# Robust Traffic Scene Recognition with a Limited Descriptor Length

Ivan Sikirić
Mireo d.d.
Buzinski prilaz 32
HR-10000 Zagreb, Croatia
ivan.sikiric@mireo.hr

Karla Brkić, Josip Krapac, Siniša Šegvić
University of Zagreb
Faculty of Electrical Engineering and Computing
Unska 3, HR-10000 Zagreb, Croatia
karla.brkic@fer.hr, josip.krapac@fer.hr, sinisa.segvic@fer.hr

## Abstract

*In this paper we describe a novel image descriptor designed for classification of traffic scene images in fleet management systems. The descriptor is computationally simple and very compact (as short as 48 bytes). It is derived from variations of two well known image descriptors: GIST and spatial Fisher vectors, thus encoding both global and local image features. Both GIST (being a global scene descriptor) and spatial Fisher vectors (that relies on local image features) are tuned to produce very short outputs (64 components), which are then concatenated. The output is further compressed by a lossy encoding scheme, without sacrificing classification performance. The encoding scheme uses as little as 3 bits to encode each vector component. The descriptor is evaluated on the publicly available FM2 dataset of traffic scene images. We demonstrate very good classification performance matching that of full-sized general purpose image descriptors.*

## 1. Introduction

Recognizing visual scenes while limiting the descriptor size is a challenging problem with potential use in many scenarios involving thin clients with limited bandwidths. Examples include autonomous unmanned aerial vehicles [35], driver assistance systems [13], fleet management systems [29], mobile robots in emergency response situations [30], etc. We are assuming a scenario in which the server is interested in retrieving and storing the information about the visual surroundings of one or multiple thin clients at regular time intervals for a prolonged period of time (ranging from a week to a year), as illustrated in Figure 1. This information is used for further processing, e.g. for cross-checking GPS data, obtaining a semantic analysis of the thin clients' behavior such as "the UAV height loss is correlated to the presence of birds in the scene", etc. In such a scenario, working with the raw image data generated by the thin clients is prohibitively expensive both in terms of data transfer and in terms of storage. Hence, a reasonable strategy is storing only the *descriptors* of the images, and making these descriptors as short as possible.

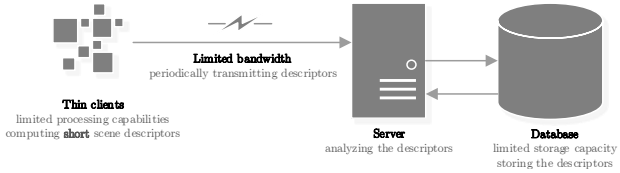



Figure 1: Our target application framework, where a number of thin clients send information about their surroundings to a server via limited bandwidth.

Figure 2: Examples of scene types of interest in fleet management systems, from the FM2 dataset [29].

This work deals with visual scene representations using a very limited descriptor size (512 bytes and less). We are specifically motivated by fleet management systems, where a central server tracks the locations of a fleet of vehicles at any given time. The vehicles are equipped with a range of sensors measuring a number of vehicle properties, a GPS sensor and a camera. The central server generates various reports consisting of e.g. routes traveled, total number of miles, fuel expenditure etc. One recurring problem in fleet management systems is the GPS sensor precision. Due to erroneous GPS output, it is often very hard to accurately reconstruct the route the vehicle has traveled. GPS errors are most common in specific and visually easily recognizable places such as tunnels, toll booths or under overpasses. Therefore, GPS ambiguity in fleet management systems could be resolved by storing the camera image of the ve-

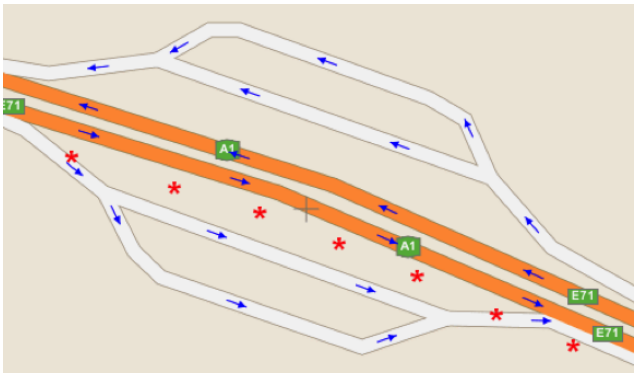



Figure 3: Ambiguous route reconstruction due to poor GPS precision. GPS readings are marked with red asterisks. It is equally plausible that the vehicle travelled the highway (in orange) and the local road (in white).

hicle's surroundings and using it as a cue for both the likely GPS failure and the correct location, e.g. by discerning local roads from highways and identifying tunnels and overpasses (Figure 2). Using this information, ambiguous routes such as the one in Figure 3 could be easily resolved. However, typical volumes of information generated by the fleet of vehicles, often involving hundreds of vehicles sending in information every minute, make it implausible to transmit and store entire images. The scene information should be stored in a descriptor of a very small size that is still sufficiently discriminative.

In this paper, we propose an efficient and short image descriptor that offers the same discriminative performance as its state-of-the-art counterparts that are one or more orders of magnitude longer. We achieve this by combining the GIST descriptor [26, 27] with bag-of-words-based spatial Fisher vectors [17]. While GIST is a global descriptor, spatial Fisher vectors aggregate local image features, and we aim to capture "the best of both worlds". We address the problem of descriptor length by proposing an efficient encoding method, allowing some loss of data accuracy and precision, but without affecting measurable classification performance.

Note that beside fleet management systems, our approach could also be used to aid place recognition systems, especially in scenarios of limited bandwidth. For example, it could provide a prior information on the scene type. The scene type could be used to partition the search space for matching scenes, thus making the search process faster.

## 2. Related work

Our work is closely related to the following topics: (i) generic image classification, (ii) short image descriptors, (iii) traffic scene understanding and fleet management systems and (iv) learning invariant properties of places and invariant image features.

### 2.1. Image classification

Bosch et al. [4] categorize generic scene modeling methods for classification as either low-level or semantic. Methods categorized as low-level simply represent the image as a collection of low-level features (e.g. color histograms) in order to reduce the dimensionality for classification. In contrast, methods categorized as semantic can be thought of as having some kind of prior understanding of what is being represented. Examples of semantic methods include semantic objects, where an object detector signals the presence or absence of a certain object type in order to assign a label to the image [22], bags of visual words [12, 5, 19], where a set of meaningful visual words is retrieved from the training data and used to classify query images, the GIST descriptor [26, 27], where the scene is represented globally via a set of filter responses, etc.

Bags of visual words have proven to be very successful in the problem of image classification with a large number of classes, as seen e.g. on the Pascal VOC dataset [11]. A number of extensions and modifications of the original bags of visual words have been proposed [8], including locality-constrained linear coding (LLC) [34] and spatial Fisher vectors (SFV) [17].

### 2.2. Traffic scene understanding

In traffic scene analysis, most researchers devise methods that are closely tailored to particular applications, given the constrained nature of the problem. Depending on the target application, methods range from scene segmentation and understanding at a pixel level [10] to generic scenery classification [32], or image-based localization [15].

Tang and Breckon [32] propose a method for road environment classification from images based on a set of color and texture-based features extracted at predefined regions of interest. They manually define the regions of interest to correspond to a sample of the road (in the center of the image), the part of the road where it is the widest (at the bottom of the image), and the environment at the left side of the road. Two classification problems are considered: either classifying images as off-road or on-road or a more fine-grained classification into off-road, urban scenery, major road and multilane motorway images. The work of Tang and Breckon is extended in [24], where only Gabor features are used, and the system is implemented using dedicated hardware.

### 2.3. Short image descriptors

The volume of research on short image descriptors is relatively modest. Bergamo et al. have recently introduced PiCoDes [3] and mc (meta-class) [2] descriptors, which produce very short image representations for the purpose of ef-

ficient image indexing in large image databases. Although very good classification performance can be obtained, producing such descriptors is computationally very intensive, which makes them unsuitable for our thin client scenario.

Sikirić et al. [29] investigate the use of a number of state-of-the-art descriptors in the problem of traffic scene classification, with an emphasis on the descriptor length. They describe simple ways to tune each of the considered methods to produce very short descriptors (as short as 64 floating point numbers) while still retaining very good classification performance. Their findings show that GIST and spatial Fisher vectors offer the best performance among the tested descriptors when the goal is to minimize the feature vector length.

### 2.4. Visual scene understanding

Visual place recognition is of special interest in the problem of simultaneous localization and mapping (SLAM) that is actively researched in robotics [9, 23, 28]. Milford and Wyeth [23] propose SeqSLAM, a SLAM algorithm that uses full images instead of features such as SIFT [21] or SURF [1], matches image sequences instead of single images, and uses local contrast enhancement in the image distance matrix with the idea of reinforcing local instead of global match optimums. Input images are reduced to a very small size (e.g. $64 \times 32$ pixels) and patch-normalized. Sünderhauf et al. [31] apply SeqSLAM on four video sequences of a 728 km journey, one for each season of the year. They report very good performance for sequences longer than 10 seconds, and a significant performance drop for mild viewpoint changes. The problem of viewpoint variance is addressed in SMART, a SeqSLAM-inspired algorithm proposed by Pepperell et al. [28], by using variable offset image matching. SMART offers several other improvements over SeqSLAM such as sky blackening. Sünderhauf and Protzel [31] introduce a descriptor for SLAM, called BRIEF-Gist, based on the BRIEF image descriptor [6], comparable to the popular FAB-Map [9], but much more computationally efficient.

### 2.5. Our contributions

In this paper, we build on the work of Sikirić et al. [29], who have found that GIST and spatial Fisher vectors perform best in traffic scene classification when limiting the descriptor length. We propose a way of combining these two descriptors and efficiently encoding them in order to obtain a short scene representation that captures both local and global image structure.

Given the constrained nature of our target application that involves a fleet management system serving a large number of vehicles with low processing power and limited bandwidth, we do not consider processing-intensive approaches such as PiCoDeS [3] and mc (meta-class) [2]. Although SeqSLAM and SMART could be suitable in our target scenario, we do not consider them because of their reliance on scaled-down images. When scaled-down with the default parameters of SeqSLAM/SMART, each image amounts to about 6 kilobytes of data, and our goal is transmitting the descriptors of the size of 512 bytes and less. For similar reasons, we do not consider BRIEF-Gist, as with best-performing parameters it produces around 3 kilobytes of data per image.

We now briefly review the GIST image descriptor and spatial Fisher vectors and proceed to describe our method.

## 3. The GIST image descriptor

The GIST image descriptor [26, 27] is a global scene descriptor that provides a rough representation of the scene structure by coarsely representing the distribution of orientations and scales in the scene. The descriptor is built by convolving the input image with a number of Gabor filters of varying scales and orientations. Each of the response images is then divided into a regular grid of regions, and the responses in each region are averaged. The final GIST descriptor is the concatenation of all the averaged responses. GIST is a compact representation that is very fast to compute [25], making it suitable for our target application scenario.

In [29], GIST is shown to be highly discriminative in traffic scene classification even when its size is reduced from 512 to 64 components. The reduction in size is achieved by using a grid of $2 \times 2$ regions instead of the default $4 \times 4$, as well as using 4 orientations per scale instead of the default 8.

## 4. Spatial Fisher vectors

Spatial Fisher vectors [17] are an extension of the well-known bag of visual words [12, 5, 19] descriptor. In bags of visual words, each image is represented as an occurrence histogram of a series of characteristic visual words. The visual words used for building the histogram comprise the so-called visual vocabulary that is learned from a set of training images. To build the vocabulary, a number of patches is extracted from each image, either through the use of an interest point detector or through dense sampling. Each patch is represented with a patch descriptor, e.g. SIFT [21]. The descriptors of the collected patches from all images are then clustered into $K$ clusters, and cluster centroids represent visual words constituting a $K$-word vocabulary. Given an image to be represented, the bag-of-visual-words descriptor is built by extracting image patches and their descriptors from the image in the same manner as in the training process, finding the visual words that are nearest to the extracted patches and building the histogram of the numbers of occurrences of all the words in the visual vocabulary.

A weakness of the classical bags of visual words is that the representation does not retain any information about the spatial layout of the image. This problem is addressed in spatial Fisher vectors [17], where the spatial mean and variance of the image regions associated with individual visual words are encoded in the descriptors. The encoding is based on the Fisher kernel framework and Gaussian mixture models, used to represent appearance and spatial layout. The positions of the patches that are assigned to a visual word are modeled using a Gaussian mixture model, so their spatial layout can be coded using one spatial Fisher vector per visual word. Additionally, Fisher kernels are also used to encode the appearance of the features, reframing the original bag of visual words descriptors as Fisher vector representations for a simple multinomial probabilistic model. It has been shown [17] that spatial Fisher vectors can achieve equal performance as the original bags of visual words using a smaller visual vocabulary, due to a more precise encoding of the appearance information. Hence, the computational cost of building a spatial Fisher vector representation is lower than for the classical bags of visual words and the representation itself is shorter, making spatial Fisher vectors especially suitable for our thin client with a limited bandwidth scenario.

## 5. Our method

The GIST descriptor is designed for scene recognition, and focuses of global image features while ignoring the small details in the scene. The spatial Fisher vectors descriptor is intended for general-purpose classification, and relies only on local image features. This distinction in the level of captured details is reflected on the classification performance of each descriptor in the context of traffic scenes, as shown e.g. in [29], where the GIST descriptor performs poorly on classes *overpass* and *traffic*, while the spatial Fisher vectors perform poorly on classes *road* and *toll booth*. Recognition of overpasses and dense traffic in the scene is heavily dependent on modeling local features, while recognizing roads and toll booths is easier using global scene models.

Intuitively, by using a simple concatenation of the GIST descriptors and the spatial Fisher vectors we would capture both the global and local image features, and therefore improve the overall classification performance. Additionally, we expect that by combining a GIST feature vector of length $K$ and a spatial Fisher vector of length $K$, we might get better classification performance than by using either GIST of spatial Fisher vectors of length $2K$. Therefore, as a first step in building a short and descriptive scene representation we propose to concatenate the GIST and the spatial Fisher vector descriptor of a scene into a single vector, as illustrated at the top of Figure 4. Each component is then normalized to have zero mean and unit variance, by subtracting the mean and dividing by variance, which are calculated from the training dataset.

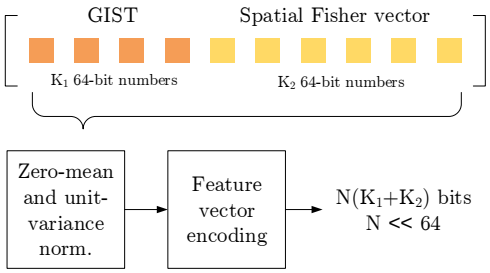



Figure 4: Our method: after obtaining shortened GIST and spatial Fisher vectors descriptors (of lengths $K_1$ and $K_2$, respectively), their concatenation is normalized to zero mean, unit variance. Each component of the result is then encoded using only N bits, so the total size of the output is $N(K_1 + K_2)$ bits.

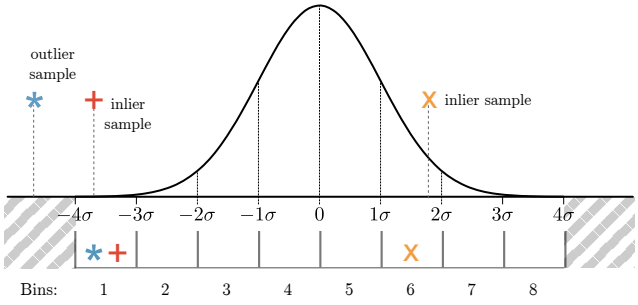
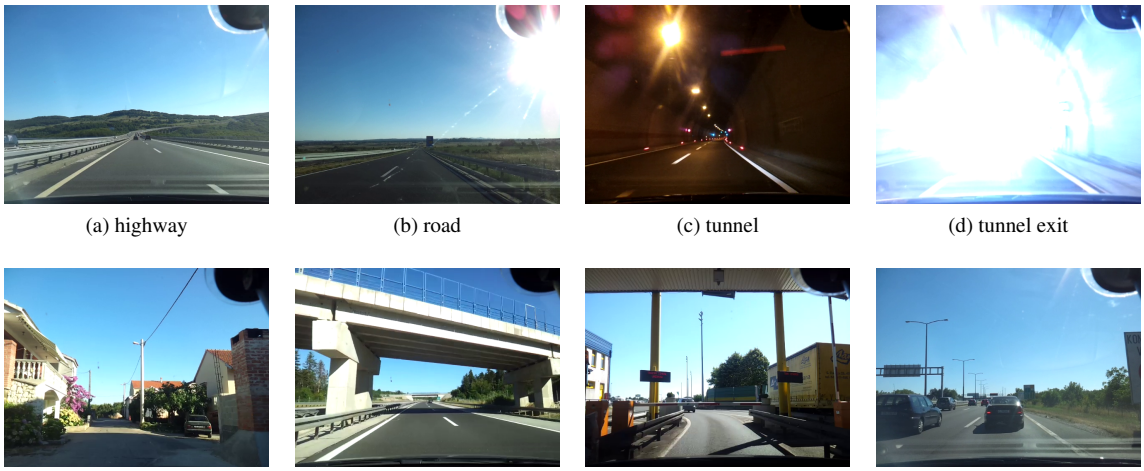



Figure 5: The encoding scheme for $N = 3$ bits and the interval width of $w = 4\sigma$. The closed interval $[-4\sigma, 4\sigma]$ is split into $2^3 = 8$ equally wide bins. Value mapping is illustrated for three samples (orange, red and blue). Note that the values outside the interval (the blue asterisk) are mapped to the nearest bin.

To additionally reduce the memory footprint of the image feature vectors (and therefore also the bandwidth consumption), we propose a non-standard feature vector representation. The standard and straightforward representation of an image feature vector is an array of floating point numbers of single or double precision, which uses 4 or 8 bytes per vector component, respectively. Our goal is to reduce the memory consumption to a maximum of 1 byte per component.

Instead of trying to encode the sign, exponent and mantissa parts of a floating point number in 8 bits or less, we encode each vector component in a fixed point representation, thus losing some accuracy and precision. More precisely, to encode a vector component using only $N$ bits, we first map its value to a closed interval $[L, R]$, which is split into $2^N$ discrete bins. The index of a bin the value falls into is the encoded value of the component. Mapping into an interval



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

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

Figure 6: Examples of classes from the FM2 dataset

is done in such a way that a value outside the interval is set to the appropriate upper or lower interval limit, while other values remain unchanged. An illustration of this process is shown in Figure 5. The illustrated example assumes normal data distribution, with zero mean. The width of the mapping interval is set to include 4 standard deviations from the mean, thus including almost all observable values. This interval is split into $2^3 = 8$ equally wide bins, and the index of the bin serves as the encoded value. If a value falls outside of the mapping interval, it is assigned to its nearest bin. So in this example, each component can be encoded using only 3 bits. Decoding process is straightforward, we simply replace each bin index with the coordinate of the bin center. In the example in Figure 5 the decoded values for the orange, red and blue samples would be $-3.5\sigma$, $-3.5\sigma$ and $1.5\sigma$, respectively. Using a simple experimental setup we can easily find the most appropriate interval width and number of bins for a given dataset and feature vector length.

## 6. Experiments

To experimentally validate the proposed descriptor, we evaluate it on a dataset of traffic scenes while varying the parameters of the underlying GIST and spatial Fisher vector descriptors, as well as the parameters of the feature vector encoding. We now give more details on the dataset we used, describe our experimental setup, and present the results.

### 6.1. The FM2 dataset

The FM2 dataset [29] is a dataset of road scenes containing 6237 images categorized into eight classes: *highway*, *road*, *tunnel*, *tunnel exit*, *settlement*, *overpass*, *toll booth* and *dense traffic*. The images were recorded on roads in Europe via a cellphone camera mounted inside a vehicle, mainly at daytime and during sunny weather, using a fairly constant viewpoint. The class labels are specifically intended to be useful in a fleet management scenario, i.e. cover possible situations in which the loss of GPS precision, traffic jams and slowdowns are likely. Example images of all classes in the dataset are shown in Figure 2. The class balance is skewed in favor of highway images, as illustrated in Table 1.

| Class label | Images |
|---|---|
| highway | 4337 |
| road | 516 |
| tunnel | 601 |
| tunnel exit | 64 |
| settlement | 464 |
| overpass | 86 |
| toll booth | 75 |
| dense traffic | 94 |

Table 1: Image distributions per class, the FM2 dataset.

The FM2 dataset comes with a publicly available premade train/validation/test splits, a 25/25/50 split for each class. In our experiments we use these splits to ensure comparability of our results with previous work [29].

We note one weakness of the FM2 dataset labeling: only one class label is assigned per image, although there are images that could be assigned to multiple classes. Examples include e.g. toll booths and overpasses on a highway, dense traffic in a settlement etc. In this work, we retain the original class labels to remain comparable to [29]; however, assigning multiple labels to a single image should be considered in future work.

## 6.2. Experimental setup

To evaluate the classification performance of the proposed descriptor combination, we use the method proposed in [29] to obtain GIST feature vectors and the spatial Fisher vectors of lengths 64 and 128. From now on, we refer to the spatial Fisher vectors of lengths 64, 128, etc. as SFV64 and SFV128, etc. Similarly for GIST, we refer to the GIST descriptors of lengths 64, 128, etc. as GIST64, GIST128, etc.

We measure the classification performance of three different descriptor concatenations: GIST64+SFV64, GIST64+SFV128 and GIST128+SFV64. The classification is performed with SVM (RBF kernel) and Random forest classifiers, and we use mAP (mean of per-class average precision) as a measure of performance. We will now describe each experimental component in more detail.

### 6.2.1 The descriptors

For spatial Fisher vectors we use the dense SIFT implementation from the VLFeat library [33] as a patch descriptor. The code for producing the spatial Fisher vectors themselves is based on code of Krapac et al. [17]. The Gaussian mixture model parameters are learned using the expectation-maximization algorithm, and the diagonal approximation of covariance matrix is used both for local descriptors and position features. We use the first 6 principal components of the SIFT descriptors, and only one Gaussian per visual word, which means we use $K$ visual words to obtain a vector of length $16K$.

For the GIST descriptor we use the MATLAB implementation provided by the authors [26, 27]. The implementation was modified to reduce the grid size to $2 \times 2$ in order to produce the 128-dimensional GIST descriptors. By further reducing the number of orientations per scale from 8 to 4, we obtain the 64-dimensional GIST descriptors.

### 6.2.2 The classification

As a preprocessing step for SVM classification, we transform all the feature vectors to zero mean and unit variance. The classification is then performed using the LibSVM library [7], using the RBF kernel. For Random forest classification we employ the code of Liaw et al. [20].

We use the default train/validation/test splits of the FM2 dataset (25/25/50 split for each class). The classifier is trained on the training set, while the validation set is used to optimize the parameters. Finally, the classifier is trained on the union of training and validation sets using the optimized parameters, and the final score is evaluated on the test set. In each step the mean average precision (mAP) is used as a performance measure.

### 6.2.3 Feature vector encoding

We will now describe the experimental setup used to explore the proposed memory-efficient vector encoding scheme. Let us briefly review the method: for each vector component, its value is mapped to a closed interval $[L, R]$, which is split into $2^N$ bins, thus encoding each component using only N bits. Values outside the interval are mapped to the nearest bin. Since we got very good results using a GIST64+SFV64 concatenation with SVM classifier, we choose this combination as our baseline. These feature vectors were transformed to have zero mean and unit variance, so the appropriate interval is of the form $[-w, +w]$. By setting the upper limit of the interval to three standard deviations ($w = 3$), most of the observed values would be included in it (even without assuming normal distribution). However, the extreme values (falling outside of range of three standard deviations) could prove to be beneficial in successful class discrimination. To explore this, we run the experiments with $w$ set to 3, 3.5, 4, 4.5 and 5. We split these intervals into $2^N$ bins, with $N$ set to 1, 2, 3, 4 and 8. For simplicity, we use the bins of equal size. We measure the classification performance using the same instance of SVM classifier which produced the optimal results for the unencoded (full-size) GIST64+SFV64 combination. The vectors from the testing dataset split are encoded and then decoded (thus introducing errors), and the classification performance is then measured.

## 6.3. Experimental results

Classification results for the concatenated descriptors are presented in Table 2 for the SVM classifier, and in Table 3 for Random forest. The original results for the plain SFV and GIST descriptors from [29] are presented for convenience. There are several things we can note. Firstly, our combined descriptor GIST64+SFV64 indeed outperforms both plain GIST128 and SFV128 descriptors, in case of both SVM and Random forest classifiers. In fact, better performance is only achieved using the spatial Fisher vectors of length 2656 with an SVM classifier. We can also see that the SVM shows much better performance than the Random forest classifier for all three combinations of our descriptors. Next, we note that slightly better performance is achieved with GIST64+SFV128 than with GIST128+SFV64 descriptor. This implies that we should put more emphasis on local image features to enable optimal class discrimination.

Graphical display of per-class performance for the best-performing setups with lengths of 128 is shown in Figure 7. The graph implies that the class *traffic*, containing images of dense traffic is the one that most benefits from combining global and local image features.

The results of our feature vector encoding experiments are shown in Table 4. The table shows dependency of mean average precision on two factors: number of bits per compo-

| Descriptor | Highway | Road | Tunnel | Exit | Settlement | Overpass | Booth | Traffic | Mean |
|---|---|---|---|---|---|---|---|---|---|
| GIST 128 + SFV 64 | 99.91 | 96.17 | 99.88 | 97.68 | 96.61 | 87.29 | 86.77 | 92.05 | 94.55 |
| GIST 64 + SFV 128 | 99.90 | 93.98 | 99.95 | 98.78 | 97.06 | 87.38 | 88.84 | 94.70 | 95.08 |
| GIST 64 + SFV 64 | 99.90 | 96.05 | 99.83 | 95.27 | 96.99 | 88.03 | 86.14 | 92.80 | 94.38 |
| GIST 512 [29] | 99.84 | 93.72 | 99.76 | 98.11 | 97.05 | 83.31 | 94.40 | 80.21 | 93.30 |
| GIST 128 [29] | 99.66 | 90.28 | 99.51 | 96.61 | 94.86 | 82.15 | 89.87 | 68.60 | 90.19 |
| SFV 2656 [29] | 99.94 | 96.30 | 99.87 | 95.79 | 97.03 | 92.51 | 90.74 | 88.57 | 95.09 |
| SFV 128 [29] | 99.19 | 77.74 | 98.83 | 82.68 | 86.29 | 80.96 | 78.74 | 69.79 | 84.28 |

Table 2: Per-class average precision (percentage), SVM classifier

| Descriptor | Highway | Road | Tunnel | Exit | Settlement | Overpass | Booth | Traffic | Mean |
|---|---|---|---|---|---|---|---|---|---|
| GIST 128 + SFV 64 | 99.72 | 94.00 | 99.88 | 96.76 | 92.48 | 76.52 | 83.49 | 87.43 | 91.28 |
| GIST 64 + SFV 128 | 99.76 | 93.79 | 99.90 | 97.62 | 92.66 | 82.50 | 80.47 | 90.84 | 92.19 |
| GIST 64 + SFV 64 | 99.74 | 93.08 | 99.89 | 97.14 | 92.43 | 82.77 | 81.72 | 88.43 | 91.90 |
| GIST 512 [29] | 99.57 | 88.33 | 99.53 | 94.76 | 94.17 | 79.56 | 91.61 | 80.23 | 90.97 |
| GIST 128 [29] | 99.16 | 81.63 | 99.26 | 94.73 | 89.59 | 78.78 | 90.16 | 69.00 | 87.79 |
| SFV 2656 [29] | 99.83 | 91.27 | 99.56 | 85.37 | 95.64 | 89.66 | 91.98 | 79.48 | 91.60 |
| SFV 128 [29] | 99.76 | 90.78 | 99.10 | 87.17 | 95.46 | 82.86 | 90.22 | 73.22 | 89.82 |

Table 3: Per-class average precision (percentage), Random Forest classifier

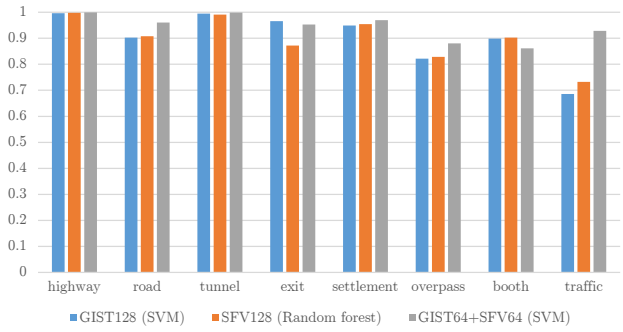



Figure 7: Per-class average precision for best performing setups with feature vectors of length 128

| w | 3 | 3.5 | 4 | 4.5 | 5 |
|---|---|---|---|---|---|
| 8 bits | 94.26 | 94.33 | 94.37 | 94.39 | 94.41 |
| 4 bits | 94.05 | 94.53 | 94.48 | 94.42 | 94.40 |
| 3 bits | 93.68 | 94.16 | 94.06 | 93.53 | 93.66 |
| 2 bits | 91.05 | 91.27 | 90.68 | 90.00 | 89.75 |
| 1 bit | 75.30 | 77.84 | 78.62 | 78.95 | 78.81 |

Table 4: Classification performance (mAP) for several encoding schemes of the GIST64+SFV64 descriptor, SVM classifier

nent, and width of the mapping interval (expressed in terms of number of standard deviations from the mean). The optimal width of the interval seems to be, in most cases, 3.5 to 4 standard deviations from the mean. The classification performance does not seem to decrease at all for 8 and 4 bits per component. By using only 3 bits to encode each vector component, we sacrifice very little classification performance compared to the baseline. For 2 bits per component we do see a significant drop in classification performance, mean average precision drops from 94% to 91%. Finally, using just 1 bit per component is clearly not enough, as the mean average precision drops to less than 80%. For FM2 dataset, we would recommend using 3 bits per component with width set to 3.5 standard deviations, as this will sacrifice very little class discrimination for a significant memory footprint reduction (10.7 times less than baseline).

## 7. Conclusion and outlook

We have confirmed that by combining global and local image features we significantly improve classification performance while not increasing feature vector length. On the FM2 dataset of traffic scene images, our combined descriptor of length 128 performs better than most other descriptors, regardless of their length. It is slightly outperformed only by a spatial Fisher vector descriptor of length 2656. According to Gehler and Nowozin [14], even greater improvements should be expected if we employ their LP-$\beta$, or any other advanced feature combination method, such as

MKL (multiple kernel learning) [18]. Our results indicate that a greater part of the feature vector components should be derived from local, rather than global image features, so this should be explored in future work.

Additionally, we have also presented a memory-efficient way of encoding the feature vectors, using as little as 3 bits per vector component. This way we have reduced the memory footprint of our combined descriptor of length 128 from 512 to 48 bytes, while measuring a drop of mean average precision of only 0.22%. Our method is simple and relies on small reduction of data accuracy and great reduction of data precision. The method does not require any modification in the training process, as the classifier can be trained on regular data and it will successfully classify the vectors with reduced precision. This is in itself an interesting result, and we invite other researchers interested in short image representation to try out our method on their datasets and descriptors. In our future work we will explore if additional improvements are possible using some method which compresses combined vector components, such as product quantization (PQ) [16].

## References


[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008. 3

[2] A. Bergamo and L. Torresani. Meta-class features for large-scale object categorization on a budget. In *Proc. CVPR*, 2012. 2, 3

[3] A. Bergamo, L. Torresani, and A. W. Fitzgibbon. PiCoDes: Learning a compact code for novel-category recognition. In *Proc. NIPS*, pages 2088–2096, 2011. 2, 3

[4] A. Bosch, X. Muñoz, and R. Martí. Review: Which is the best way to organize/classify images by content? *Image Vision Comput.*, 25(6):778–791, June 2007. 2

[5] A. Bosch, A. Zisserman, and X. Muñoz. Scene classification via pLSA. In *Proc. ECCV*, pages 517–530, Berlin, Heidelberg, 2006. Springer-Verlag. 2, 3

[6] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *Proc. ECCV*, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag. 3

[7] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. 6

[8] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *Proc. BMVC*, 2011. 2

[9] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *Int. J. Robot. Res.*, 27(6):647–665, 2008. 3

[10] A. Ess, T. Mueller, H. Grabner, and L. v. Gool. Segmentation-based urban traffic scene understanding. In *Proc. BMVC*, pages 84.1–84.11. BMVA Press, 2009. 2

[11] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010. 2

[12] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proc. CVPR*, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society. 2, 3

[13] M. Forster, R. Frank, M. Gerla, and T. Engel. A cooperative advanced driver assistance system to mitigate vehicular traffic shock waves. In *Proc. INFOCOM*, pages 1968–1976, April 2014. 1

[14] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *Proc. ICCV*, pages 221–228, Sept 2009. 7

[15] N. Ho and P. Chakravarty. Localization on freeways using the horizon line signature. In *Proc. of Workshop on Visual Place Recognition in Changing Environments at Int. Conf. on Rob. Autom. (ICRA)*, 2014. 2

[16] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(1):117–128, Jan 2011. 8

[17] J. Krapac, J. J. Verbeek, and F. Jurie. Modeling spatial layout with Fisher vectors for image categorization. In *Proc. ICCV*, 2011. 2, 3, 4, 6

[18] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, Dec. 2004. 8

[19] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society. 2, 3

[20] A. Liaw and M. Wiener. Classification and regression by randomForest. *R News*, 2(3):18–22, 2002. 6

[21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004. 3

[22] J. Luo, A. E. Savakis, and A. Singhal. A Bayesian network-based framework for semantic image understanding. *Pattern Recogn.*, 38(6):919–934, June 2005. 2

[23] M. Milford and G. Wyeth. SeqSLAM: visual route-based navigation for sunny summer days and stormy winter nights. In N. Papanikolopoulos, editor, *Proc. ICRA*, pages 1643–1649, River Centre, Saint Paul, Minnesota, 2012. IEEE. 3

[24] L. Mioulet, T. Breckon, A. Mouton, H. Liang, and T. Morie. Gabor features for real-time road environment classification. In *Proc. ICIT*, pages 1117–1121. IEEE, February 2013. 2

[25] A. Murillo, G. Singh, J. Kosecka, and J. Guerrero. Localization in urban environments using a panoramic gist descriptor. *Robotics, IEEE Transactions on*, 29(1):146–160, Feb 2013. 3

[26] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, May 2001. 2, 3, 6

[27] A. Oliva and A. B. Torralba. Scene-centered description from spatial envelope properties. In *Proc. BMCV*, pages 263–272, London, UK, UK, 2002. Springer-Verlag. 2, 3, 6

[28] E. Pepperell, P. Corke, and M. Milford. All-environment visual place recognition with SMART. In *Proc. ICRA*, pages 1612–1618, May 2014. 3

[29] I. Sikirić, K. Brkić, J. Krapac, and S. Šegvić. Image representations on a budget: Traffic scene classification in a restricted bandwidth scenario. *Proc. IEEE Intelligent Vehicles Symposium*, 2014. 1, 3, 4, 5, 6, 7

[30] D. Summers-Stay, T. Cassidy, and C. Voss. Joint navigation in commander/robot teams: Dialog & task performance when vision is bandwidth-limited. In *Proceedings of the Third Workshop on Vision and Language*, pages 9–16, Dublin, Ireland, August 2014. Dublin City University and the Association for Computational Linguistics. 1

[31] N. Sünderhauf, P. Neubert, and P. Protzel. Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons. In *Proc. of Workshop on Long-Term Autonomy at Int. Conf. on Rob. Autom. (ICRA)*, May 2014. 3

[32] I. Tang and T. Breckon. Automatic road environment classification. *IEEE Trans. Int. Transp. Sys.*, 12(2):476–484, June 2011. 2

[33] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008. 6

[34] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. CVPR*, pages 3360–3367, 2010. 2

[35] S. Wei, L. Ge, W. Yu, G. Chen, K. Pham, E. Blasch, D. Shen, and C. Lu. Simulation study of unmanned aerial vehicle communication networks addressing bandwidth disruptions. In *Proc. SPIE*, volume 9085, pages 90850O–90850O–10, 2014. 1