key Laboratory of Robotics, Chinese Academy of Sciences in 2009. He is a Research Fellow of National University of Singapore (NUS) and Nanyang Technological University (NTU) from 2009 to 2011, respectively. Now, he is an Associate Researcher of Chinese Academy of Science. His current research interests include compute vision, pattern recognition, multimedia and robot navigation. He is a member of IEEE.

Ji Liu is currently a graduate student of the Department of Computer Sciences at University Wisconsin-Madison. He received his bachelor degree in automation from University of Science and Technology of China in 2005 and master degree in Computer Science from Arizona State University in 2010. His research interests include optimization, machine learning, computer vision, and graphics. He won the KDD best research paper award honorable mention in 2010.

Junsong Yuan (M'08) a Nanyang assistant professor at Nanyang Technological University (NTU), Singapore. He received the Ph.D. and M.Eng degrees from Northwestern University and National University of Singapore, respectively. Before that, he graduated from the special program for the gifted young in Huazhong University of Science and Technology, Wuhan, P.R.China. He is currently the Program Director of Video Analytics at Infocomm Center of Excellence, School of EEE, NTU. His research interests include computer vision, video analytics, large-scale visual search and mining, human computer interaction, biomedical image analysis, etc. He received the Outstanding EECS Ph.D. Thesis award from Northwestern University, and the Best Doctoral Spotlight Award from IEEE Conf. Computer Vision and Pattern Recognition Conference (CVPR'09). He is the co-chair of IEEE CVPR 2012 and 2013 Workshop on Human action understanding from 3D data (HAU3D'12'13), and the co-chair of CVPR 2012 Workshop on Large-scale video search and mining (LSVSM'12).

Jiebo Luo (S93-M96-SM99-F09) received a BS degree in electrical engineering from the University of Science and Technology of China in 1989 and his PhD degree from the University of Rochester in 1995. He was a Senior Principal Scientist with the Kodak Research Laboratories in Rochester before joining the Computer Science Department at the University of Rochester in 2011. His research interests include signal and image processing, machine learning, computer vision, social media data mining, medical imaging, and computational photography. He has authored over 200 technical papers and holds over 70 U.S. patents. Dr.Luo has been actively involved in numerous technical conferences, including serving as the general chair of ACM CIVR 2008, program co-chair of IEEE CVPR 2012, ACM Multimedia 2010 and SPIE VCIP 2007. Currently, he serves on multiple IEEE SPS Technical Committees (IMDSP, MMSP, and MLSP). He has served on the editorial boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, Pattern Recognition, Machine Vision and Applications, and the Journal of Electronic Imaging. He is also a Fellow of the SPIE and IAPR.

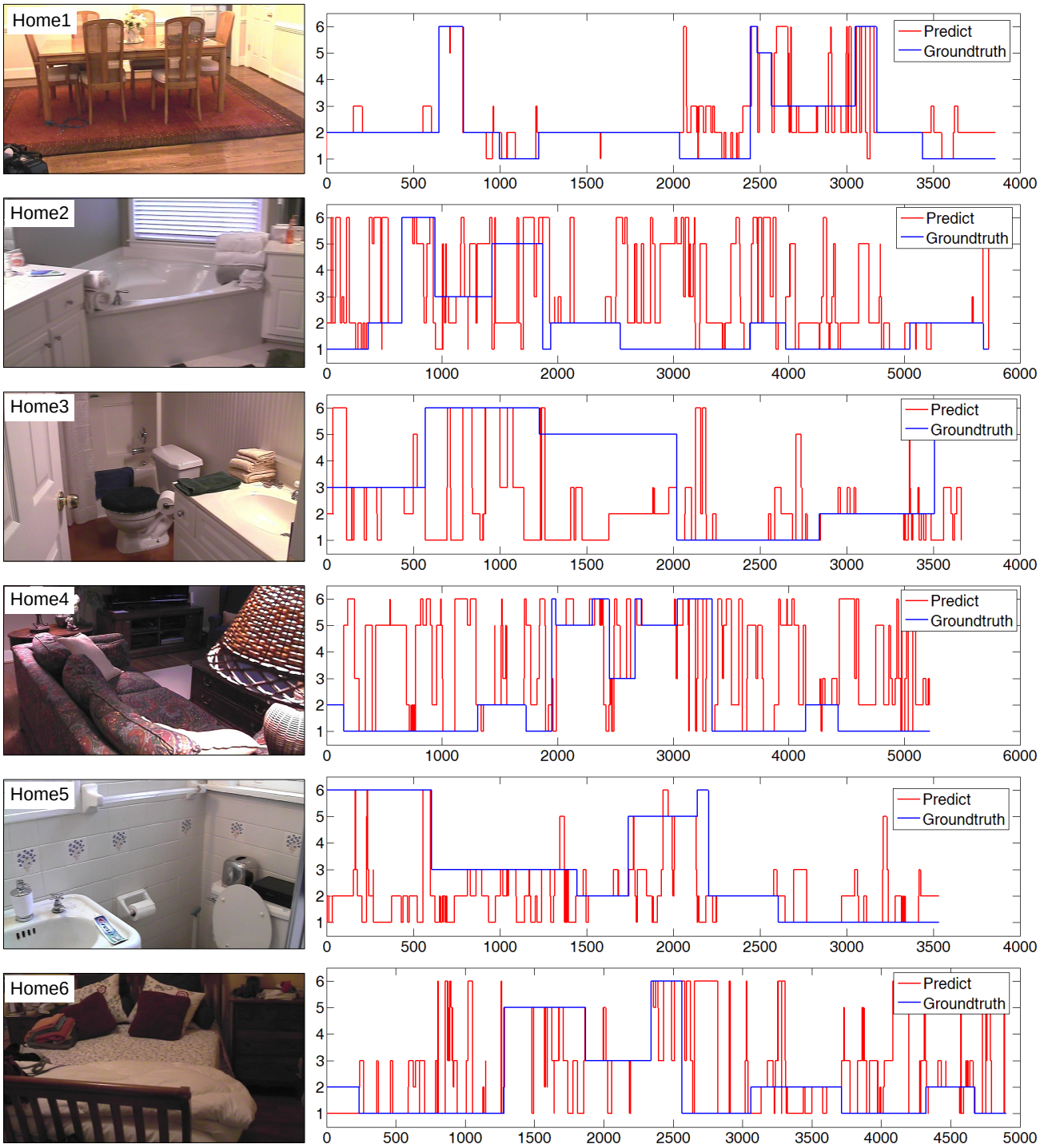


Fig. 10. The results of scene categorization for VPC 09. The images of the left column are examples of each home. Each figure of the right column is the frame-level result, where the red and blue line correspond to the predicted result of our methodology after smooth filter and the ground truth, respectively, and the x-axis is the frame index and the y-axis is the 5 class labels (bed, bath, kitchen, living and dining correspond to label 1, 2, 3, 5 and 6 respectively with label 4 absent).