Image Representations on a Budget: Traffic Scene Classification in a Restricted Bandwidth Scenario

Ivan Sikirić[1], Karla Brkić[2], Josip Krapac[2] and Siniša Šegvić[2]

Abstract— **Modern fleet management systems typically monitor the status of hundreds of vehicles by relying on GPS and other simple sensors. Such systems experience significant problems in cases of GPS glitches as well as in areas without GPS coverage. Additionally, when the tracked vehicle is stationary, they cannot discriminate between traffic jams, service stations, parking lots, serious accidents and other interesting scenarios. We propose to alleviate these problems by augmenting the GPS information with a short descriptor of an image captured by an on-board camera. The descriptor allows the server to recognize various scene types by image classification and to subsequently implement suitable business policies. Due to restricted bandwidth we focus on finding a compact image representation that would still allow reliable classification. We therefore consider several state-of-the-art descriptors under tight representation budgets of 512, 256, 128 and 64 components, and evaluate classification performance on a novel image dataset specifically crafted for fleet management applications. Experimental results indicate fair performance even with very short descriptor sizes and encourage further research in the field.**

I. Introduction

The goal of this paper is to develop a visual scene classification system that can be used to improve current fleet management systems. Fleet management systems consist of one central server to which hundreds of clients (tracking devices inside vehicles) report their status in regular intervals, typically over GPRS. The status of a tracked vehicle usually includes its position, speed and bearing, which are obtained via GPS. It can also include other sensor readings, such as fuel level or temperature of cargo hold, and state of any additional vehicle equipment, such as taxi meter, ambulance siren or snowplow. The server accumulates and processes this data and presents it to a human operator in a meaningful way, e.g. it generates monthly reports, reconstructs routes taken, raises alarms in real time in case of unexpected behavior. This helps to ensure the safety of the drivers, vehicles and the cargo. It also enables monitoring proper usage of the company resources and ultimately minimizes the total expenses.

Our first contribution is the augmentation of the vehicle status with visual cues extracted from images captured by an on-board camera. The server could use these cues to infer the properties of the vehicle's surroundings, which would help it in further decision making. For example, the server could infer the location of the vehicle (e.g. open road, tunnel, gas station), or cause of stopping (e.g. congestion, traffic lights, road works). Knowing the type of location would aid route reconstruction in cases of missing or imprecise GPS data. Loss of GPS precision usually occurs under or near tall objects, and is in some cases undetectable without additional cues. Detecting such scenarios using visual data would be very beneficial, especially in systems that offer real-time tracking of transported valuables.

We consider detecting interesting traffic scenarios by means of image classification. One particular approach to achieve that would be to perform the whole classification on the clients and report the result to the server. However, this would require a software update on every client every time a class is added or removed, or when a refined classifier is obtained on a larger training dataset. We therefore propose a more flexible approach in which clients send visual cues to the server that then performs the classification. The size of a client status in current systems is around 30 bytes. Adding an entire image taken by a camera to every status report would raise the data transfer by several orders of magnitude. Even though GPRS bandwidth is more than adequate to accommodate for this, there are still reasons to keep the data transfer as low as possible. Firstly, mobile data transfer costs money, especially in cases of international roaming. Even if we ignore the possibility of international travel, the data plan for clients is not necessarily flat-rate. It is usually possible to get better deals with limited data plans, especially if hundreds of devices are involved. Secondly, if we assume that an average client sends one status per minute, then it follows that a fleet management server requires about 1.5 GB of storage space per month for 1000 clients. Raising this by several orders of magnitude would require drastic changes to server hardware and software. Thirdly, the clients are equipped with a limited amount of flash memory to buffer all the accumulated data in cases when there is no GPRS connectivity signal, or the server is unavailable for other reasons. The capacity of this storage is typically not large enough to keep many images. Our solution to these problems is calculating the descriptor of the image on the client itself, before transmitting it to the server for further analysis. The image descriptor should be as short as possible while still enabling good separation between classes. We call this approach *image representation on a budget*.

In this paper we analyze the performance of several image descriptors: GIST, bag-of-words (BoW), Locality-

This work has been financially supported by the Research Centre for Advanced Cooperative Systems ACROSS (EU FP7 #285939).

This work has been supported by the project VISTA - Computer Vision Innovations for Safe Traffic, IPA2007/HR/16IPO/001-040514 which is co-financed by the European Union from the European Regional Development Fund.

[1]Ivan Sikirić is with Mireo d.d., Zagreb, Croatia ivan.sikiric@mireo.hr

[2]Authors are with Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia name.surname@fer.hr

constrained Linear Coding (LLC) and Spatial Fisher vectors (SFV), paired with classifiers Random Forest (RF) and Support Vector Machine (SVM). When considering image descriptors, we are not only focusing on finding the one with the best classification performance (regardless of feature vector size), but are also comparing their performance with imposed restrictions on feature vector length. The GIST image descriptor is especially interesting, as it is not vocabulary-based. It is rarely necessary to change the visual vocabulary used by a descriptor, but if the need ever arises, it would have to be updated on every client. This could be expensive, so it would be best to avoid this problem if possible.

Our second contribution is introduction of a new dataset, called the FM2 dataset, containing 6237 traffic scenes suitable for fleet management purposes and associated labels. This is an extended version of the FM1 dataset, introduced in [1]. Using the FM2 dataset we perform experimental evaluation of the proposed method and report detailed results.

The remainder of the paper is organized as follows: In the next section we give an overview of previous related work. We then describe the FM2 dataset in detail, and define the classification problem. This is followed by a brief description of the used descriptors and classifiers. After that we describe our experimental framework in detail and present the results. We conclude the paper by giving an overview of contributions and discussing some interesting future directions.

II. Related work

Active computer vision research related to image/scene classification mainly focuses on recognizing images of a large number of diverse classes [2]. It is driven by benchmark datasets such as Pascal VOC dataset (20 classes), Caltech 101 (101 classes), LabelMe etc. Image classes in these datasets range from people and animals to potted plants and other common household objects, appearing in more or less cluttered environments.

Current approaches to generic image classification can be divided into two categories [3]: low-level approaches and semantic approaches. Low-level approaches aim to merely reduce the dimensionality of the image prior to classification by representing it with low-level features, either globally or in local sub-blocks. In contrast, semantic approaches additionally add a level of understanding of *what* is in the image. According to Bosch et al. [3], there are three subtypes of semantic approaches: (i) methods based on semantic objects, where object detectors are employed to classify the image [4], (ii) methods based on local semantic concepts, such as the bag-of-words (BoW) approach [5], [6], [7], where meaningful features are discovered in local structures of the training images, and (iii) methods based on semantic properties, such as the GIST descriptor [8], [9], that measure a set of semantic properties of an image, e.g. naturalness or openness.

In this paper, we focus on methods from subtypes (ii) and (iii), i.e. methods based on local semantic concepts and methods based on semantic properties. Namely, we apply the bag-of-words approach [5], [6], [7], its derivative that uses locality-constrained linear coding (LLC) [10], spatial Fisher vectors (SFV) [11] and GIST descriptors [8], [9] to the problem of traffic scene classification. Individual methods that we use are described in detail in Section IV. Given the lack of easily distinguishable objects in some traffic scene categories of interest (e.g. open road, tunnel exit), we do not study methods based on semantic objects.

The volume of work focused on classifying traffic scenes is considerably smaller than generic image classification research. Existing approaches are mainly specifically tailored to traffic scenes, with few works that assess the performance of a general-purpose method on the problem [1]. For instance, Ess et al. [12] propose a segmentation-based approach for urban scene understanding, where pre-trained classifiers are used to label segmented regions. Each segmented region is assigned one of thirteen labels (e.g. car, street etc.) using a set of thirteen AdaBoost classifiers in a one-vs-all setup. Based on the segmentation information, three sets of features that capture discriminative properties of a road scene are extracted. These features are then fed to a classifier that distinguishes between different traffic scene classes.

Tang and Breckon [13] suggest analyzing three predefined regions of interest in a traffic scene image: (i) a rectangular region near the center of the image, (ii) a tall rectangular region on the left side of the image and (iii) a wide rectangular region at the bottom of the image. Each of the three regions of interest is represented by a predefined set of color, edge and texture-based features, including e.g. various components of RGB, HSV and YCrCb color spaces, gray-level co-occurrence matrix statistics and Gabor filters. The rationale is that specific features will respond to specific structures that are expected to occur within the predefined regions of the traffic scene image (e.g. road, or road edge). These features form the basis of feature vectors that are fed to a classifier (artificial neural networks and k-nearest neighbors are considered). A new dataset with four classes is introduced: motorway, offroad, trunkroad and urban road.

Mioulet et al. [14] build on the ideas of Tang and Breckon [13], retaining the three predefined regions of interest, but representing them with Gabor features only, using dedicated hardware. An image descriptor is obtained by constructing a histogram of Gabor filter responses and concatenating them over all three regions of interest. A Random Forest classifier is used.

In the application scenario considered in this paper, there is a constraint both on the descriptor size and on the processing time required to build the image descriptor. We therefore do not consider processing-intensive approaches, such as the segmentation-based approach of Ess et al. [12]. Rather, we focus on evaluating the performance of state-of-the-art general purpose methods for image classification that are fast to compute, with a special emphasis on the influence of feature vector length on the final performance. There is some research on short representations for image classification (e.g. a recently proposed, processing-intensive

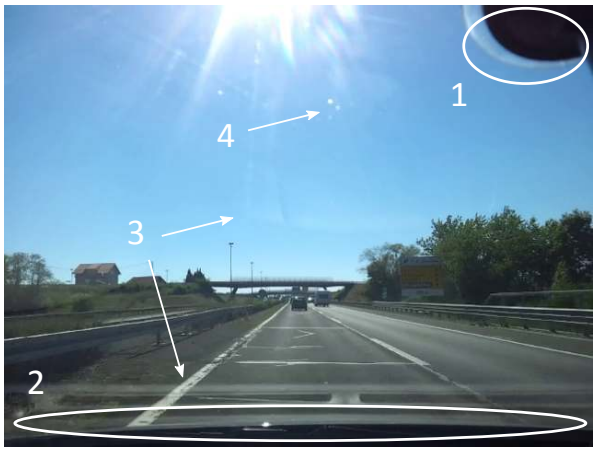

Fig. 1: A typical traffic scene. Note the camera mount (1), the dashboard (2), reflections of car interior (3) and a speck of dirt (4)

descriptor PiCoDes [15]), but to our knowledge there is no existing evaluation of the performance of general purpose image classification methods for traffic scene classification, with an emphasis on descriptor length. A reader interested in a detailed comparison of different variants of the standard bag-of-words approach is referred to the work of Chatfield et al. [16].

III. The FM2 Dataset

Since we want to run the experiments on traffic scenes and scenarios relevant to fleet management systems, we introduce a novel dataset of labeled traffic scenes, called the FM2 dataset[1]. It is an extension of our previously introduced dataset FM1 [1], and contains 6237 labeled images of traffic scenes as seen from the drivers perspective. The images were extracted from videos of several drives on European roads, obtained using a camera mounted inside a personal vehicle. The videos were recorded in mp4 format, with resolution 640×480, at 30 frames per second, using a built-in camera of Samsung Galaxy SII smartphone. All drives were made during daytime, and the largest percentage of the footage was recorded on highways. We avoided driving during moderate and heavy rain, to eliminate the possible windshield wipers occlusions. The position and orientation of the camera were changed only slightly between videos.

A typical frame of a video is shown in Figure 1. Some parts of the image are not parts of the traffic scene, but rather the interior of a vehicle, such as camera mount visible in the upper right corner, the dashboard in the bottom part, and occasional reflections of car interior visible on the windshield. The windshield itself can be dirty, and various artifacts can appear on it depending on the position of the sun.

A total of 8 classes were chosen for our experiments: highway, road, tunnel, tunnel exit, settlement, overpass, toll

[1] http://www.zemris.fer.hr/~ssegvic/datasets/unizg-fer-fm2.zip

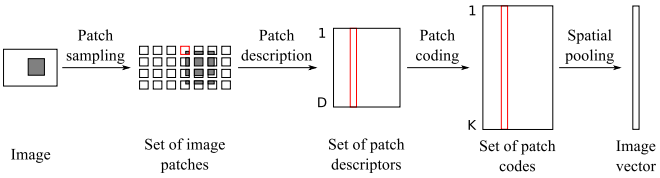

Fig. 2: The bag-of-words pipeline.

booth and dense traffic. The distribution of the classes in the dataset and their descriptions are listed in Table I. The highway, road, settlement and tunnel classes describe the general location of the vehicle, while other classes describe events considered to be interesting for fleet management purposes. The separation of highways and slower roads is useful for fleet management purposes because it will aid route reconstruction when GPS alone cannot be used to resolve ambiguity between a highway and a slower road that runs parallel to it. As was discussed in the introduction, we are very interested in detecting the environments in which the loss of GPS signal precision is likely to occur, or the vehicle is likely to stop or drive slowly. The tunnel is an environment in which a loss of GPS signal is almost certain, while toll booth and overpass are only indicating possible loss of GPS precision. The loss of precision is also more likely to occur in settlements (proximity to tall buildings), than on an open road. Additionally, recognizing a toll booth will improve the quality of cost-analysis reports provided by some fleet management systems. Recognizing a dense traffic scenario is useful for explaining the current driving pattern. Also, heavy occlusions of scene by other vehicles in dense traffic can completely eliminate any other useful visual cues. Finally, the tunnel exit class was separated because it is a signal to the fleet management server that the restoration of GPS signal can be expected soon, and because the camera reaction to the sudden increase in sunlight is very slow, which results in extremely bright images, so other visual cues are difficult to detect (similar problem is not encountered during tunnel entry). As we collect more footage, we plan to add other interesting classes, such as ferries, dirt roads, towing, etc.

IV. Descriptors and Classifiers

In this section we give a brief overview of image representations that we consider in our experiments. First we focus on representations based on bag-of-words (BoW) framework, as they are state-of-art for scene classification. Then we give an overview of methods for coding the global spatial layout of an image, since spatial layout is an important cue for scene classification. Finally, we give an overview of the GIST image representation [8], because it yields competitive classification performance on a small representation budget [1].

Figure 2 describes the BoW pipeline. An image is first represented by a set of patches and each patch is mapped to a patch descriptor. The mapping is constructed to achieve invariance of the descriptor with respect to local geometric

(a) highway (b) road (c) tunnel (d) exit

(e) settlement (f) overpass (g) booth (h) traffic

Fig. 3: Examples of classes from the FM2 dataset

TABLE I: Classes from the FM2 dataset

class label	scene description	number of occurrences
highway	an open highway	4337
road	an open non-highway road	516
tunnel	in a tunnel, or directly in front of it, but not at the tunnel exit	601
exit	directly at the tunnel exit (extremely bright image)	64
settlement	in a settlement (e.g. visible buildings)	464
overpass	in front of, or under an overpass (the overpass is dominant in the scene)	86
booth	directly in front of, or at the toll booth	75
traffic	many vehicles are visible in the scene, or completely obstruct the view	94

and photometric transformations of the input patch, so the distance between patch descriptors reflects visual similarity between patches. At this point the image is represented by a set of D-dimensional patch descriptors, e.g. SIFT descriptors [17]. Image classification depends on the definition of similarity function between images. Therefore, it is crucial to define a good image similarity measure that can be evaluated efficiently. To that end, BoW methods first map the set of patch descriptors into an image vector, e.g. BoW histogram [18], and then determine similarity between images via similarity between image vectors. The image vector is obtained in two steps. First, the patch descriptors are coded into discrete "visual words". Then, the vector is constructed by spatial pooling of the descriptor codes.

BoW histogram [18] was the first BoW representation used for image classification. BoW histogram codes patch descriptors via visual vocabulary. The visual vocabulary is obtained by clustering a set of randomly sampled patch descriptors from the training set into K clusters. The visual words correspond to cluster centers. Each descriptor is then coded by an index of the closest visual word. This encoding can be represented by a K-dimensional vector that has zeros everywhere except at the position corresponding to the assigned visual word. Finally, an image vector is obtained by summing the patch code vectors. If the image vector is normalized to L_1 norm the image vector can be interpreted as a histogram, i.e. a distribution over visual word indices. In this case similarity between probability distributions is used to evaluate similarity between images, e.g. χ^2 distance.

The visual vocabulary can be viewed as a generative model of patches in the image, and k-means clustering as a way of learning the parameters that maximize the likelihood of training patches. However, it is a very poor generative model since it does not model the distribution of patch descriptors assigned to the same visual word. A more expressive generative model would be a Gaussian mixture model (GMM). The parameters of the GMM (visual word weights, mixture means and covariance matrices) are learned using the expectation-maximization algorithm. The Fisher vector approach [19] codes each descriptor with the gradients of the descriptor with respect to GMM parameters. As in BoW histogram case, the image vector is also obtained as a sum of the patch codes.

Locality-constrained linear coding [20] uses the same kind of visual vocabulary as BoW histogram. However, it codes the patch descriptor by coefficients of a linear combination that minimizes mean square error of descriptor reconstruction from its nearest visual words. In this case the image vector is obtained from patch codes by max-pooling: the value corresponding to a visual word is the maximum over all patch codes for the image.

These image representations are invariant to the layout of the patches in the image: the same set of patches in different spatial layouts results in the same image representation. This is a drawback for scene classification, since spatial layout is a powerful cue for discrimination of scenes.

To code a spatial layout, Lazebnik et al. [21] used a

quad-tree to define a spatial grid. Each node in a quad-tree corresponds to an image region. Descriptor codes are first pooled for each region, and then the region representations are concatenated to obtain the image representation. However, to encode fine differences between spatial layouts, a large number of regions is necessary. As representation grows linearly with number of regions, this presents an important drawback in cases where the image representation is limited by a budget. Spatial Fisher vector (SFV) [11] addresses this problem by using the Fisher vector principle to encode the spatial layout. All visual words share the same generative model of patch positions (single Gaussian per visual word). The gradients of patch positions with respect to this generative model are called spatial Fisher vectors, and are used to code spatial layout. Note that although the generative model is the same for all visual words, the spatial Fisher vectors are gradients of image data with respect to this model, and therefore different for each image and each visual word. This encoding of spatial layout is much more compact than when region descriptors are concatenated: in SFV for each visual word additional 4 numbers are added (two for position mean and two for position variance), while a quad-tree adds C image regions per visual word. As a result, representation that uses 8 regions is two times greater than the representation that uses the same model to encode appearance, but uses SFV for spatial coding.

The GIST descriptor [8] has been developed specifically for scene recognition. It is a very low dimensional representation of the scene that captures perceptual features meaningful to a human observer, such as naturalness, openness, roughness, etc. It is calculated by first subdividing the image into 16 regions (a 4×4 grid), and then concatenating the average energies of 32 orientation filter responses (8 orientations on 4 scales) for each cell. Therefore the length of the feature vector is $16 \cdot 32 = 512$.

Once the image is embedded into a vector space, a classifier is learned for each class in a one-vs-all fashion. We evaluate two commonly used classifiers, SVM [22] and Random Forest [23]. The SVM classifier minimizes regularized hinge loss on training data, while Random Forest maximizes mutual information between the data in the leaves of the forest and class labels.

V. Experiments

In order to evaluate the performance of the considered descriptors in the *image representation on a budget* scenario, we tuned the parameters of each descriptor to obtain image feature vectors of varying lengths. Descriptor performance was evaluated on the FM2 dataset using SVM and Random Forest classifiers. Mean average precision (mAP, mean of per-class average precision) was used as a performance measure. The exact process of obtaining the image feature vectors of desired lengths depends on the descriptor used, and will now be described in more detail.

For BoW and LLC methods we have extracted patches from a dense grid with a step of two pixels, while for SVF we used a step of five pixels. BoW histograms and LLC use patches of size 16, 24, 32 and 40, while SFV uses patches of size 40, 60, 80 and 100. The patches were described using the SIFT descriptor implementation from the VLFeat library [24]. For BoW histograms and LLC we constructed a vocabulary of K visual words using k-means clustering. We divided the image into $C = 8$ regions: whole image, 4 quadrants corresponding to the first level of quad-tree and 3 horizontal strips. Therefore, BoW histograms and LLC image vectors have a size of KC. Our baseline results for these representations are obtained using $K = 512$. To evaluate these representations we used the implementation provided by Chatfield et al. [16].

For SFV we learned GMM using the EM algorithm. The size of SFV image vector is $K(1+2d+5C)$ where d is the size of SIFT descriptor projected to PCA subspace. We have used diagonal approximation of covariance matrix, both for local descriptors and position features. Our baseline results for SFV are obtained with $d = 80$, $K = 16$ and $C = 1$. The evaluation was performed using the code of Krapac et al. [11].

For the GIST descriptor we used 32 orientation filters (8 orientations at 4 scales) over a 4×4 grid to get the baseline image vector of length 512. We used the MATLAB implementation provided by the authors.

We have varied parameters to obtain image vectors of lengths 512, 256, 128 and 64. For BoW histograms and LLC we kept $C = 8$ and changed the number of visual words K accordingly. For SFV we always used $C = 1$, and therefore the gradient with respect to weight for spatial component is subsumed in the gradient for the weight of visual word. We have discarded these to obtain desired lengths of image vectors, resulting in image vectors of size $K(2d + 4)$. We have fixed $d = 6$ and varied K accordingly. This choice was motivated by observation that first eigenvalues capture majority of variability of SIFT descriptor, therefore the majority of energy is conserved in low-dimensional linear subspace obtained by PCA. For GIST we reduced the number of orientations per scale from 8 to 4 to obtain vectors of length 256. We reduced the size of grid to 2×2 to obtain a 128-dimensional image vector. Finally, we combined both of these modifications to get our most compact, 64-dimensional image vector.

The proposed setup results in a total of 19 different image vectors, five for every descriptor except GIST, whose baseline is already of length 512. The classification was performed using SVM and Random Forest (RF) classifiers. For SVM classification we used the LibSVM library [25], and for RF classification we used a MATLAB interface to Liaw et al.'s C code [26]. For LLC and SFV descriptors we used linear SVM, for BoW histograms we used Hellinger kernel, and for GIST we used RBF χ^2 kernel.

The dataset was divided into three subsets: train, validation and test, using a 25/25/50 split of instances for each class. The validation subset was used to determine the best parameters for classifiers trained on train subset, using mean average precision (mAP) as a performance measure. The optimized parameters were then used to train the classifier on train and

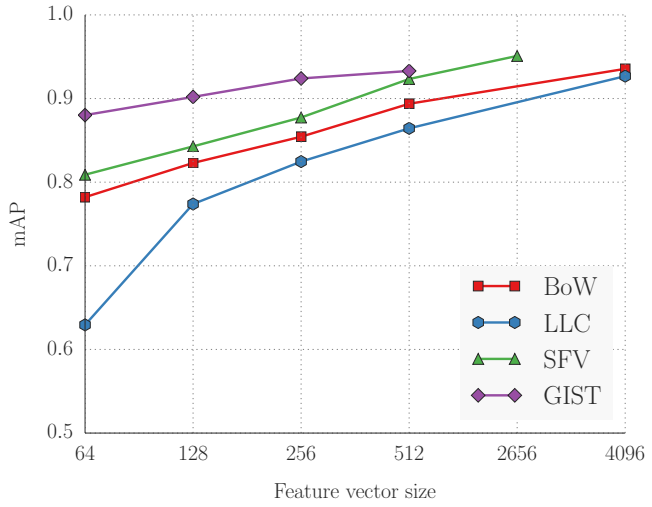

Fig. 4: Performance of descriptors using the SVM classifier (mAP)

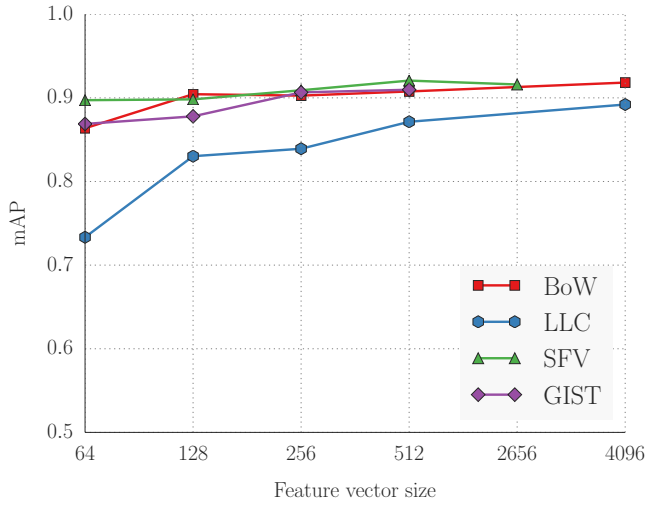

Fig. 5: Performance of descriptors using the RF classifier (mAP)

validation subsets, and final performance was evaluated on the test subset.

The results for the SVM classifier are shown in Table II, while the results for the Random Forest classifier are shown in Table III. SVM outperforms RF for every baseline descriptor, and also for every feature vector length of GIST descriptor. As feature vectors for the BoW, LLC and SFV descriptors become shorter, the performance of the SVM drops rapidly. RF, on the other hand, seems to be more resistant to the reduction of feature vector length, and outperforms SVM for shorter variants of the BoW, LLC and SFV descriptors. The only non-baseline variant of the BoW, LLC and SFV descriptors for which SVM shows better performance is SFV of length 512. The best classification performance, without considering representation budget, is achieved with the combination of the SFV descriptor and the SVM classifier. SFV would be considered to be the best descriptor for our use case if the length of the feature vector were not a factor.

For the remainder of this discussion we will consider only short descriptors, i.e. the descriptors with feature vector lengths of 512 or lower. Comparison of the descriptors with respect to feature vector length is shown in the Figure 4 for the SVM and in the Figure 5 for the RF classifier. We can see that for the SVM case the GIST descriptor is a clear winner. It shows the best performance for every feature vector length below or equal to 512. The GIST descriptor of length 64 is outperformed only by SFV and BoW descriptors of length 512, as all other descriptors combined with SVM achieve mAP below 88%. If we take a look at the RF performance, we can see that every variant of SFV descriptor and every variant of BoW except of length 64 also outperforms GIST of length 64 with SVM. The LLC descriptor shows poor performance regardless of the classifier, especially if we consider the overpass class, where the average precision drops very rapidly as we shorten the vector length. It is interesting to note that SFV descriptor of length 512 performs better than the baseline if RF classifier is used (92.06% to 91.60%). The SFV of length 512 is obtained using 32 appearance components, while baseline SFV is using only 16, so it seems that this parameter is very important to the RF classifier.

Figure 6 shows some examples of images that were hard to classify. We only took the baseline descriptors into account, paired with both the SVM and the RF classifier, which makes a total of 8 combinations. The highway example was misclassified in 6 different cases, while all other examples were misclassified in all 8 cases. The highway example is in most cases confused with road, and vice versa. This is not surprising, as both of these classes are similar in appearance. The tunnel example is in most cases confused with a highway because the highway is clearly visible, and the tunnel in question is very different in appearance to most other tunnels in the dataset. The next example is a scene labeled as a tunnel exit, while being further from the actual exit than the tunnel scene in the previous example. It is of no surprise that it has been classified as a tunnel in most cases. The traffic scene example, which is in most cases misclassified as a settlement scene, contains only one vehicle. While that vehicle is obstructing a large part of scene, many elements of a settlement scene are still clearly visible. This example suggests that we should allow assigning multiple labels to a single image. Other examples demonstrate that there are some scenes that are obvious and easy for a human observer to classify, while our method still fails to do the same. While other examples of misclassifications could perhaps be resolved by more sophisticated labeling of dataset images, these errors can only be reduced by expanding the dataset, improving the methods or optimizing their parameters.

VI. Conclusion and Future Work

Our experiments demonstrate that it is certainly possible to achieve good classification performance of traffic scene images, even if we impose great restrictions on the length

TABLE II: Per-class average precision (percentage), SVM classifier

descriptor	highway	road	tunnel	exit	settlement	overpass	booth	traffic	mean
BoW 4096	99.86	92.67	99.35	95.93	97.02	84.99	96.80	81.77	93.55
BoW 512	99.55	80.98	99.47	93.99	92.17	75.88	93.75	79.21	89.37
BoW 256	99.39	68.04	99.08	88.27	89.36	69.72	93.52	76.10	85.43
BoW 128	98.81	63.77	98.61	87.25	86.02	68.36	89.77	65.78	82.30
BoW 64	97.57	59.66	97.44	78.57	77.88	66.22	87.26	61.04	78.21
LLC 4096	99.86	94.69	99.52	94.19	96.67	80.29	96.32	79.87	92.68
LLC 512	99.54	82.17	99.22	92.13	90.73	67.71	89.67	70.31	86.43
LLC 256	99.34	70.08	98.74	87.17	87.89	61.45	86.16	68.85	82.46
LLC 128	98.61	59.24	97.40	82.80	83.79	52.50	83.73	61.04	77.39
LLC 64	95.54	46.88	93.95	57.63	66.55	30.88	62.73	49.31	62.94
SFV 2656	99.94	96.30	99.87	95.79	97.03	92.51	90.74	88.57	95.09
SFV 512	99.82	91.26	99.40	95.47	93.43	89.91	87.50	81.90	92.34
SFV 256	99.64	88.68	99.46	87.19	93.02	81.86	82.45	69.57	87.73
SFV 128	99.19	77.74	98.83	82.68	86.29	80.96	78.74	69.79	84.28
SFV 64	98.96	65.10	98.53	88.64	83.05	76.45	68.39	67.99	80.89
GIST 512	99.84	93.72	99.76	98.11	97.05	83.31	94.40	80.21	93.30
GIST 256	99.77	91.79	99.79	98.14	96.25	83.41	94.18	75.87	92.40
GIST 128	99.66	90.28	99.51	96.61	94.86	82.15	89.87	68.60	90.19
GIST 64	99.45	84.20	99.31	92.02	92.88	83.55	88.86	63.88	88.02

TABLE III: Per-class average precision (percentage), Random Forest classifier

descriptor	highway	road	tunnel	exit	settlement	overpass	booth	traffic	mean
BoW 4096	99.85	94.58	99.62	89.38	95.53	77.60	95.32	82.74	91.83
BoW 512	99.80	93.48	99.67	89.95	96.10	70.32	94.52	82.29	90.77
BoW 256	99.78	92.79	99.60	89.78	95.84	72.17	95.05	77.12	90.27
BoW 128	99.69	90.64	99.49	87.49	94.69	78.47	95.49	77.54	90.44
BoW 64	99.39	84.92	99.18	81.13	92.41	73.13	93.69	67.07	86.36
LLC 4096	99.72	91.91	99.63	89.82	93.26	71.12	94.43	73.73	89.20
LLC 512	99.62	89.24	99.48	89.35	91.08	63.74	90.58	74.02	87.14
LLC 256	99.34	84.53	99.11	86.72	90.45	55.23	91.35	64.59	83.92
LLC 128	99.36	83.17	98.54	85.41	90.29	54.11	89.99	63.32	83.02
LLC 64	98.21	73.04	96.05	69.35	80.84	35.91	73.39	59.86	73.33
SFV 2656	99.83	91.27	99.56	85.37	95.64	89.66	91.98	79.48	91.60
SFV 512	99.81	92.76	99.39	90.37	95.44	85.13	94.25	79.36	92.06
SFV 256	99.77	91.59	99.20	89.68	94.87	85.40	90.55	76.20	90.91
SFV 128	99.76	90.78	99.10	87.17	95.46	82.86	90.22	73.22	89.82
SFV 64	99.69	89.07	99.22	86.93	93.70	87.48	91.39	70.29	89.72
GIST 512	99.57	88.33	99.53	94.76	94.17	79.56	91.61	80.23	90.97
GIST 256	99.54	88.01	99.57	93.59	94.14	81.00	91.73	77.67	90.66
GIST 128	99.16	81.63	99.26	94.73	89.59	78.78	90.16	69.00	87.79
GIST 64	98.96	78.89	99.21	94.39	88.31	82.88	85.88	66.53	86.88

of the image descriptor. There are many good combinations of existing descriptors and classifiers that are suitable for this task. Our experiments indicate that the spatial Fisher vector and SVM classifier provide the best performance in case of unrestricted bandwidth, while other descriptors follow closely. If, however, we need to represent images on a budget, then the GIST descriptor used with SVM classifier shows better performance. As a bonus, the GIST descriptor is not based on a visual vocabulary, so we do not have to worry about updating the visual vocabularies of fleet management clients. SVM classifier would perform well for feature vectors of length 512, but for shorter vectors of feature based descriptors the Random Forest classifier would perform better.

It is possible to obtain feature vectors of the same length by tuning the descriptor parameters in many different ways. We have only tested one such setup, chosen by some educated guessing and quick experiments. It is possible that some other combinations of parameters would deliver better performance, depending on the classifier used. Of course, some of the obtained image feature vectors could be additionally shortened by applying some generic compression technique. Further experiments are needed to determine how much reduction can be achieved in this manner, and which of the descriptors are the most compressible.

Our results indicate that the performance drops more rapidly for some classes than others when the feature vector length is reduced. Those are mainly the classes with low number of instances, so we should aim to expand the dataset to include more examples of such classes. Only then will we be able to tell if there are any classes that are intrinsically hard to classify. We are also considering to redefine the classification problem to allow the assignment of multiple labels to a single image. The reason for this is that many misclassifications are occurring on the boundaries of scenes (tunnel entrances, exits) and because it is possible for a scene to have properties of multiple classes (overpass and traffic scenes can occur on highways or settlements). Another future goal is to reduce or eliminate the need for labeling the dataset manually. This might be achieved by automatic labeling in cases where we can infer the details of vehicle's environment with high degree of probability using other, non-visual cues.

(a) highway as: road (4 cases), traffic and overpass

(b) road as: highway (6 cases) and tunnel (2 cases)

(c) tunnel as: highway (5 cases), overpass, exit and traffic

(d) exit as: tunnel (6 cases), settlement, traffic

(e) settlement as: highway (6 cases), overpass, tunnel

(f) overpass as: settlement (5 cases), exit (2 cases), highway

(g) booth as: settlement (5 cases), road (2 cases) and highway

(h) traffic as: settlement (5 cases), exit (2 cases) and highway

Fig. 6: Examples of images that were hardest to classify. Only baseline descriptors paired with both tested classifiers were taken into account, for a total of 8 combinations. The highway was misclassified in 6 different cases, while all other examples were misclassified in all 8 cases. The incorrect classification predictions are listed under each example, along with the total number of cases in which they occurred.

REFERENCES

[1] I. Sikirić, K. Brkić, and S. Šegvić, "Classifying traffic scenes using the GIST image descriptor," in *CCVW 2013 Proceedings of the Croatian Computer Vision Workshop*, pp. 1–6, September 2013.

[2] A. Pinz, "Object categorization," *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 4, 2005.

[3] A. Bosch, X. Muñoz, and R. Martí, "Review: Which is the best way to organize/classify images by content?," *Image Vision Comput.*, vol. 25, pp. 778–791, June 2007.

[4] J. Luo, A. E. Savakis, and A. Singhal, "A Bayesian network-based framework for semantic image understanding," *Pattern Recogn.*, vol. 38, pp. 919–934, June 2005.

[5] F.-F. Li and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *CVPR*, (Washington, DC, USA), pp. 524–531, IEEE Computer Society, 2005.

[6] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification via pLSA," in *ECCV*, (Berlin, Heidelberg), pp. 517–530, Springer-Verlag, 2006.

[7] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, (Washington, DC, USA), pp. 2169–2178, IEEE Computer Society, 2006.

[8] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vision*, vol. 42, pp. 145–175, May 2001.

[9] A. Oliva and A. B. Torralba, "Scene-centered description from spatial envelope properties," in *BMCV*, (London, UK, UK), pp. 263–272, Springer-Verlag, 2002.

[10] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, pp. 3360–3367, 2010.

[11] J. Krapac, J. J. Verbeek, and F. Jurie, "Modeling spatial layout with Fisher vectors for image categorization," in *ICCV*, 2011.

[12] A. Ess, T. Mueller, H. Grabner, and L. v. Gool, "Segmentation-based urban traffic scene understanding," in *BMVC*, pp. 84.1–84.11, BMVA Press, 2009.

[13] I. Tang and T. Breckon, "Automatic road environment classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 476–484, June 2011.

[14] L. Mioulet, T. Breckon, A. Mouton, H. Liang, and T. Morie, "Gabor features for real-time road environment classification," in *ICIT*, pp. 1117–1121, IEEE, February 2013.

[15] A. Bergamo, L. Torresani, and A. W. Fitzgibbon, "PiCoDes: Learning a compact code for novel-category recognition," in *NIPS*, pp. 2088–2096, 2011.

[16] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *BMVC*, 2011.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, pp. 91–110, 2004.

[18] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[19] F. Perronnin and C. R. Dance, "Fisher kernels on visual vocabularies for image categorization," in *CVPR*, 2007.

[20] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010.

[21] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.

[22] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[23] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[24] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms." `http://www.vlfeat.org/`, 2008.

[25] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[26] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.