Real-Time Vehicle Make and Model Recognition Based on a Bag of SURF Features

Abdul Jabbar Siddiqui, Abdelhamid Mammeri, and Azzedine Boukerche, *Fellow, IEEE*

***Abstract*—In this paper, we propose and evaluate unexplored approaches for real-time automated vehicle make and model recognition (VMMR) based on a bag of speeded-up robust features (BoSURF) and demonstrate the suitability of these approaches for vehicle identification systems. The proposed approaches use SURF features of vehicles' front- or rear-facing images and retain the dominant characteristic features (*codewords*) in a *dictionary*. Two schemes of dictionary building are evaluated: "*single dictionary*" and "*modular dictionary.*" Based on the optimized dictionaries, the SURF features of vehicles' front- or rear-face images are embedded into BoSURF histograms, which are used to train multiclass support vector machines (SVMs) for classification. Two real-time VMMR classification schemes are proposed and evaluated: a single multiclass SVM and an ensemble of multiclass SVM based on attribute bagging. The processing speed and accuracy of the VMMR system are affected greatly by the size of the dictionary. The tradeoff between speed and accuracy is studied to determine optimal dictionary sizes for the VMMR problem. The effectiveness of our approaches is demonstrated through cross-validation tests on a recent publicly accessible VMMR data set. The experimental results prove the superiority of our work over the state of the art, in terms of both processing speed and accuracy, making it highly applicable to real-time VMMR systems.**

***Index Terms*—Intelligent surveillance, vehicle classification, intelligent transportation.**

I. Introduction

AUTOMATED Vehicle Make and Model Recognition (VMMR) systems is an area of great interest in numerous Intelligent Transportation Systems (ITS) applications. The most significant applications of VMMR for ITS includes automated vehicular surveillance in specific areas that are highly vulnerable to security threats, such as parking lots of public spaces (e.g., malls, stadiums or airports). Another important application of VMMR is related to situations in which the police are searching for a specific vehicle type, make, or model. Other applications include traffic studies and analyses. In this work, we propose Bag of Speeded Up Robust Features (BoSURF)-based approaches for automated VMMR, building on the Bag-of-Features framework [1] for representing vehicles' front or rear views as captured by commonly used 2D vision cameras. Traditional vehicle identification systems recognize makes and models of vehicles relying on manual human observations or license plate recognition systems, barely meeting real-time constraints, for instance [2]. Both approaches are failure-prone and have several limitations. First, it is difficult in a practical sense for human observers to remember and efficiently distinguish between the wide variety of vehicle makes and models. Second, it becomes a laborious and time-consuming task for a human observer to monitor and observe a multitude of screens and record the incoming or outgoing makes and models, or to even spot the make and model being searched for [3] and [4]. On the other hand, the VMMR systems that rely on license plates suffer from the following disadvantages. License plates can easily be forged, damaged, modified, or occluded, as depicted in Fig. 1. This can prevent VMMR systems from detection and recognition of vehicles and their make and model. Also, there are some license plates that can be ambiguous (e.g., between "0" and "O"), as shown in the left of Fig. 1. Moreover, in some areas, it may not be necessary to display the license plate at the front or rear. If the license plates recognition system is not equipped to check for license plates at both (front and rear) views of the vehicle, it could fail. So, when license plate recognition systems fail to accurately read the detected license plates due to the above issues, the wrong make and model information could be retrieved from the license plate registry or database.

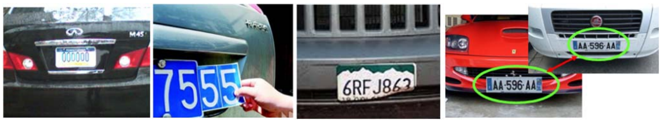


Fig. 1. From left to right: Some cases in which VMMR systems based on license plates can fail due to ambiguity, forgery, damage, or license-plate duplication.

To overcome the above shortcomings in traditional vehicle identification systems, automated VMMR techniques have recently gained attention, but without considering processing speed as the primary factor [3] and [4]. The make and model of the vehicle recognized by the VMMR system can be cross-checked with the license plate registry to screen for fraud. In this way, automated VMMR systems augment traditional license plate recognition-based vehicle identification systems to further enhance security.

We tackle the problem of real-time vehicle make and model recognition as a challenging multi-class classification problem. In this work, a "class" refers to a vehicle make and model (Eg: Toyota Altis, Toyota Camry, Nissan Xtrail are three different classes). There are two broad categories of challenges in VMMR: (1) "*Multiplicity,*" and (2) "*Ambiguity*" [5]. The *multiplicity* problem occurs when a vehicle model (of the same

Manuscript received July 26, 2015; revised December 31, 2015; accepted March 11, 2016. Date of publication April 25, 2016; date of current version October 28, 2016. The Associate Editor for this paper was Z. Duric.

The authors are University of Ottawa, Ottawa, ON K1N6N5, Canada (e-mail: asidd074@uottawa.ca; amammeri@uottawa.ca; boukerch@site.uottawa.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2016.2545640

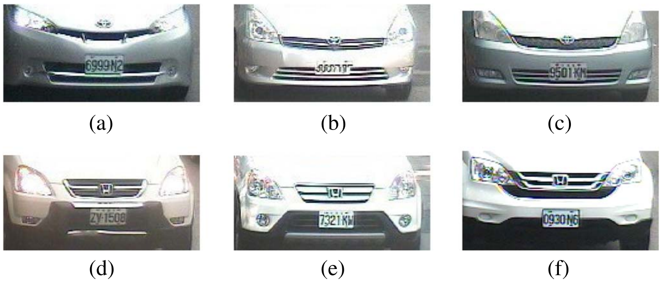


Fig. 2. Multiplicity problems with (a)–(c) Toyota Wish and (d)–(f) Honda CRV in the NTOU-MMR data set [6]. The multiplicity problem means one vehicle MM often displays different shapes on the road. (a) T Wish 2010. (b) T Wish 2009. (c) T Wish 2005. (d) H CRV 2003. (e) H CRV 2005. (f) H CRV 2009.

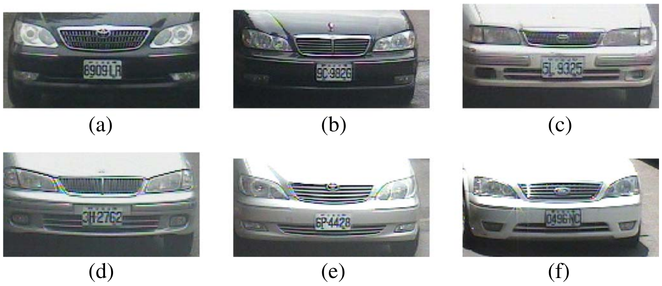


Fig. 3. Intermake ambiguity problems between (a) and (b), (c) and (d), and (e) and (f) in the NTOU-MMR data set of [6]. "T," "N," and "F" stand for "Toyota," "Nissan," and "Ford," respectively. The ambiguity problem refers to the case where vehicles manufactured by different companies have comparable shapes. (a) T Camry 2005. (b) N Cefiro 1999. (c) T Tercel 2005. (d) N Sentra 2003. (e) T Camry 2006. (f) F Mondeo 2005.

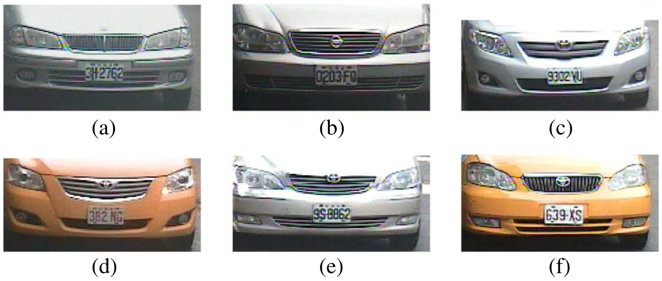


Fig. 4. Intramake ambiguity problems between (a) and (b), (c) and (d), (c) and (e), and (e) and (f) in the NTOU-MMR data set of [6]. "N" and "T" stand for "Nissan" and "Toyota," respectively. Intramake ambiguity results when different vehicles (models) from the same company (make) have a comparable shape or appearance. (a) N Sentra 2003. (b) N Cefiro 1997. (c) T Altis 2008. (d) T Camry 2008. (e) T Camry 2006. (f) T Altis 2006.

make) has different shapes. Fig. 2 shows some examples of the multiplicity problem in NTOU-MMR Dataset [6]. We further classify the *ambiguity* problem into two categories: (a) "*Inter-Make Ambiguity*," and (b) "*Intra-Make Ambiguity*." The former ambiguity refers to the issue of vehicles (models) of different companies (makes) having a visually comparable shape or appearance, which might lead to confuse between vehicles, i.e., two different make-model classes have comparable front/rear views (See Fig. 3). The latter type of ambiguity results when different vehicles (models) from the same company (make) have a comparable shape or appearance. For example, the "Altis" and "Camry" models of the "Toyota" make have comparable front faces (See Fig. 4).

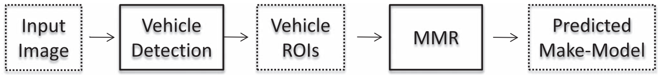


Fig. 5. General architecture of VMMR systems.

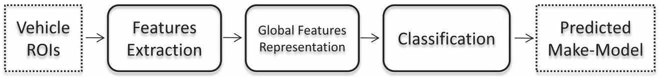


Fig. 6. Flowchart of the MMR module of VMMR systems.

To address the above-mentioned challenges in VMMR, the major contributions of this work on vehicle make and model recognition are summarized as follows with an objective of achieving a processing speed of at least 7 fps to meet real-time requirements while having an accuracy of around 95%: (1) Unexplored approaches for VMMR are proposed and evaluated based on the BoSURF framework, in which the dominant features of all makes and models are learned and represented in an optimized dictionary; (2) Two schemes for Dictionary Building are studied and evaluated to address the multiplicity and ambiguity problems of VMMR: (a) the "*Single-Dictionary*," and (b) the "*Modular-Dictionary*"; (3) The optimal dictionary sizes for VMMR are recommended by studying their effect on processing speed and accuracy, as shown in Section VIII-A; (4) Two real-time classification schemes are proposed and evaluated: (a) *Single Multi-Class SVM Classifier (SVM)* and (b) *Attribute Bagging based Ensemble of SVM Classifiers (AB-SVM)*, in order to simultaneously learn the inter-class differences (to solve inter-make and intra-make ambiguity issues) and the intra-class similarities (to solve the multiplicity issues); (5) The effectiveness and superiority of our BoSURF approaches for VMMR are validated on random training-testing dataset splits of NTOU-MMR Dataset [5].

The remainder of the paper is organized as follows. We give an overview of the related state-of-the-art VMMR works in Section II, and discuss their limitations. The target environment and dataset used to demonstrate the effectiveness of our approaches are described in Section III. The description of the proposed BoSURF approaches is provided in Section IV. Section V explains the two dictionary building schemes proposed in this work for VMMR. Then, Section VI presents the method by which BoSURF features are formed using the learned dictionaries. The two real-time classification schemes we propose for VMMR are presented in Section VII. After describing the experimental setup and the performance metrics in Section VIII, we present the results and discussions in Section IX. Finally, Section X provides the conclusions and future work.

II. Related Work

The problem of automated vehicle make and model recognition is an important task for vehicular surveillance and other ITS applications. We provide the general architecture of VMMR systems in Fig. 5. The two main modules are: (A) Vehicle "Detection," and (B) "Make and Model Recognition" (MMR). The MMR module is comprised of three steps: (1) Features Extraction, (2) Global Features Representation, and (3) Classification, as shown in Fig. 6.

TABLE I
SUMMARY OF FEATURES EXTRACTION, GLOBAL FEATURES REPRESENTATION, AND CLASSIFICATION APPROACHES IN VMMR WORKS

Work	Local Features	Global Representations	Classification
Dlagnekov [4] (2005)	SIFT [9]	-	Brute-Force Matching
Munroe and Madden [3] (2005)	Edges	Concatenation of edge image pixels	Decision Tree, k-Nearest Neighbors, Feed Forward Neural Networks
Pearce and Pears [2] (2011)	Harris Corners	LNHS and SMG	Nearest Neighbors; Naive Bayes
Baran et al. [10] (2013)	SIFT [9]; SURF; Edge Histogram	Dictionary-based Sparse Vector of Occurrence Counts	Multi-Class SVM
Hsieh et al. [5] (2014)	SURF[13]	Grid-based Representation	Ensemble of Multi-Class SVM
Fraz et al. [12] (2014)	SIFT [9]	Fisher Encoding-based MLR	Euclidean Distance-based Matching
Llorca et al. [21] (2014)	Emblem Positions & Sizes, HOG of Emblem Regions	-	Linear SVM-based Bayesian Inference Voting
Chen et al. [15] (2015)	HOG	Grid-based concatenation of features	Sparse Representation and Hamming Distance-based
He at el. [16] (2015)	Edges- and Gradients-based	Normalised fixed-length numerical vectors, after normalisations and textural filtering	Ensemble of Neural Networks; SVM; kNN; AdaBoost

In this section, we provide an overview of works completed on the different modules of VMMR systems (See Table I). Since our work is focused on developing and evaluating improved approaches for the Global Features Representation and Classification steps of the MMR module, we provide a comprehensive discussion of related works in the context of these steps.

A. Vehicle Detection

The problem of detecting vehicles in image sequences from surveillance cameras has been well investigated by many researchers. The objective of vehicle detection approaches is to find a vehicle Region of Interest (ROI) over the given image, such that it outlines the vehicle (or vehicle's front/rear face) by filtering out the background regions. The VMMR modules can then work on these ROIs instead of the whole image, which could otherwise decrease the VMMR accuracy. As the objective of this work is to achieve a VMMR module that can augment traditional license plate recognition-based vehicle identification systems, we choose to use the detected license plates as a cue to define the vehicle ROIs. Any real-time and robust license plate detection technique such as [7] can be integrated with our VMMR module. License-plate *recognition* systems are highly failure-prone; license plate *detection* techniques, on the other hand, have been proven to be highly robust to different lighting conditions and have the advantages of higher processing speed, lower computational complexity and minimal failure cases [8].

B. Features Extraction and Global Features Representation

To describe the vehicle makes and models, various local features are extracted from the vehicle ROIs (Features Extraction), with or without embedding them into Global Features Representations. Works such as [4] use raw image features like Scale Invariant Feature Transform (SIFT [9]) to describe make-model instances. In fact, SIFT has been used by many VMMR works such as [4], [5], [10]–[12]. Due to the high dimensionality and relatively slow computational speed of SIFT, some works have adopted the Speeded Up Robust Features (SURF [13]) (e.g., [5], [10]) and the Histogram of Oriented Gradients (HOG [14]) (e.g., in [5] and [15]). Other features based on edges, gradients or corners (e.g., by [2], [3], [16]), and MPEG-7 descriptors such as Edge Histograms [17], [18] (e.g., by [10]) were also explored for VMMR purposes. In most approaches, the raw features are embedded into global representations of vehicle makes and models ([3], [5], [10], [12], [15]) as shown in Table I. Some works (such as [12]) refer to the global representations as *Mid Level Representation (MLR)*. The quality of a global features representation technique is assessed by its processing speed, computational complexity in forming the holistic representations, and VMMR accuracy, which reflects its discriminative power in representing the different makes and models while generalizing over the multiplicity issues within a make-model class.

Edge images of vehicles' faces have been considered in [3] as numerical feature vectors. Pearce and Pears [2] concatenate the Square-Mapped Gradients (SMG) or Locally Normalised Harris Strengths (LNHS) as global feature vectors for the images. Varjas and Tanacs [19] also used concatenated SMG. The SMG-based techniques require well-aligned ROIs with strictly frontal views, or planar projection of skewed views onto frontal-like views. However, as we demonstrate later in the paper, our approaches are greatly successful in achieving a highly accurate VMMR system even under a wide range of viewpoints (or vehicle orientations) without requiring projection onto perfectly frontal views.

A grid-based global representation of features is proposed by Hsieh *et al.* [5], who group the SURF features extracted from frontal vehicle faces in a grid-wise fashion. Chen *et al.* [15], [20] proposed a grid-based concatenation of HOG features from the vehicle images into a global ensemble representation. Using their dataset, we prove that the performance of our approaches is superior. Certain works, such as [21], use the positions and sizes of car emblems (model symbol, trim level, etc.) and HOG features of emblem regions to classify vehicle models, assuming the make is known. However, it is unclear if their approach can achieve both make and model recognition.

Baran *et al.* [10] use local features like SURF to build a dictionary, which is used to represent vehicle images as sparse vectors of occurrence counts of the words of a dictionary. In contrast to their work, we investigate optimal dictionary building parameters in the context of VMMR challenges, through two schemes of dictionary building. Amongst the most recent works on VMMR is that of Fraz *et al.* [12]. They form a lexicon that is comprised of all training images' features as words. The words of the lexicon are computed based on a Fisher Encoded

Mid-Level-Representation (MLR) of image features such as SIFT. Their MLR construction is computationally expensive, reported to consume about 0.4 s per image, and hence unsuitable for real-time VMMR. Unlike [12], we learn a dictionary by retaining only the dominant features of training images as codewords, and not all the features.

C. Classification Approaches

In the literature, there have been various classification approaches proposed for VMMR based on the local features and/or global features representations of the make-model classes. For example, [4] and [11] employed a simple brute-force matching scheme using raw SIFT features to match query images to the gallery images. The brute-force pattern matching approach is very time consuming, and hence unsuitable for real-time VMMR. On the other hand, Munroe and Madden [3] use machine learning algorithms such as C4.5 Decision Trees, k-Nearest Neighbors (kNN), and Feed-forward Neural Networks as classifiers for VMMR. He *et al.* [16] built an ensemble of neural networks for classification and also tested kNN, AdaBoost, and SVM. However, such approaches based on edges from images suffer greatly in cases of occlusions, and hence are not applicable in real-life scenarios [3].

In [2], kNN and Naive Bayes classifiers were tested with a variety of features. A kNN-based classification scheme was also used by Varjas and Tanacs [19], but with a correlation-based distance metric. In these approaches, accuracy is degraded when ROIs are even slightly different than ground truth ROIs. The degradation in accuracy is due to the inefficiency of corner- and gradient-responses based global feature vectors for the VMMR problem. The classification scheme adopted in [12] includes matching a probe words of images with the gallery of lexicons in a brute-force manner. Such an exhaustive matching scheme makes their approach inapplicable to real-time VMMR systems.

Baran *et al.* [10] utilized a simple multi-class SVM trained over sparse occurrence vectors. However, they did not investigate optimization of the dictionaries for VMMR. Unlike them, we propose optimized dictionaries and two SVM-based classification schemes that are designed to solve VMMR issues. Moreover, the superiority of our approaches is proven by using a more challenging dataset. Hsieh *et al.* [5] employ a grid-wise ensemble of SVM classifiers, each of which is trained over SURF features from a specific grid-block over frontal vehicle faces. On the other hand, Chen *et. al.* [15], [20] propose a classification approach for VMMR, based on sparse representation and Hamming Distance.

In spite of the various works that have been published on the theme of VMMR, the multiplicity and ambiguity problems are yet to be solved, perhaps through more representative and discriminating global features representation techniques. Many works rely on strictly frontal view images of cars, and use images with very negligible variation in scale, rotation, and orientation of vehicles. Another major challenge is the lack of a proper benchmark dataset for VMMR. Most studies evaluate their approaches and report the results based on private datasets, which prevents us from comparing our work to theirs. These datasets have several issues that interfere with the reliability and conviction of their performance results. Apart from the unbalanced nature of the datasets, the images are not partitioned into training and testing subsets using any of the standard procedures followed in image classification works. Unlike the related works, to prove the effectiveness of our approaches, we randomly partition the dataset repeatedly into different training and testing subsets a number of times, and average our accuracies across them.

Fig. 7. Examples of the targeted environment where VMMR is needed (Gates of a cross-border checkpoint) to be used by the VMMR systems.

III. Target Environment & Dataset Description

To demonstrate the effectiveness of the proposed BoSURF approaches for VMMR, we target scenarios such as the entrances or exits of parking facilities at public places such as malls, airports, stadiums, etc. (See Fig. 7). Such public areas are highly vulnerable to security threats. The camera(s) are fixed on the entrance(s)/exit(s) of a given parking facility. Vehicles may be occluded by pedestrians or other objects. The proposed approaches for VMMR can be easily applied to other scenarios in which the camera is not fixed, e.g., an on-board camera on a mobile surveillance vehicle, etc.

The above characteristics of the target environment are closely represented by the NTOU-MMR dataset [6], which is a very recent and publicly available dataset for vehicle makes and models with published results of several MMR works. Hence, it serves as a good benchmark dataset to compare performances of our approach with other works. In what follows, we further describe the dataset and note a few problems therein.

Published in the recent related work of Hsieh *et al.* [5], the NTOU-MMR dataset was collected under the Vision-based Intelligent Environment (VBIE) project [22] and can be accessed at [6]. Speeds of up to 65 km/h were allowed for the oncoming vehicles. The original dataset is divided into a training and a testing set collected in different weather conditions as explained bellow. There are 2,846 images for training, and 3,793 images for testing. The total number of classes is 29.

The motivation to use this dataset in our work stems from the following characteristics of the dataset. The images have vehicles in different viewing angle pans ranging from -20° to 20°, which sufficiently represent real-life scenarios. Moreover, the images of the dataset were taken throughout the daytime and night-time, and under weather conditions varying between sunny, cloudy and rainy. In addition, there are also images with vehicles occluded by irrelevant objects (such as pedestrians). As we shall present in Section IX, the effectiveness of our

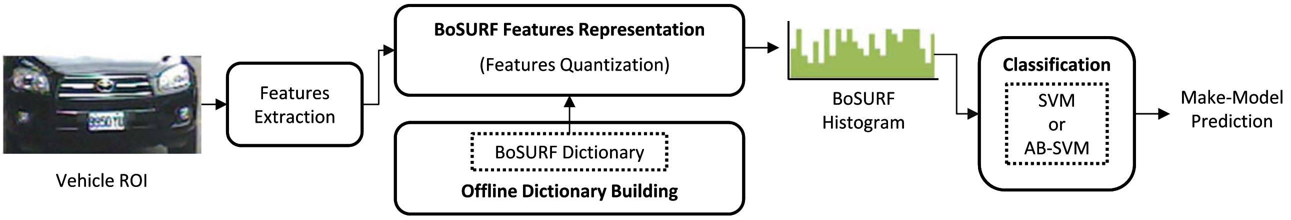


Fig. 8. Overview of our BoSURF approach for VMMR.

approaches can be proven even in such challenging scenarios (See Fig. 17).

However, we note some problems with the NTOU-MMR dataset (downloaded from [6]: (1) Wrongly placed images: some class directories have images belonging to other classes, (2) Duplicated images: many classes have duplicate images (with different names), (3) Biased partitioning of training and testing data: it is unclear which strategy is employed to partition data into training and testing for each class. The manner in which data is partitioned into training and testing greatly impacts performance results. A biased partitioning can give misleading results of the accuracy, as we demonstrate in Fig. 14 (Section IX-D).

IV. Overview of BoSURF-MMR

The BoSURF-MMR module proposed in this paper is illustrated in Fig. 8. We extract SURF features [13] from training samples of all classes and retain the dominant ones in a “bag” or *dictionary*, hence the name Bag-of-SURF (BoSURF). Note that we have employed the 64-dimensional SURF. The dictionary is then used to represent the vehicles’ images as BoSURF histograms. Our work is inspired by the popular Bag-of-Features (BoF) framework [1], [23] which has been widely used to describe objects of interest using their raw image features embedded into global representations. The BoF has been very successful and widely adopted in the works on object recognition [24], scene classification [25], image classification [26], and image retrieval [27], [28], etc. However, to the best of our knowledge, BoF has not been extensively studied in the context of VMMR. Encouraged by the success of BoF in the aforementioned works, we propose and evaluate BoSURF-based approaches for the VMMR module.

There are three main steps involved in the proposed BoSURF approaches for VMMR, as shown in Fig. 8: (1) Offline Dictionary Building (See Section V), (2) BoSURF Features Representation (See Section VI) and (3) Classification (See Section VII).

In this work, we investigate two dictionary building schemes in the context of real-time VMMR: (1) *Single Dictionary (SD)* (See Section V-A), and (2) *Modular Dictionary (MD)* (See Section V-B). The SD is based on the standard method of dictionary building in the BoF framework, in which dictionary codewords are learned from the collective pool of training data (i.e., of local image features) from all combined classes. The MD, on the other hand, is composed of many individual dictionaries, each corresponding to a make-model class. The codewords of each such sub-dictionary are learned from the training data of the respective make-model class.

In Features Extraction, SURF has gained wide popularity in many computer vision applications. It has been shown to have higher accuracy and speed in comparison to other feature descriptors in the context of object recognition, image classification, etc. [13]. Both the *Offline Dictionary Building* and the *BoSURF Features Representation* steps rely on local image features such as SURF [13]. In fact, SURF can be easily replaced with any good feature descriptor in our BoSURF MMR module. We had also explored using Scale-Invariant Feature Transform (SIFT) instead of SURF, but the results were not encouraging. This is due to high dimensionality and relatively slow computational speed of SIFT. Hence, we choose to employ the SURF features as the building blocks of our BoSURF-based approaches for VMMR. To build the BoSURF representations from SURFs of vehicle ROIs, SD or MD are used.

The BoSURF representations from different vehicle makes and models are then used to train multi-class classifiers to be used in VMMR testing. We present and evaluate two multi-class classification schemes for VMMR: (1) Single Multi-Class SVM Classifier, referred to as SVM (See Section VII-A), and (2) Attribute Bagging based Ensemble of Multi-Class SVM Classifiers, referred to as AB-SVM (See Section VII-B).

V. Offline Dictionary Building

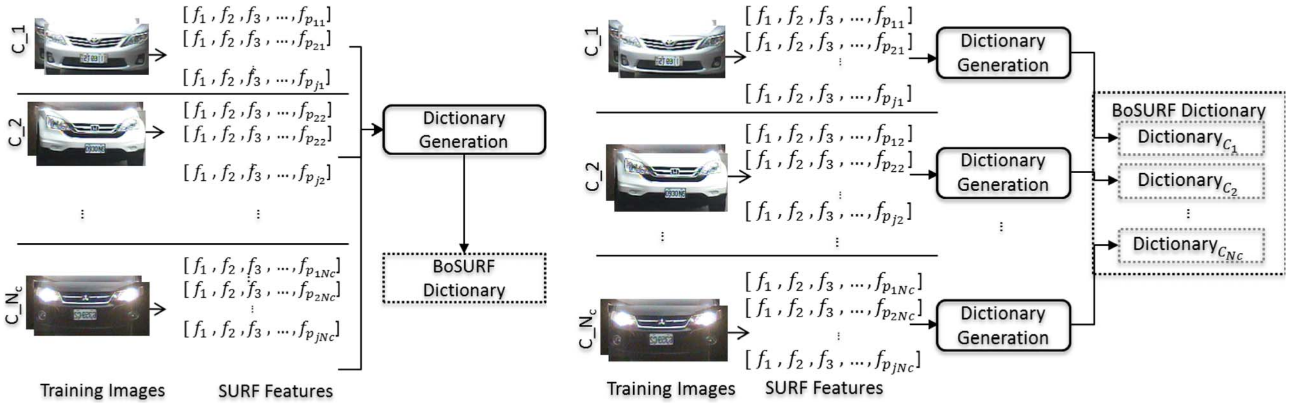
The training images of all classes are used to extract their SURF features [13]. The dominant features (*codewords*) are then retained in a “bag” or *dictionary*. We capture and describe the overall appearance of the front or rear face for each vehicle make and model using the built dictionary. The dictionary can be considered as a compact representation comprised of the dominant features (codewords) from training images of all classes. The vehicles’ images are represented as BoSURF features which are histograms of occurrences of the dictionary codewords. Building the dictionary is usually done offline and only when needed, so that it may be used in the training and testing phases. The two dictionary schemes evaluated in this work are described in this section. An overview of the SD and MD schemes is depicted in Fig. 9(a) and (b) respectively.

Let $\mathbf{I}$ represent the set of training images for N_c number of classes as shown in Equation (1), where $\mathbf{I_i}$ represents the set of training images of class i in the dataset being used.

$$\mathbf{I} = \{\mathbf{I_1}, \mathbf{I_2}, \ldots, \mathbf{I_{N_c}}\}. \tag{1}$$

From each j-th image in $\mathbf{I_i}$, we extract its set of SURF features, F_{ji}, as represented in Equation (2):

$$F_{ji} = \{f_1, f_2 \ldots, f_{p_{ji}}\} \tag{2}$$


C_1
C_2
C_N_c
$[f_1, f_2, f_3, \ldots, f_{p_{11}}]$
$[f_1, f_2, f_3, \ldots, f_{p_{21}}]$
$[f_1, f_2, f_3, \ldots, f_{p_{j1}}]$
$[f_1, f_2, f_3, \ldots, f_{p_{12}}]$
$[f_1, f_2, f_3, \ldots, f_{p_{22}}]$
$[f_1, f_2, f_3, \ldots, f_{p_{j2}}]$
$[f_1, f_2, f_3, \ldots, f_{p_{1Nc}}]$
$[f_1, f_2, f_3, \ldots, f_{p_{2Nc}}]$
$[f_1, f_2, f_3, \ldots, f_{p_{jNc}}]$
Dictionary Generation
BoSURF Dictionary
Dictionary$_{C_1}$
Dictionary$_{C_2}$
Dictionary$_{C_{Nc}}$
Training Images
SURF Features

(a) (b)

Fig. 9. Offline dictionary building. (a) Single-dictionary building scheme and (b) modular dictionary building scheme of our BoSURF-MMR.

```
Input: The sets of images from all classes, Dictionary sizes for SD and MD
Output: $\mathbf{D}$ or $\mathbf{D_M}$
Initialize: $\mathbf{M_F}$
Step 1: Collecting Local Features
for class $i = 1$; $i \leq N_c$; $i++$ do
    Initialize $\mathbf{F_i}$
    for image $j = 1$; $j \leq N_i$; $j++$ do
        $F_{ij} \leftarrow$ FeaturesExtraction$(I_{ij})$
        $\mathbf{F_i} = \mathbf{F_i} \cup \{\mathbf{F_i}\}$
    end
    $\mathbf{M_F} = \mathbf{M_F} \cup \mathbf{F_i}$
end
Step 2: Dictionary Building
Initialize: Single-Dict., Modular-Dict. and Sub-dic.
Build SD: $\mathbf{D} \leftarrow$ Cluster$(\mathbf{M_F}, S_{SD})$
Build MD: for class $i = 1$; $i \leq N_c$; $i++$ do
    $\mathbf{D} \leftarrow$ Cluster$(\mathbf{F_i}, S_{DI})$
    $\mathbf{D_M} \leftarrow \mathbf{D_M} \cup \mathbf{D_i}$
end
```

Algorithm 1: Offline dictionary building

where $f_{p_{ji}}$ is the p-th SURF feature in image-j and p_{ji} is the number of SURF features extracted from the j-th image of class i.

The pool of features from images of all classes is represented by $\mathbf{F}$ in Equation (3):

$$\mathbf{F} = \{\mathbf{F_1}, \mathbf{F_2}, \ldots, \mathbf{F_{N_c}}\}. \tag{3}$$

A. Single Dictionary (SD)

To build the SD (denoted by $\mathbf{D}$), dominant features are selected by performing clustering on $\mathbf{F}$. For that purpose, Kmeans++ clustering technique is used because it yields an optimal solution compared to KMeans. This involves grouping the training features into a number of clusters of similar patterns. The most commonly used clustering techniques are: K-Means, K-Means++, and Meanshift [29].

The cluster centres are the selected dominant features that make up the dictionary, and are referred to as the *visual words*, or *codewords*, represented by cw_k in Equation (4) (further described in Section VIII-A). The number of selected dominant features are determined in Section VIII. The number of clusters (or codewords) determines the overall Dictionary Size, S_D. See Fig. 9(a) for an overview of the SD scheme. We refer to the BoSURF approach based on the SD scheme as BoSURF-SD.

$$\mathbf{D} = \{cw_k | k = 1, \ldots, S_D\}. \tag{4}$$

B. Modular Dictionary (MD)

In this second scheme of dictionary building, we build the main dictionary (denoted by $\mathbf{D_M}$) by combining individual dictionaries of each class, motivated by the results of [29]. The intuition behind this scheme is that, in the Single Dictionary scheme, several distinct features could be clustered under the same codeword due to their closeness. More importantly, having a modular dictionary greatly reduces the time consumed in dictionary building and also provides flexibility. If classes need to be added (removed), their respective dictionaries can be flexibly appended (deleted) to (from) the main dictionary without requiring a reconstruction the entire dictionary, thus saving a considerable amount of time. The *Modular Dictionary* $(\mathbf{D_M})$ is formed as:

$$\mathbf{D_M} = \{\mathbf{D_i} | i = 1, 2, \ldots, N_c\} \tag{5}$$

$$\mathbf{D_M} = \{cw_k | k = 1, \ldots, (S_{Di} \cdot N_c)\} \tag{6}$$

where each $\mathbf{D_i}$ is the individual dictionary of class i, built by retaining S_{Di} dominant features (codewords) out of $\mathbf{F_i}$ by the similar clustering procedure as mentioned in Section V-A. The size of the overall dictionary $\mathbf{D_M}$, and hence the number of codewords (cw_k), is then $S_D = S_{Di} \cdot N_c$. See Fig. 9(b) for an overview of the MD scheme. The BoSURF-MMR approach based on the MD scheme is referred to as BoSURF-MD.

C. Size of the Dictionary

The dictionary size (S_D) is an important parameter that affects processing speed, *discriminative capacity* and *generalizability* of the built dictionary, and hence affects the overall performance of the BoSURF approach. A small dictionary could suffer due to reduced discriminatory capacity. In small dictionaries, more than one feature could get assigned to the same cluster, despite being different. On the other hand, a large dictionary loses capacity for generalization, adds higher penalties to noises, and increases processing overhead [30]. As a contribution of this work, we study the effect of various dictionary sizes (for both SD and MD schemes) on overall VMMR speed and accuracy (further described in Section VIII-A).

VI. BoSURF Features Representation

The second step uses the dictionary $\mathbf{D}$ or $\mathbf{D_M}$ to embed given images' local features into global BoSURF representations through *Features Quantization* (See Fig. 8). For a given image I_j of class i, its BoSURF features representation is a histogram H_{ij}, of votes to the dictionary codewords. The histogram H_{ij} can be represented by (See Fig. 8):

$$H_{ij} = \{h_k | k = 1, \ldots, S_D\} \tag{7}$$

where the bins (h_k) hold the number of votes to the respective codewords (cw_k), respectively. To build the BoSURF features representation (histogram) of image I_j, each SURF feature $f_{p_{ji}}$ from I_j is matched to its nearest codeword cw_k of the dictionary ($\mathbf{D}$ or $\mathbf{D_M}$), and the corresponding histogram bin h_k vote-count is incremented. This step is also referred to as *Features Quantization*. In this manner, we obtain the final histogram after matching all features of a given image, and we call it a BoSURF histogram or feature. The BoSURF histogram for the set of SURF features F_{ji} of a given image I_j of class i is computed as follows:

$$H_{ij}(k) = \frac{1}{p_{ji}} \sum_{p=1}^{p_{ji}} \begin{cases} 1 & \text{if } k = \underset{t \in [1, S_D]}{\arg\min} \operatorname{dist}\left(cw_t, f_{p_{ji}}\right) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where dist(a, b) is the euclidean distance between features a and b, and $H_{ij}(k) = h_k$; $f_{p_{ji}}$ is the pth SURF feature and p_{ji} is the number of SURF features extracted from the image I_j.

VII. Classification

The third step includes training a classifier over the BoSURF features of all training images, to be used subsequently in VMMR testing. In this work, we propose two multi-class Support Vector Machine (SVM)-based classification schemes for VMMR. The SVM [31], [32] is a very effective binary classifier in which the support vectors are a subset of the training data samples representing the best separation between two classes. A test data sample is classified based on its distance from these support vectors. A collection of many such binary classifiers are used to build a single multi-class SVM classifier. We have conducted extensive cross-validation experiments to find the optimal SVM parameters, because the datasets are usually unbalanced (see Section VIII). The two approaches for multi-class classifier training and testing presented are: (A) Single Multi-Class SVM Classifier (referred to as SVM), and (B) Ensemble of Multi-class SVM Classifiers based on Attribute Bagging (referred to as AB-SVM).

A. Single Multi-Class SVM Classifier

For each training image of given classes in the training phase, SURF features are extracted and embedded into the BoSURF histograms using the Single Dictionary (or the Modular Dictionary), as described in Section VI. These BoSURF histograms from all training images are collected and used to train the multi-class SVM classifier. For testing, the BoSURF histogram of the given test image is generated using the same dictionary used in training. Based on this histogram, each of the binary classifiers that make up the multi-class SVM adds a vote to its predicted class. The class with the highest votes is assigned as the predicted make-model class of the test image.

B. Ensemble of Multi-Class SVM Classifiers Based on Attribute Bagging

Instead of training a single classifier over the entire set of feature vectors, we explore the idea of building an ensemble of individual multi-class classifiers that are trained over different random feature subspaces (i.e., random feature subsets). This technique is referred to by different names in the literature: *Attribute Bagging (AB)*, *Multiple Feature Subsets*, and *Random Subspace Method*. The Random Subspace Method is a more generic term which could refer to: (1) applying Random Subsampling over the training data samples to create bootstrap subsets of the training dataset, or, (2) applying Random Subsampling over the feature-space to create random subsets of feature-vectors (used in this work). We prefer to use the term *Attribute Bagging*, as it best describes the technique with which feature subsets are created. In testing, the predictions from each of the classifiers in the ensemble are combined using a certain combination rule to produce the final prediction.

1) Motivation to Use Attribute Bagging (AB): The motivation to adopt AB for training the individual classifiers of the ensemble arises from the following observation. In the MMR dataset used in this work, the training samples per class are too few in number when compared to our feature vector dimensions, which could lead to over-fitting problems for classifiers such as SVM. For example, while the average number of training samples per class is 182 in the 80-20 Dataset versions we make from NTOU-MMR Dataset (See Table III and Section III), the best performing feature vector length is 2000. Remember that the feature vector length is equal to the size of the dictionary used to generate the feature, and that each attribute of the feature vector corresponds to the votes assigned to the respective dictionary codeword. Employing an ensemble of classifiers built using AB helps to avoid over-fitting problems.

To avoid the over-fitting issue by reducing the difference between the number of training data samples and the feature vector dimensions, random subsets of feature vectors are created. We do not create random subsets of the training dataset, but we create random subsets out of training feature vectors. Moreover, several works in the literature (e.g., [24]) have shown that the AB-based ensemble of classifiers could perform better than the stand-alone individual classifiers (i.e., classifiers trained over whole feature vectors). Motivated by their findings, we are interested in exploring whether the AB-based ensemble of multi-class SVM classifiers, hereby referred to as AB-SVM, trained over BoSURF representations, could improve the performance of VMMR in comparison to the single classifier scheme of Section VII-A.

TABLE II
NOMENCLATURE (FOR SYMBOLS USED IN SECTION VII-B)

Symbol	Meaning
N_{ss}	# of Random Feature Subspaces, # of Classifiers in Ensemble
S_{ss}	Size of (#Attributes in) a Random Feature Subspace
N_i	# of training images of class-i
N_{tr}	$= \sum_{i=1}^{N_c} N_i$ (i.e., total number of training images)
S_D	Size of SD or MD, #Codewords in dictionary
A_g	Set of randomly chosen attribute indices for g^{th} subspace
H_{ij}	BoSURF feature vector of image-j of class-i
H_{ij}^g	g^{th} feature subset of H_{ij}, extracted based on A_g
$\mathbf{H_i^g}$	Training set of class-i from g^{th} feature subspace (containing all H_{ij}^g)
$\mathbf{H^g}$	Training set of all classes from g^{th} feature subspace (containing all $\mathbf{H_i^g}$)
C^g	a multi-class SVM classifier trained over $\mathbf{H^g}$
C	the ensemble of C^g
a_x	the value of the x^{th} attribute

2) Creating Random Feature Subsets by AB: We will illustrate the AB method of creating feature subspaces through a simple example (see Table II for the definition of symbols). Assume we have BoSURF feature vectors such as $F = \{a_x | x = 1, \ldots, 10\}$, where a_x is the value of the xth attribute. Let the number of random feature subspaces to create be $N_{ss} = 4$, each comprising of $S_{ss} = 5$ attributes (or dimensions). Let A_g denote the set of randomly chosen attribute indices for the gth feature subspace, as described by Equation (9), where $|A| = S_{ss}$, and $g = 1, \ldots, N_{ss}$. For example, consider $A_1 = \{9, 6, 2, 5, 8\}$, $A_2 = \{1, 7, 9, 3, 4\}$, $A_3 = \{8, 1, 6, 10, 3\}$, and $A_4 = \{5, 1, 2, 6, 9\}$. So, out of each original feature vector F, we would extract $N_{ss} = 4$ random feature subsets based on A_1, A_2, A_3, and A_4 respectively, resulting in the following random feature subsets out of F: $F_1 = \{a_9, a_6, a_2, a_5, a_8\}$, $F_2 = \{a_1, a_7, a_9, a_3, a_4\}$, $F_3 = \{a_8, a_1, a_6, a_{10}, a_3\}$, and $F_4 = \{a_5, a_1, a_2, a_6, a_9\}$. We would then build an ensemble of $N_{ss} = 4$ classifiers, each trained over the respective feature subspace

$$A_g = \{x | x \in [1, S_D]\}. \tag{9}$$

Now, let us generalize the application of AB over our training dataset. Let H_{ij} be the BoSURF feature vector for a given image j of class i, comprised of S_D attributes:

$$H_{ij} = \{h_{jk} | k = 1, \ldots, S_D\}. \tag{10}$$

To create each feature subset H_{ij}^g, we must randomly select $S_{ss} < S_D$ different attributes (without replacement) from H_{ij}, based on the attribute indices in A_g. Sampling without replacement ensures that, within a subset, each attribute is selected only once. However, an attribute could be chosen in more than one subset. All such feature subsets of class i (i.e., H_{ij}^g), are collected in $\mathbf{H_i^g}$ as shown in Equation (11) where the dimensions of $\mathbf{H_i^g}$ are $(N_i) \times (S_{ss})$

$$\mathbf{H_i^g} = \left[H_{i1}^g, H_{i2}^g, \ldots, H_{iN_i}^g\right]^T. \tag{11}$$

The classwise pools of feature subsets $(\mathbf{H_i^g})$ for all classes $i = 1, \ldots, N_c$ are then collected in the respective overall training set for the gth feature subspace $(\mathbf{H^g})$ as shown in Equation (12):

$$\mathbf{H^g} = \left[\mathbf{H_1^g}, \mathbf{H_2^g}, \ldots, \mathbf{H_{N_c}^g}\right]^T \tag{12}$$

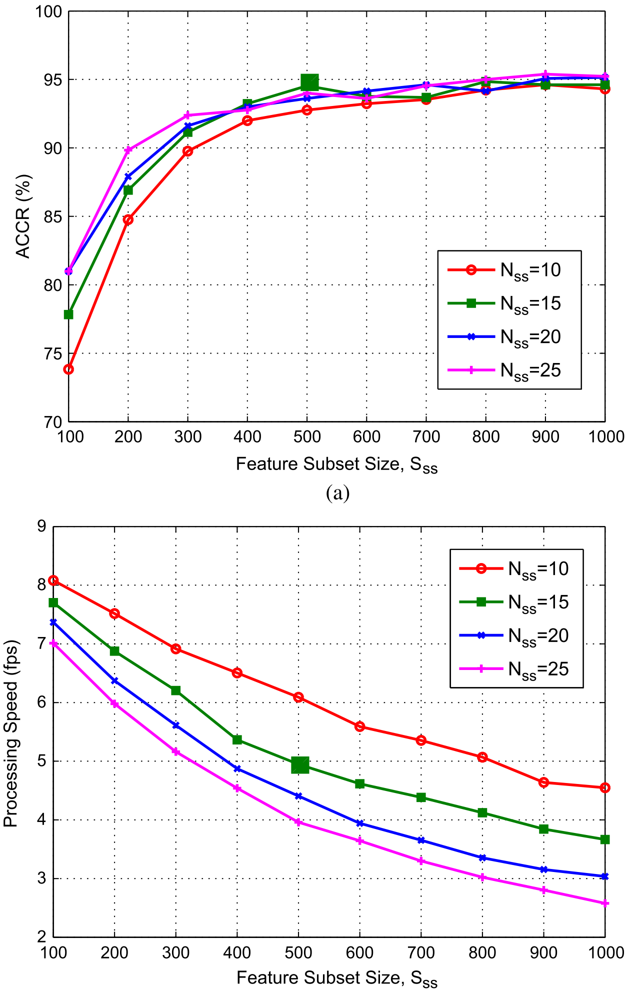
where each $\mathbf{H^g}$ is of dimensionality $N_{tr} \times S_{ss}$. The feature subsets in $\mathbf{H^g}$ are used to train the corresponding multi-class classifier C^g. In this way, we achieve an ensemble of classifiers C (composed of C^g), to predict make and model of a vehicle in given test images. A greater value for N_{ss} will yield a larger number of feature subspaces, which would increase the chances of having qualitatively different C^g, as discussed by [33].

3) Classification: In the testing phase, the BoSURF feature vector H_t is sub-sampled into subsets H_t^g based on the respective set of attribute indices A_g. To create each subset, the same sequence of S_{ss} attributes indices (given by A_g) that were selected in creating $\mathbf{H^g}$ are used. The classifier C^g is then used to predict the label of H_t^g, adding a vote to the winning class. We employ a majority-voting scheme to combine the outputs of all the C^g to produce the final prediction of C. The class that wins the majority of the votes is produced as the predicted make and model of the test image.

4) Choosing Optimal Parameters: Given that the dimensionality of the BoSURF feature vectors is quite high, if the number of feature subspaces N_{ss}, and hence the number of classifiers, is not sufficiently large enough, there could be cases where some attributes may never be chosen. In the illustrative example given at the beginning of Section VII-B2, if $N_{ss} = 2$, and say we choose the two feature subspaces as per A_1 and A_2 only, then we see that the 10th attribute is left out. Similarly, if A_1 and A_3 are chosen, 4th and 7th attributes would be left out in the resulting feature subsets. It may occur that the omitted attributes had high discriminative capacity or significance. Based on the exhaustive experimental evaluations, we have selected the optimal values for N_{ss} and S_{ss} as explained in Section VIII.

In Fig. 10(a) and (b), we show the effect of feature subspace sizes (S_{ss}) and the number of feature subspaces (N_{ss}) on the processing speed and accuracy of BoSURF-SD with AB-SVM. That is, we have varied S_{ss} from 100 to 1000 (in steps of 100), for each test with $N_{ss} = 10, 15, 20,$ and 25. It is clear to see from Fig. 10(a) and (b) that while accuracy tends to increase with the increase in S_{ss} and N_{ss}, speed tends to decrease. This is obvious because a higher S_{ss} indicates greater dimensionality of the feature vectors, and a higher N_{ss} represents a greater number of classifiers in the ensemble, both of which lead to an increase in processing time consumption.

To find the optimal values of N_{ss} and S_{ss} for BoSURF-SD, we observe the accuracy vs. speed plot as shown in Fig. 11. In this figure, we see that for accuracies above 94%, the speed tends to fall drastically while accuracy improves only slightly towards 95%. With an objective of achieving a processing speed of at least 5 fps to meet real-time requirements while having an accuracy of around 95%, we observe from these figures that $S_{ss} = 500$ and $N_{ss} = 15$ gives an accuracy of 94.5% and speed of around 5 fps (represented by the highlighted green square datapoint in Figs. 10(a), (b), and 11). Similar experiments were conducted for AB-SVM based BoSURF-MD and $N_{ss} = 15$, $S_{ss} = 1500$ yielded the best speed-accuracy trade-off in our targeted environment, and using the dataset NTOU-MMR, as defined in Sections III and VIII. So, we adopt the values ($N_{ss} = 15$, $S_{ss} = 500$) and ($N_{ss} = 15$, $S_{ss} = 1500$) for the AB-SVM based BoSURF-SD and BoSURF-MD approaches respectively, in the later experiments (further discussed in Section VIII).



(a)

(b)

Fig. 10. Effect of feature subset sizes S_{ss} and number of subsamples N_{ss} on (a) average correct classification rate and (b) processing speed of BoSURF-SD with AB-SVM. The computer used to achieve these results is Intel Core i5 3475S CPU (2.94 GHz), with 16-GB RAM.

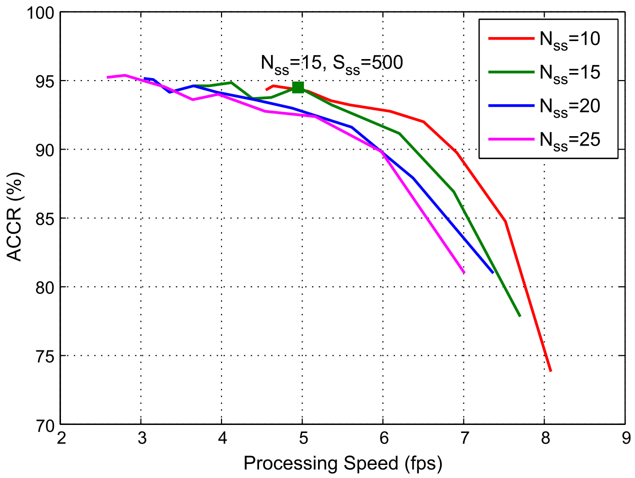


Fig. 11. ACCR versus processing speed of BoSURF-SD with AB-SVM for different number of subsamples N_{ss} and feature subset sizes S_{ss}. The best speed–accuracy tradeoff is at N_{ss} = 15 and S_{ss} = 500.

TABLE III
VEHICLE MAKE–MODEL CLASSES AND THE NUMBER OF TRAINING (#Tr) AND TESTING (#Te) IMAGES IN THE 80-20 DATASETS

Make	Toyota					
Model	Altis	Camry	Vios	Wish	Yaris	Previa
#Tr	972	556	547	240	244	50
#Te	240	136	136	58	60	12
Make	Toyota				Suzuki	Nissan
Model	Innova	Surf	Tercel	RAV4	Solio	Tiida
#Tr	36	67	145	127	101	208
#Te	9	16	36	31	25	51
Make	Nissan					
Model	March	Livna	Teana	Sentra	Cefiro	Xtrail
#Tr	157	203	76	77	123	111
#Te	38	50	19	18	30	27
Make	Mitsubishi				Honda	
Model	Zinger	Outlander	Savrin	Lancer	CRV	Civic
#Tr	26	36	43	64	496	265
#Te	6	8	10	16	123	66
Make	Honda	Ford				Total
Model	FIT	Liata	Escape	Mondeo	Tierra	
#Tr	84	16	90	91	52	5302
#Te	20	3	22	20	13	1299
						6601

VIII. EXPERIMENTAL SETUP

In this paper, we propose and investigate unexplored approaches for real-time automated Vehicle Make and Model Recognition (VMMR) based on BoSURF and SVMs. The three main steps in our VMMR approach, as depicted in Fig. 8, are: (1) *Offline Dictionary Building* (to be used for Features Quantization), (2) *BoSURF Features Generation* and (3) *Classification*, which involves classifier training and testing.

The focus of our work, like most related works [2], [3], [5], [10], [12], [20], etc. is on the use of the front or rear faces of the vehicles for VMMR. This is based on the observation that the other regions of the image of a vehicle, such as the hood, windshield, etc. have very little dissimilarity across different makes and models. Including features from such regions could lead to classifier confusion and many false positives. In the datasets we use to prove the effectiveness of our approaches and to compare against other VMMR works, there are only frontal views of vehicles (dataset described in Section III). Hence, we will base our experiments and discussions on vehicle front faces only. However, our approach can be easily applied to datasets having rear face images as well. In all our experiments, the computing platform used is an Intel Core i5 3475S CPU (2.94 GHz), with 16 GB RAM, very similar to that used by the works we compare our results with.

Unlike previous works based on the dataset, we repeatedly randomly partition the original NTOU-MMR dataset to form N_D number of different training and testing splits for each class. For each split, 80% of images are randomly chosen for training, and the remaining 20% for testing. We refer to these as 80-20 NTOU-MMR Datasets, or simply 80-20 Datasets. Table III outlines the number of training (#Tr) and testing (#Te) images in each of the N_D datasets. The mean accuracies and processing speeds of our approaches are determined by averaging the results over the N_D datasets. The 80-20 ratio for the training and testing split is one of the standard dataset partitioning schemes employed by many works in object recognition and image classification.

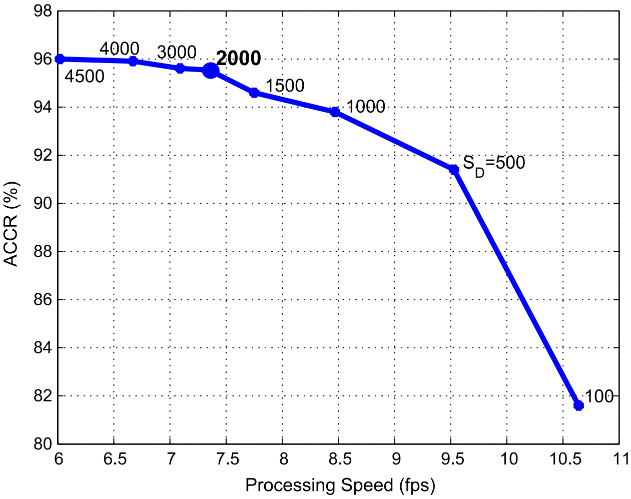


Fig. 12. Effect of varying S_D (from 100 to 4000) on accuracy and speed of BoSURF-SD-based VMMR. $S_D = 2000$ yields the best tradeoff between speed and accuracy.

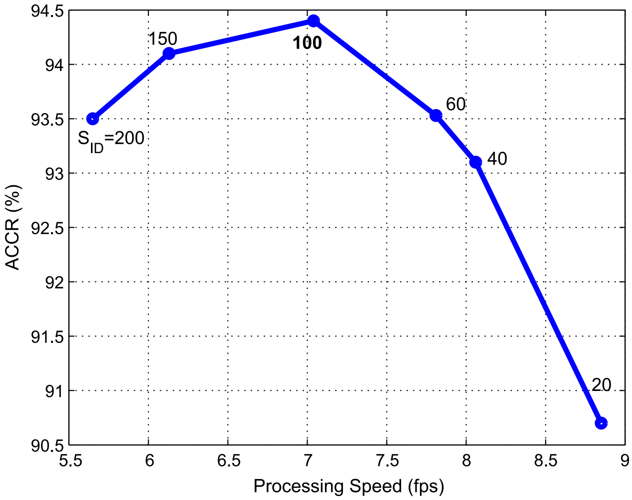


Fig. 13. Effect of varying size of individual dictionaries S_{Di} from 20 to 200, on accuracy and processing speed of BoSURF-MD-based VMMR. At $S_{Di} = 100$ (and thus the overall size of MD $S_D = S_{Di} \cdot N_c = 100 \cdot 29$), we obtain the best trade-off between speed and accuracy.

A. *Optimal Parameters Selection*

We obtain the optimal parameters for each step of our BoSURF MMR approach by cross-validation, using the 80-20 Datasets described in Section III. In the Offline Dictionary Building step, the significant parameter affecting the processing speed and accuracy of the overall VMMR module is the Dictionary Size, S_D. Varying the S_D in the SD scheme from 100 to 4000, we found that $S_D = 2000$ yielded the best tradeoff between speed and accuracy (as shown in Fig. 12). With $S_D = 2000$, we obtain an accuracy of 95.54% and speed of 7.4 fps. Although a greater S_D such as 4000 results in a higher accuracy (96%), the speed is reduced to 6.7 fps. So, we choose $S_D = 2000$ in our experiments, unless otherwise stated. As for the MD scheme, we conducted similar experiments by varying the size of individual dictionaries (S_{Di}) and found that the $S_{Di} = 100$ (which makes the overall MD of size $S_D = S_{Di} \cdot N_c = 100 \cdot 29$), yielded the best trade-off between speed and accuracy (See Fig. 13). Increasing the S_{Di} beyond 100 gradually decreases accuracy and speed. Hence, in our experiments based on BoSURF-MD, we adopt $S_{Di} = 100$.

Based on the obtained optimal dictionary sizes, we then find the optimal classifier parameters. For the Classification step, we utilize the multi-class SVM library of OpenCV [34] which is based on LibSVM [31], [35] to build the two classification approaches proposed in this work. The values $C = 50$, $\gamma = 5$ for the SVMs were empirically determined to yield the best results. As discussed in Section VII-B, the optimal parameters to build the Attribute Bagging based ensemble of SVM classifiers (AB-SVM) are $N_{ss} = 15$, $S_{ss} = 500$ for BoSURF-SD, and $N_{ss} = 15$, $S_{ss} = 1500$ for BoSURF-MD.

B. *Performance Metrics*

In order to be used in real-life scenarios, a good VMMR system needs to meet real-time processing speed requirements, apart from being accurate. For the processing speed of a VMMR approach, we take the inverse of average time taken per image (in seconds) in extracting features, building the global features representation, and classifying it to predict the MMR class label. We report the processing speed of the VMMR approaches in frames-per-second (fps).

Let $L = \{l_i | i = 1, 2, \ldots, N_c\}$ be the set of labels for all N_c number of classes in a dataset. The accuracies of the VMMR approaches can be represented by the following metrics:

- The average classwise accuracies, based on the ratio of the number of correctly classified images of l_i to the total number of test images for l_i, averaged over N_D dataset splits.
- Mean Average Correct Classification Rate: the overall VMMR accuracy, a metric similar to [36], which is the ratio of the total number of correctly classified images (of all classes) to the total number of test images in the dataset, averaged over N_D different dataset splits.

To visualize the discriminative capabilities of VMMR approaches, the confusion matrix serves as a good tool. While the row indices of the matrix correspond to Ground Truth class labels, the column indices correspond to Predicted class labels. The value at rth row and cth column, i.e., at (r, c), represents the percentage of images of r predicted to be of class c by the VMMR approach. The main diagonal values represent the ACCR_{c_r}, i.e., at each (r, r), the value is the ACCR_{c_r} for Class-r.

The confusion matrix [as shown in Fig. 15(a) and (b)] helps us identify the classes which could be apparently similar (in the feature space) and could be leading to inaccurate predictions or classifications.

IX. Results & Discussions

A. *Performance of SVM-Based BoSURF-MMR*

Using the single multi-class SVM classifier of Section VII-A (simply referred to as SVM), we investigate the performance of BoSURF with the two dictionary building schemes proposed to solve the issues in VMMR: Single-Dictionary and Modular-Dictionary.

TABLE IV
PERFORMANCE OF OUR BOSURF-MMR WITH SD AND SVM

Make	Toyota					
Model	Altis	Camry	Vios	Wish	Yaris	Previa
ACCR(%)	99	97.35	97.57	96.38	98	90.83
Avg #Cor	237.6	132.4	132.7	55.9	58.8	10.9
Make	Toyota				Suzuki	Nissan
Model	Innova	Surf	Tercel	RAV4	Solio	Tiida
ACCR(%)	95.56	90.63	89.72	99.35	94.40	92.94
Avg #Cor	8.6	14.5	32.3	30.8	23.6	47.4
Make	Nissan					
Model	March	Livna	Teana	Sentra	Cefiro	Xtrail
ACCR(%))	93.42	96.00	89.47	65.56	75.67	92.96
Avg #Cor	35.5	48	17	11.8	22.7	25.1
Make	Mitsubishi				Honda	
Model	Zinger	Outlander	Savrin	Lancer	CRV	Civic
ACCR(%)	80.00	75.00	82.00	98.13	99.43	96.67
Avg #Cor	4.8	6	8.2	15.7	122.3	63.8
Make	Honda	Ford				Overall
Model	FIT	Liata	Escape	Mondeo	Tierra	
ACCR(%)	90.50	66.67	79.09	88.00	80.77	94.84
Avg #Cor	18.1	2	17.4	17.6	10.5	1232

TABLE V
PERFORMANCE OF OUR BOSURF-MMR WITH MD AND SVM

Make	Toyota					
Model	Altis	Camry	Vios	Wish	Yaris	Previa
ACCR(%)	98.67	97.65	97.21	95.00	97.50	90.83
Avg #Cor	236.8	132.8	132.2	55.1	58.5	10.9
Make	Toyota				Suzuki	Nissan
Model	Innova	Surf	Tercel	RAV4	Solio	Tiida
ACCR(%)	95.56	90.63	88.61	98.06	93.20	88.82
Avg #Cor	8.6	14.5	31.9	30.4	23.3	45.3
Make	Nissan					
Model	March	Livna	Teana	Sentra	Cefiro	Xtrail
ACCR(%)	93.42	96.00	89.47	65.56	75.67	92.96
Avg #Cor	35.5	48	17	11.8	22.7	25.1
Make	Mitsubishi				Honda	
Model	Zinger	Outlander	Savrin	Lancer	CRV	Civic
ACCR(%)	83.33	80.00	75.00	99.38	99.27	94.24
Avg #Cor	5.0	6.4	7.5	15.9	122.1	62.2
Make	Honda	Ford				Overall
Model	FIT	Liata	Escape	Mondeo	Tierra	
ACCR(%)	89.00	70.00	80.00	84.00	77.69	93.70
Avg #Cor	17.8	2.1	17.6	16.8	10.1	1217.2

1) Speed: The average processing speeds of the SD- and MD-based BoSURF approaches with SVM are 7.5 fps and 6.99 fps respectively, which proves the suitability of BoSURF for real-time VMMR applications. Higher speeds can be obtained with a slight compromise in accuracy by decreasing the S_D (as previously shown in Figs. 12 and 13). Depending on the requirements of the specific application, the BoSURF parameters could be easily adapted to meet high processing speeds with a slight compromise in accuracy, or vice-versa. Common surveillance cameras have a frame rate of 25–30 fps [37], [38]. However, to run the VMMR on each and every incoming frame would waste computational resources. Instead, every 5th incoming frame can be processed seamlessly for VMMR purposes, which effectively requires only 5–6 frames per second to be processed. In this manner, both of our BoSURF approaches are highly suitable for real-time VMMR applications.

2) Accuracy: We show the classwise ACCR_{c_i} and the average number of correctly classified images for the BoSURF approaches based on SD and MD in Tables IV and V, respectively. The accuracies are averaged over $N_D = 10$ random 80-20 training-testing dataset splits. The dictionary sizes used are $S_D = 2000$ (for SD) and $S_D = S_{Di} \cdot N_c = 100 \cdot 29$ (for MD), as described in Section VIII-A. The mean average correct classification rates (mACCR) of our BoSURF-SD and BoSURF-MD approaches are 94.84% and 93.7% respectively. Although the overall performance of BoSURF-SD is superior to that of BoSURF-MD, we note that BoSURF-MD had better ACCR for some classes than BoSURF-SD. See for example: Toyota Camry, Nissan Sentra, Mitsubishi Zinger, Outlander, Lancer, Ford Liata, and Escape, in Tables IV and V.

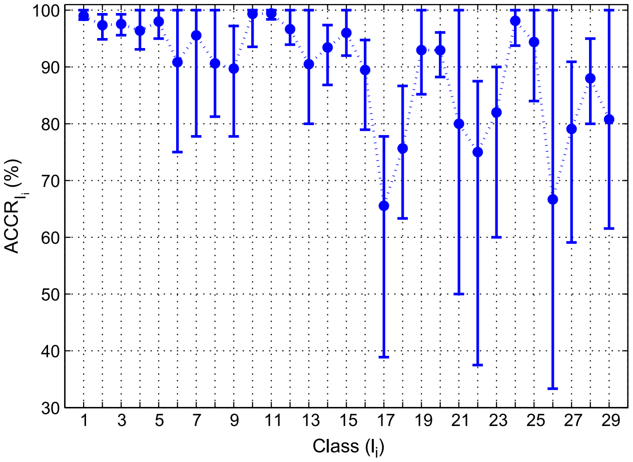
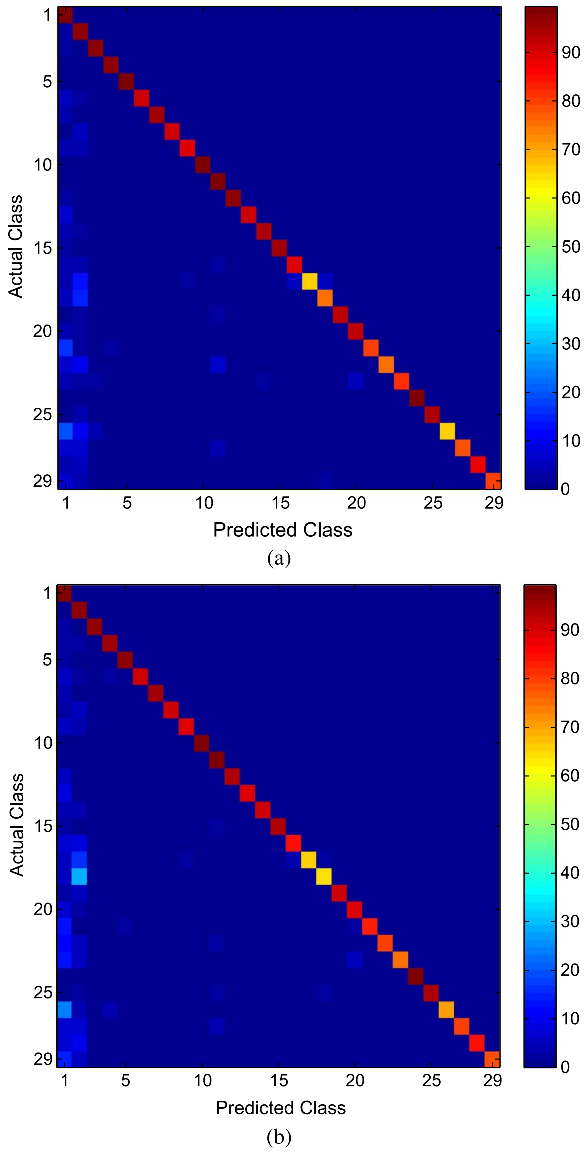


Fig. 14. Average, maximum, and minimum classwise ACCR_{c_i} for BoSURF-SD-based VMMR, run over the $N_D = 10$ different 80-20 data-set splits.

Contrary to our expectation that BoSURF-MD would perform better than BoSURF-SD, the results indicate otherwise. One reason could be the fixed size of individual sub-dictionaries of all classes, which could lead to many less discriminative and noisy features being selected as codewords in the overall dictionary. In future, we shall investigate dictionary pruning methods to build a more robust Modular Dictionary.

The average, minimum and maximum classwise ACCR_{c_i} of our SVM based BoSURF-SD approach, over the $N_D = 10$ different 80-20 training-testing dataset versions, are shown in Fig. 14. Each of the ten dataset versions is populated by randomly choosing 80% of the total images for training and the rest for testing. While most of the classes have high ACCR, some classes consistently performed badly even across the 10 different dataset versions. These include classes 17 (Nissan Sentra), 18 (Nissan Cefiro), 21 (Mitsubishi Zinger), 22 (Mitsubishi Outlander), 23 (Mitsubishi Savrin), 26 (Ford Liata), 27 (Ford Escape), 28 (Ford Mondeo), and 29 (Ford Tierra). In case of Ford-Liata (Class 26), there were only 3 images for testing, and 16 for training. The low numerical accuracy can be attributed to the lack of sufficient number of training and testing images.

It is noteworthy to mention that for some classes with low accuracy, e.g: Class 26, BoSURF-SD had a CCR of around 100% with at least one of the dataset versions (see Fig. 14). Similarly, for Class 27 (Ford Escape), although the accuracy (averaged over all N_D datasets) turned out to be 79.09%, for at least one of the 80-20 dataset splits, an ACCR_{c_i} of more than 90% was achieved. This clearly indicates that accuracy greatly depends on how images were distributed into training and testing. Whereas we assign images to training-testing sets randomly, the assignment of images into training and testing sets in the original NTOU-MMR Dataset [5] is not clear and seems to be biased (described in Section III).



(a)

(b)

Fig. 15. Confusion matrices for (a) BoSURF-SD and (b) BoSURF-MD, averaged over the ten 80-20 data sets.

We show the confusion matrix for the 29 make-model classes using BoSURF-SD and BoSURF-MD approaches in Fig. 15(a) and (b), respectively. One can observe that most of the inaccurate classifications are towards Classes 1 (Toyota Altis) and 2 (Toyota Camry). One of the major reasons for this effect could be a considerably greater amount of training data available for these classes (as we can see from Table III) which may have lead to a biased classifier. Note that the bias mentioned here is the one caused by large number of training samples for some classes, while a very small number of training samples for other classes. By building a more comprehensive dataset (as mentioned in our future work), we could have a similar number of images for each class. In this way, the classifier and dictionary can be expected to be unbiased. Under such imbalanced training data, even the dictionary could have become biased by retaining more codewords from classes 1 and 2 than from other classes.

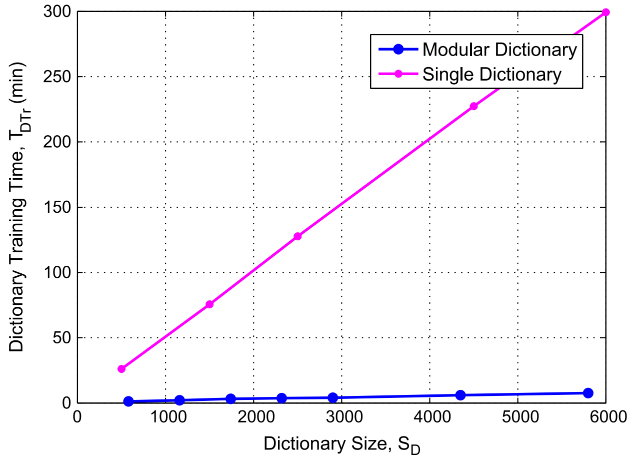


Fig. 16. Effect of dictionary size S_D on the dictionary training time T_{DTr}, for the single and modular dictionaries.

3) Dictionary Training Time: Although the accuracy and speed of BoSURF-MD VMMR is slightly less than compared to BoSURF-SD, the time required to build or re-build the MD, i.e. the *Dictionary Training Time* (T_{DTr}), is drastically less than compared to SD's T_{DTr}. One can observe in Fig. 16 that there is a huge difference in T_{DTr} for MD and SD. Unlike SD, the increase in dictionary size does not cause the T_{DTr} of MD to increase rapidly. The cost in time for training and re-training of MD is therefore significantly less than that of SD. In real-life scenarios, security personnel may be looking for different subsets of vehicle makes and models at different times. Therefore, a VMMR system should recognize only those makes and models, rather than all that are passing by. In such applications where re-building of dictionaries due to addition or removal of desired or undesired make-model classes is needed, BoSURF-MD would be a more efficient choice. However, in applications where the reconstruction of the dictionary is unnecessary, then BoSURF-SD stands as a better choice.

B. Performance of AB-SVM Based BoSURF-MMR

Motivated by the success of using AB to build an ensemble of classifiers in several works such as [24], we investigate whether the AB-based ensemble of multi-class SVM classifiers (AB-SVM), trained over BoSURF representations of different makes and models, could improve the performance of VMMR in comparison to the single classifier scheme of Section VII-A.

The classwise ACCR$_{c_i}$ and average number of correctly classified images for AB-SVM based BoSURF-SD and BoSURF-MD approaches are shown in Tables VI and VII, respectively. These results are obtained using ($N_{ss} = 15$, $S_{ss} = 500$) for BoSURF-SD, and ($N_{ss} = 15$, $S_{ss} = 1500$) for BoSURF-MD, as obtained in Section VIII-A. The average processing speed with these configurations was around 5 fps with SD, and around 3 fps with MD. The average accuracies turned out to be around 93.02% and 93.68% with SD and MD, respectively. It can be observed that BoSURF-MD with AB-SVM had very similar accuracy as BoSURF-MD with SVM, although processing speed was compromised. However,

TABLE VI
PERFORMANCE OF OUR BOSURF-MMR WITH SD AND AB-SVM

Make	Toyota					
Model	Altis	Camry	Vios	Wish	Yaris	Previa
ACCR(%)	98.58	96.40	96.54	93.97	98.00	90.00
Avg #Cor	236.6	131.1	131.3	54.5	58.8	10.8
Make	Toyota				Suzuki	Nissan
Model	Innova	Surf	Tercel	RAV4	Solio	Tiida
ACCR(%)	94.44	91.88	90.00	98.71	93.6	88.04
Avg #Cor	8.5	14.7	32.4	30.6	23.4	44.9
Make	Nissan					
Model	March	Livna	Teana	Sentra	Cefiro	Xtrail
ACCR(%)	88.68	95.20	86.32	57.22	63.67	90.00
Avg #Cor	33.7	47.6	16.4	10.3	19.1	24.3
Make	Mitsubishi				Honda	
Model	Zinger	Outlander	Savrin	Lancer	CRV	Civic
ACCR(%)	73.33	77.50	76.00	96.88	99.84	95.45
Avg #Cor	4.4	6.2	7.6	15.5	122.8	63.0
Make	Honda	Ford				Overall
Model	FIT	Liata	Escape	Mondeo	Tierra	
ACCR(%)	92.00	66.7	73.64	77.00	76.92	93.02
Avg #Cor	18.4	2.0	16.2	15.4	10.0	1210.5

TABLE VII
PERFORMANCE OF OUR BOSURF-MMR WITH MD AND AB-SVM

Make	Toyota					
Model	Altis	Camry	Vios	Wish	Yaris	Previa
ACCR(%)	99.08	97.65	96.32	94.66	98.17	89.17
Avg #Cor	237.8	132.8	131.0	54.9	58.9	10.7
Make	Toyota				Suzuki	Nissan
Model	Innova	Surf	Tercel	RAV4	Solio	Tiida
ACCR(%)	95.56	90.63	90.56	98.06	94.80	89.22
Avg #Cor	8.6	14.5	32.6	30.4	23.7	45.5
Make	Nissan					
Model	March	Livna	Teana	Sentra	Cefiro	Xtrail
ACCR(%)	89.74	95.60	86.32	65.56	60.33	90.00
Avg #Cor	34.1	47.8	16.4	11.8	18.1	24.3
Make	Mitsubishi				Honda	
Model	Zinger	Outlander	Savrin	Lancer	CRV	Civic
ACCR(%)	83.33	82.50	77.00	97.50	99.43	94.55
Avg #Cor	5.0	6.6	7.7	15.6	122.3	62.4
Make	Honda	Ford				Overall
Model	FIT	Liata	Escape	Mondeo	Tierra	
ACCR(%)	89.00	70.00	78.64	86.00	76.92	93.02
Avg #Cor	17.8	2.1	17.3	17.2	10.0	1210.5

TABLE VIII
PERFORMANCE SUMMARY OF OUR BOSURF-MMR APPROACHES

Approach	Classifier	mACCR (%)			Speed (fps)		
		Max	Avg.	Min	Max	Avg.	Min
BoSURF-SD	SVM	95.77	94.84	94.00	7.6	7.5	7.25
	AB-SVM	94.53	93.02	91.46	5.01	4.97	4.94
BoSURF-MD	SVM	94.38	93.7	92.3	7.25	6.99	6.51
	AB-SVM	94.3	93.68	92.76	2.83	2.81	2.78

the performance of BoSURF-SD with AB-SVM (in terms of accuracy and speed) is slightly reduced, compared to SVM-based BoSURF-SD and BoSURF-MD. The reduced performance of AB-SVM based BoSURF-SD could be attributed to the random selection of dictionary codewords to form feature subsets, without considering the importance (or discriminative cap-acity) of the selected codewords. This could have led to the selection of noisy or non-discriminative codewords. By incorporating the discriminative capacity of the codewords, or by increasing N_{ss} (as discussed in Section VIII-A), AB-SVM could perform better with BoSURF for VMMR.

A summary of mACCR and processing speeds of our BoSURF-based approaches for VMMR is given in Table VIII. Based on our findings, we recommend using a single multi-class SVM-based BoSURF-SD or BoSURF-MD for real-time VMMR systems, owing to their higher processing speeds and accuracies.

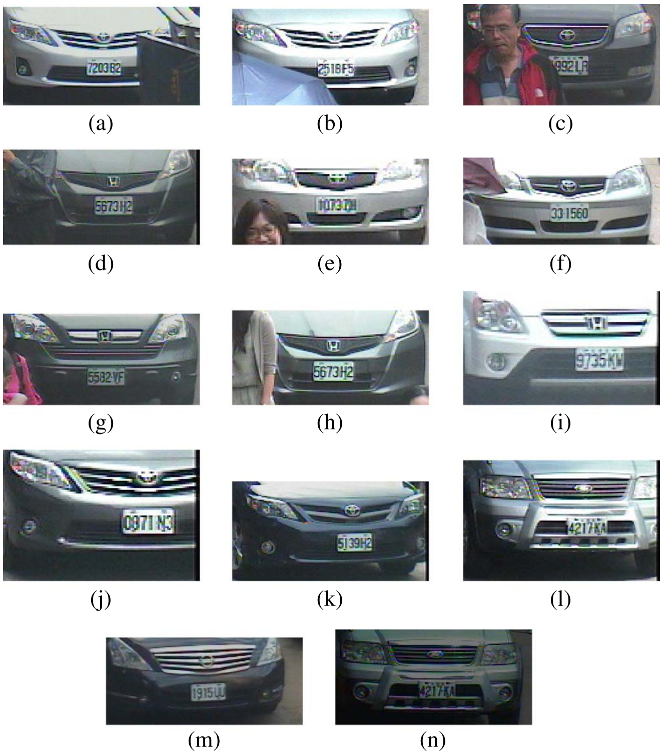


Fig. 17. Some challenging cases of vehicles under (a)–(h) occlusion, (i) and (j) partially out of the view of the camera, (k)–(m) nonfrontal views, or (m)–(n) under low lighting. The BoSURF-based VMMR approaches were successful in predicting the make–model class in the above cases.

C. Performance in Challenging Conditions

The BoSURF approaches for VMMR perform well even in challenging scenarios such as vehicles under occlusion, non-frontal views, and low lighting, as depicted in Fig. 17. The invariance to such challenging conditions could be attributed to the sparse nature of BoF-based global representations, in which the non-zero values are the aggregated votes of similar keypoint-based patches.

When the vehicle face is under occlusion, there can be two cases: (a) The occluding object has a relatively texture-less surface (e.g., a uni-color umbrella), or (b) the occluding object has a highly varying texture or appearance (e.g., a person). In the former case, there are little or no keypoint-based patches (due to scarcity or absence of corners), and hence it doesn't affect the overall BoF representation. In the latter case, the occluding object may also result in keypoint based patches (due to presence of corners). These occluding patches would cast votes to the dictionary codewords, thereby adding noise to the overall BoF representation. There could be two sub-cases in such a scenario: (a) The occluding patches are very widely scattered in feature space and hence cast scattered votes (noise per bin of the BoF histogram is very minimal), or (b) The occluding patches may be close in feature-space and hence add a considerable noise to the overall BoF histogram. In the former sub-case, since the noise is distributed, the overall BoF histogram's shape will not be affected considerably. However, in the latter sub-case, the overall BoF histogram's shape could be severely affected, leading to inaccurate predictions.

TABLE IX
PERFORMANCE COMPARISON OF BOSURF-MMR WITH OTHER WORKS

Method	mACCR (%)	Speed (fps)
SIFT-Features + Brute-force Matching [4]	90.52	2.19
Edges-based Features + Nearest Neighbors [3]	67.33	0.93
LNHS + Naive Bayes [2]	85.9	1.73
FID-SRC-HDC [15]	91.1	0.46
BoSURF-SD-(SVM)	**94.84**	**7.5**
BoSURF-SD-(AB-SVM)	93.02	4.97
BoSURF-MD-(SVM)	93.7	6.99
BoSURF-MD-(AB-SVM)	93.68	2.81

D. Comparisons With Related Works on NTOU-MMR Dataset

The proposed BoSURF approaches for VMMR presented in this paper outperform several related VMMR works, both in terms of processing speed and classification accuracy. A performance comparison of our work with results of other related works on the NTOU-MMR Dataset is presented in Table IX. Both of our BoSURF approaches (BoSURF-SD and BoSURF-MD) significantly outperform the works of [2]–[4]. The work in [4] employs a brute-force matching scheme of the local SIFT features, making it highly inefficient for real-time VMMR systems. The approach of [3] results in the worst performance, due to its reliance on edge pixels to build global representations of vehicle makes and models, which are prone to image noises and occlusions, and which lack discriminating power. The low accuracy in [2] indicates the inefficiency of Locally Normalized Harris Strengths (LNHS) for the VMMR problem.

More recently, a sparse representation scheme and a Hamming Distance-based classification for VMMR was proposed by Chen *et. al.* [15]. Considering the average accuracy and speed of their best performing scheme (referred to as FID-SRC-HDC) on the 29 make-model classes of the NTOU-MMR Dataset [6], we see in Table IX that our approaches outperforms research works cited in this Table. Based on the above comparisons, one can conclude that the BoSURF-VMMR approaches are superior in terms of both accuracy and processing speed.

X. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed and evaluated unexplored approaches for real-time automated vehicle make and model recognition (VMMR) based on Bag-of-SURF features. The major contributions of this work are as follows: (1) Two schemes for Dictionary Building are studied and evaluated to address the multiplicity and ambiguity problems of VMMR; (2) The optimal dictionary sizes for both dictionaries are recommended via experimental evaluations in the context of VMMR; (3) Two multi-class classification schemes are proposed and evaluated for accurate and efficient make-model prediction: (a) Single-SVM and (b) Attribute Bagging based Ensemble of SVMs (AB-SVM). The effectiveness and superiority of our approaches over the state-of-the-art works are validated using random training-testing splits of the NTOU-MMR Dataset [5], [6]. Thorough experimental evaluations have shown that our BoSURF-based VMMR approaches are highly suitable for real-time vehicle identification applications.

For future work, we plan to enhance BoSURF-VMMR approaches by exploring dictionary pruning methods. Necessitated by the lack of a standard publicly available benchmark dataset for VMMR works, we plan to build a comprehensive VMMR dataset that exhibits the real-world challenges in VMMR and includes a wider variety of colors, makes and models. We shall also explore developing a real-time on-device mobile VMMR system.

REFERENCES

[1] J. Sivic and A. Zisserman, “Video google: A text retrieval approach to object matching in videos,” in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, vol. 2, pp. 1470–1477.

[2] G. Pearce and N. Pears, “Automatic make and model recognition from frontal images of cars,” in *Proc. 8th IEEE Int. Conf. AVSS*, Aug. 2011, pp. 373–378.

[3] D. T. Munroe and M. G. Madden, “Multi-class and single-class classification approaches to vehicle model recognition from images,” in *Proc. 16th Irish Conf. Artif. Intell. Cogn. Sci.*, Sep. 2005, pp. 93–104.

[4] L. Dlagnekov, “Video-based car surveillance: License plate, make, and model recognition,” M.S. thesis, Dept. Comput. Sci. Eng., Univ. California, San Diego, San Diego, CA, USA, 2005.

[5] J.-W. Hsieh, L.-C. Chen, and D.-Y. Chen, “Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 6–20, Feb. 2014.

[6] NTOU-MMR Dataset. [Online]. Available: http://mmplab.cs.ntou.edu.tw/mmplab/MMR/MMR.html

[7] A. Mammeri, E.-H. Khiari, and A. Boukerche, “Road-sign text recognition architecture for intelligent transportation systems,” in *Proc. 80th IEEE VTC Fall*, Sep. 2014, pp. 1–5.

[8] S. Sivaraman and M. Trivedi, “Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013.

[9] D. Lowe, “Object recognition from local scale-invariant features,” in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, 1999, vol. 2, pp. 1150–1157.

[10] R. Baran, A. Glowacz, and A. Matiolanski, “The efficient real- and non-real-time make and model recognition of cars,” *Multimedia Tools Appl.*, vol. 74, no. 12, pp. 4269–4288, Jun. 2013.

[11] P. Badura and M. Skotnicka, “Automatic car make recognition in low-quality images,” in *Information Technologies in Biomedicine*, vol. 3, ser. Advances in Intelligent Systems and Computing, E. Pitka, J. Kawa, and W. Wieclawek, Eds. Springer-Verlag, 2014, vol. 283, pp. 235–246.

[12] M. Fraz, E. A. Edirisinghe, and M. S. Sarfraz, “Mid-level-representation based lexicon for vehicle make and model recognition,” in *Proc. 22nd ICPR*, Aug. 2014, pp. 393–398.

[13] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Speeded-up robust features (SURF),” *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[14] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2005, vol. 1, pp. 886–893.

[15] L.-C. Chen, J.-W. Hsieh, Y. Yan, and D.-Y. Chen, “Vehicle make and model recognition using sparse representation and symmetrical SURFs,” *Pattern Recognit.*, vol. 48, no. 6, pp. 1979–1998, Jun. 2015.

[16] H. He, Z. Shao, and J. Tan, “Recognition of car makes and models from a single traffic-camera image,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3182–3192, Dec. 2015.

[17] P. Salembier and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*, B. Manjunath, Ed. New York, NY, USA: Wiley, 2002.

[18] D. K. Park, Y. S. Jeon, and C. S. Won, “Efficient use of local edge histogram descriptor,” in *Proc. ACM Workshops MULTIMEDIA*, Los Angeles, CA, USA, 2000, pp. 51–54.

[19] V. Varjas and A. Tanacs, “Car recognition from frontal images in mobile environment,” in *Proc. 8th Int. Symp. ISPA*, Sep. 2013, pp. 819–823.

[20] L.-C. Chen, J.-W. Hsieh, Y. Yan, and D.-Y. Chen, “Vehicle make and model recognition using sparse representation and symmetrical SURFs,” in *Proc. IEEE 16th Int. Conf. Intell. Transp. Syst.*, Oct. 2013, pp. 1143–1148.

[21] D. Llorca, D. Colas, I. Daza, I. Parra, and M. Sotelo, “Vehicle model recognition using geometry and appearance of car emblems from rear view images,” in *Proc. 17th IEEE Int. Conf. Intell. Transp. Syst.*, Oct. 2014, pp. 3094–3099.

[22] The Industrial Liaison Program of VBIE. [Online]. Available: http://vbie.eic.nctu.edu.tw/en/introduction

[23] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. ECCV Workshop Stat. Learn. Comput. Vis.*, 2004, pp. 1–22.

[24] L. Nanni and A. Lumini, "Heterogeneous bag-of-features for object/scene recognition," *Appl. Soft Comput.*, vol. 13, no. 4, pp. 2171–2178, Apr. 2013.

[25] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *Proc. IEEE CVPR*, Jun. 2013, pp. 923–930.

[26] P. Pinto, A. Tome, and V. Santos, "Visual detection of vehicles using a bag-of-features approach," in *Proc. 13th Int. Conf. Auton. Robot Syst. Robotica*, Apr. 2013, pp. 1–4.

[27] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, vol. 2, pp. 2169–2178.

[28] S. Singh, S. Choudhury, K. Vishal, and C. Jawahar, "Currency recognition on mobile phones," in *Proc. 22nd ICPR*, Aug. 2014, pp. 2661–2666.

[29] L. Hazelhoff, I. Creusen, and P. de With, "Optimal performance-efficiency trade-off for bag of words classification of road signs," in *Proc. 22nd ICPR*, Aug. 2014, pp. 2996–3001.

[30] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proc. 6th ACM Int. CIVR*, 2007, pp. 494–501.

[31] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, Jun. 1998.

[32] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1995.

[33] R. Bryll, R. Gutierrez-Osuna, and F. Quek, "Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognit.*, vol. 36, no. 6, pp. 1291–1302, Jun. 2003.

[34] Support Vector Machines Implementation. [Online]. Available: http://docs.opencv.org/modules/ml/doc/support_vector_machines.html

[35] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, Apr. 2011.

[36] L. Hazelhoff, I. Creusen, D. van de Wouw, and P. H. N. de With, "Large-scale classification of traffic signs under real-world conditions," in *Proc. SPIE Multimedia Mobile Devices Multimedia Content Access, Algorithms Syst.*, 2012, vol. 8304, pp. 1–10.

[37] CCTV Camera Pros. [Online]. Available: http://www.cctvcamerapros.com/

[38] Security Camera Warehouse (SCW). [Online]. Available: https://www.security-camera-warehouse.com/ip-camera/

Abdul Jabbar Siddiqui received the M.A.Sc. degree from University of Ottawa, Ottawa, ON, Canada, in 2015. He is currently working toward the Ph.D. degree in electrical and computer engineering with University of Ottawa.

His research interests include intelligent surveillance systems, mobile computer vision, multimedia retrieval, intelligent transportation systems, advanced driver assistance systems, vehicular ad hoc networks, and smart city applications.

Abdelhamid Mammeri received the M.Sc. degree in electrical and computer engineering from Université Catholique de Louvain, Louvain-la-Neuve, Belgium, and the Ph.D. degree in electrical and computer engineering from Université de Sherbrooke, Sherbrooke, QC, Canada.

He is a Senior Research Associate with DIVA Strategic Research Network, University of Ottawa, Ottawa, ON, Canada. He has extensively published in top-tier international conferences and journals in areas of his research interests, which include intelligent transportation systems, advanced driver assistance systems, vehicular ad-hoc networks, and multimedia sensor networks.

Dr. Mammeri has served as a Technical Program Committee (TPC) Chair and a Track Chair for IEEE Vehicular Technology Conference in 2014, IEEE International Workshop on Performance and Management of Wireless and Mobile Networks in 2013 and 2015, and the IFIP International Conference on New Technologies, Mobility and Security in 2015. He has also served as a TPC member for several IEEE/ACM international conferences. He received the prestigious Fonds de recherche du Quebéc—Nature et technologies (FQRNT) Quebec PostDoctoral Scholarship Award in 2012.

Azzedine Boukerche (F'15) held a faculty position with University of North Texas, Denton, TX, USA. He worked as a Senior Scientist with the Simulation Sciences Division, Metron Corporation, San Diego, CA, USA. He is currently a Full Professor and holds the Senior Canada Research Chair Tier 1 position with University of Ottawa, Ottawa, ON, Canada. He is also the Scientific Director of the Natural Sciences and Engineering Research Council (NSERC) DIVA Strategic Research Network and NSERC-CREATE TRANSIT Network and the Director of PARADISE Research Laboratory with University of Ottawa. He spent a year at the NASA Jet Propulsion Laboratory (JPL), California Institute of Technology, Pasadena, CA, USA, where he contributed to a project on the specification and verification of the software used to control interplanetary spacecraft operated by NASA JPL.

Dr. Boukerche has served as a Steering Committee Chair for several IEEE and Association for Computing Machinery (ACM) international conferences. He currently serves as an Associate Editor for several IEEE Transactions and ACM journals. He received the Ontario Distinguished Researcher Award, the Premier of Ontario Research Excellence Award, the G. S. Glinski Award for Excellence in Research, the IEEE Computer Society Golden Core Award, the IEEE Computer Society (CS) Meritorious Award, the University of Ottawa Award for Excellence in Research, the IEEE Canada Gotlieb Medal Award, the IEEE CS Technical Committee on Parallel Processing Leaderships Award, and the IEEE Communication Society AHSN Leaderships Award. He is a Fellow of the Engineering Institute of Canada, the Canadian Academy of Engineering, and the American Association for the Advancement of Science.