Segmentation-Based Urban Traffic Scene Understanding

Andreas Ess[1]
aess@vision.ee.ethz.ch
Tobias Müller[1]
muelleto@bluewin.ch
Helmut Grabner[1]
grabner@vision.ee.ethz.ch
Luc van Gool[1,2]
vangool@vision.ee.ethz.ch

[1] Computer Vision Laboratory
ETH Zürich
Switzerland
[2] ESAT-PSI / IBBT
K.U. Leuven
Belgium

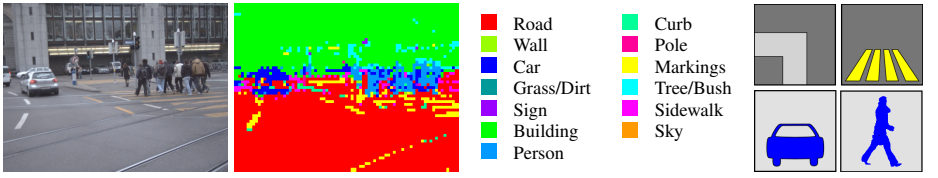
Abstract

We propose a method to recognize the traffic scene in front of a moving vehicle with respect to the road topology and the existence of objects. To this end, we use a two-stage system, where the first stage abstracts from the underlying image by means of a rough super-pixel segmentation of the scene. In a second stage, this meta representation is then used to construct a feature set for a classifier that is able to distinguish between different road types as well as detect the existence of commonly encountered objects, such as cars or pedestrian crossings. We show that by relying on an intermediate stage, we can effectively abstract from any peculiarities of the underlying image data due to *e.g.* color abberrations. The method is tested on two long, challenging urban data sets, covering both day light and dusk conditions. Compared to a state-of-the-art descriptor, we show improved classification performance, especially for object classes.

1 Introduction

Recognizing the traffic scene in front of a car is an important asset for autonomous driving, *e.g.* [5], as well as for safety systems. While GPS-based maps abound and have reached an incredible level of accuracy, they can still profit from additional, image-based information. Especially in urban scenarios, GPS reception can be shaky, or the map might not contain the latest detours due to constructions, demonstrations, etc. Furthermore, such maps are static and cannot account for other dynamic traffic agents, such as cars or pedestrians. In this paper, we therefore propose an image-based system that is able to recognize both the road type (straight, left/right curve, crossing, ...) as well as a set of often encountered objects (car, pedestrian, pedestrian crossing). The obtained information could then be fused with existing maps and either assist the driver directly (*e.g.*, a pedestrian crossing is ahead: slow down) or help in improving object tracking (*e.g.*, where are possible entrance/exit points for pedestrians or cars?).

BMVC 2009 doi:10.5244/C.23.84



(a) (b) (c) (d)

Figure 1: Given an input image (a), we calculate a meta representation based on a patch-wise scene classification (b) into a set of urban texture classes (c), which is then used to classify the scene with respect to road type, as well as to detect the presence of certain objects (d).

To this end, we employ a two-stage architecture termed Segmentation-Based Urban Traffic Scene Understanding (SUTSU) that first builds an intermediate representation of the image based on a patch-wise image classification. This yields a meta representation of the scene that is more suitable for further processing, Fig. 1. We choose this approach over a standard feature-based bag-of-words representation for several reasons: its reduced set of urban texture classes can be learned in a supervised way, it provides class probabilities for each image patch, and it naturally encodes the spatial layout of the input image. Furthermore, it is an intermediate representation that can easily be adapted to other underlying image data (*e.g.* dusk, rain, ...), without having to change the high-level classifier.

The paper is structured as follows. After reviewing related work in the upcoming section, Section 3 describes the patch-wise image classification. Section 4 shows how the obtained meta information can then be used for the actual scene classification. The employed data set, along with the used annotation methodology, is described in Section 5. Results for this data set, also comparing our method with another state-of-the-art scene descriptor, are presented in Section 6, before the paper is concluded in Section 7.

2 Related Work

Scene categorization has been a very active research field over the past years, we are however not aware of any direct application to autonomous driving in urban scenarios.

By segmenting an incoming image into meaningful classes, our intermediate representation can be thought of as a first level of scene categorization. Several works have proposed systems that first divide the image into a set of super pixels (either based on an oversegmentation [15] or a regular grid [21]) and then use a set of appearance and geometry features to obtain a class label for each image patch [2, 15, 21, 22, 24]. Most of these works focus on the improvement of object detection [16, 24] or single-image 3D reconstruction [15]. The local labeling however fails to capture higher-level relationships: we know which pixels belong to the road or might be a building, but it is not immediately evident how these relate with each other, what the scene in front of the vehicle actually is.

Another branch of work is interested in a more holistic interpretation of the image. Fostered especially by data sets such as CALTECH-101 [8] or the PASCAL VOC challenge [7], researchers have proposed several methods to classify an image into categories. The most successful and popular underlying approach is the bag-of-words representation [10, 17], with some approaches going into concurrent object segmentation and classification [3]. It is based on a visual vocabulary that would need to be retrained for new image sets (in our approach,

only the intermediate patch classifier needs retraining), and does not readily encode spatial relationships. Also aimed at a global image classification, Oliva and Torralba [19] suggest a "spatial envelope" (GIST) that can reliably distinguish between a wide variety of different classes of scenes, even between streets and highways. To some extent, GIST also manages to distinguish subclasses of street, as we show in the results section.

Understanding traffic scenarios per se is currently often limited to analyzing typical objects, *e.g.* by object tracking [6, 12], lane finding [18] and traffic sign detection [1] algorithms. We are, in contrast, mainly interested in the type of the upcoming road section as well as the presence of a typical set of objects. This requires more high-level information than a patch-wise segmentation, and needs more specialized information than standard classification approaches in order to deal with the highly similar subclasses and their difficult appearances.

3 Patch Classification

Rough scene segmentation is usually done by first dividing the image into subregions (patches), then defining appropriate features for the desired classes, and finally applying some learning algorithm. We base our method on the work of Wojek and Schiele [24], but restrict ourselves here to their basic patch classifier and adapt the set of classes for inner-city scenarios.

Features. To ensure similar color appearance, the input color image is corrected in $L^*a^*b^*$-space by applying a gray-world assumption. That is, one assumes the mean color in an image from an urban scenario to be gray, corresponding to the fact that the mean of the a^* and b^* channels needs to equal 0. After color correction, the image is subdivided in $N \times M$ patches of size 8×8, which seems to be a good compromise between sufficient texture information and segmentation granularity. In each patch, the Walsh-Hadamard transform (*e.g.* [14]) is calculated on all three color channels. This transformation is a decomposition of a patch into square waves, capturing its texture properties. To increase context and allow for a certain level of scale invariance, we not only calculate the transform in the patch, but additionally also on an extended 16×16 neighborhood, centered around the patch. If available, depth information from stereo can be used to generate two additional features: the median depth of a patch, as well as its height above the ground, which especially helps to distinguish standing structures (*i.e.* objects, buildings, ...) from the ground. In total, this yields a set of 961 features per patch, or 963 with additional depth information.

Learning. We assume that each patch belongs to either one of $C_p = 13$ classes (street, car, ...; see Fig. 1 (c)). The classes were chosen as to reflect the most common textures found in an urban scene. For each class, a discriminate classifier is trained independently in a one-versus-all manner. Learning is performed using discrete AdaBoost [11] for feature selection [23]. In general, boosting forms a strong classifier by a linear combination of weak classifiers. The weak classifiers are trained sequentially on a reweighed set of the labeled training data (*i.e.*, weights of misclassified examples are increased and thus the algorithm focuses on the hard-to-learn examples). For feature selection, each weak classifier corresponds to a feature (defined above) and the best performing feature (the one with the lowest error) is chosen and added to the strong classifier. We use a simple decision stump as weak learning algorithm, and select 500 features from the available pool.

Application. In order to label an image, each patch is processed independently, first calculating the features and then applying all C_p classifiers. The responses can be interpreted as the probability that the patch corresponds to the class. These probabilities are used as input for the further processing stages. For hard decisions, *e.g.* for images, the label of the classifier with the maximum response is chosen.

The obtained segmentation can be used for a wide variety of applications (*e.g.*, [15, 16, 24]). In the following section, we will demonstrate its use for urban scene classification.

4 Scene Classification

Based on the meta representation obtained from the previous stage, we proceed to infer a basic understanding of the current traffic scene in front of the vehicle. Specifically, we aim to distinguish 8 different types of road layouts, and to detect the presence of cars, pedestrians, or a pedestrian crossing in front of the vehicle, Tab. 1.

Features. We employ 3 different sets of features to capture the discriminating properties of a traffic scene: rough layout, periodic structures, and orientation histograms.

- **Rough layout.** First, to get the basic underlying structure of the image, we use a hierarchical representation obtained by downsampling the patch classifier's probability maps into maps of size 2×2, 4×4, and 8×8 by mean-filtering, yielding $C_p\cdot(4+16+64)$ features. This is in a sense similar to image pyramids [13], which were shown to be very effective in image classification [17].

  However, the higher levels of a hierarchy wash out the spatial information, while learning on the actual segmentation is not invariant to slight perspective changes unless considerably more training data is used. A certain invariance can be achieved by calculating the mean classifier strength of all classes for each row, respectively each column, yielding another $C_p\cdot(N+M)$ features. The mean over rows is *e.g.* helpful to detect whether an object is front of the observer disregarding its x-position, whereas columns can give an idea of the road structure.

- **Periodic structures.** With either of the above feature sets, it is difficult to keep a periodic structure like a pedestrian crossing apart from standard road markings. Therefore, we try to measure periodicity using another feature set that is constructed by again subsampling the patch classifier's probability maps into either 8 or 16 rows and then performing auto-correlation over columns. The periodicity is then recorded as the location and strength of the first local maximum, another $2N\cdot(8+16)$ features.

- **Orientation.** Lastly, to get an idea of the road direction (straight, curve, ...), we measure the orientation of road markings and the curb. Specifically, we apply an orientation operator [4] to both the probability maps corresponding to road marking and curb, divide the resulting maps into 4×4 regions and create an orientation histograms with 18 bins for each region, giving yet another 288 features.

Learning. With one probability map for each of the $C_p = 13$ classes, and $N = 80, M = 60$, this gives a pool of $6,080$ features. As with the patch classification, we use boosting for feature selection to select the most important features from the available pool for each class.

	# Frames											
DAY	22,500	9,697	2,313	2,254	621	852	2,735	1,701	1,377	8,424	3,924	5,398
DUSK	15,000	7,512	1,373	1,568	186	248	999	641	2,121	3,300	5,478	2,601

Table 1: Distribution of classes in the employed sequences (number of frames containing a specific category). At 13 *fps*, this corresponds to roughly 50 (29+19) minutes of data.

Again, the classifiers for each road type/object are learned independently using a one-versus-all training scheme, and 200 features are selected.

Temporal Smoothing. Working from video, it is sensible to use some sort of temporal smoothing for the classifiers' output. We opt for a Hidden Markov Model (HMM, *cf*. *e.g.* [20]). To allow online application, we only report the output of the forward pass of the HMM in our experiments. For the road type, we use a single HMM with 8 states corresponding to the respective type, where the transition probabilities between the different types are learned from the training set. For the object classes, we use one independent HMM each, as their occurence is not directly coupled with the road type or the other objects. Again, the respective transition probabilities are learned from the training set.

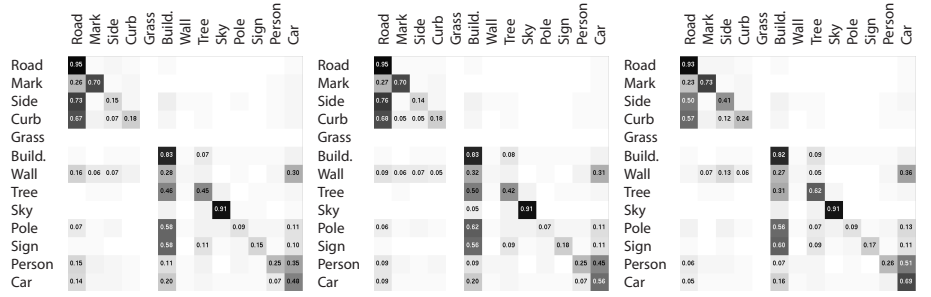
Application. For each input image of the test set, the scene classifiers are applied independently, and the HMMs are run. For the road type, we report the current maximum state, for the object classes, we report their existence if the classifier's probability is > 0.5.

5 Data Set

We use two challenging data sets recorded from a moving vehicle in the same urban environment. The car is equipped with a synchronized pair of cameras, yielding roughly 13 images per second at a resolution of 640×480 px. We will use the left camera's output, but in some cases also the depth maps generated from the stereo pair using the algorithm of [9]. The first data set (Seq. DAY), spanning 22,500 frames, was recorded during the day and is used for training/testing the scene classifier via cross validation. The second data set (Seq. DUSK), spans 15,000 frames and was recorded later on the same day but is quite different with respect to the number of moving objects (rush hour) and color distribution of the images (dusk and read/headlights from cars). Seq. DUSK is only used for testing the scene classifier.

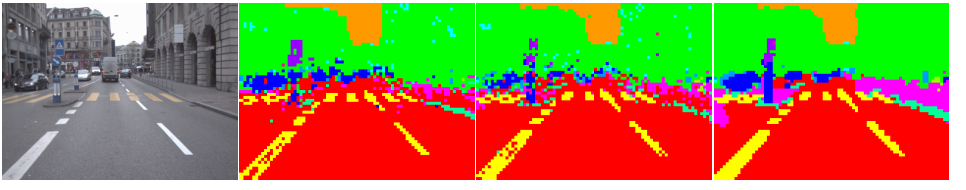
Patch Classifier. For training the patch classifier, 39 images from another, similar training set were segmented manually into the texture classes (Fig. 1 (c)).

Scene Classifier. Each image of both sequences was assigned to one of eight road types. Additionally, the existence of an object (pedestrian crossing, car, pedestrian) was marked. An overview of the class distribution on the training data can be seen in Tab. 1. Annotation was done as follows: each image was assigned to one of the available eight road types as soon as the beginning of the road type was below the lower third of the image (this corresponds to a distance of ≈ 20 m and was similar to the subjective feeling of when an image is considered as a given road type). *E.g.*, as soon as the lower border of an incoming street of a junction was below the middle line, the road type was assigned to "junction". The rather large distance



(a) Appearance-only classifier (b) Including depth (c) MRF and depth

Figure 2: Confusion matrices for different variations of the basic patch classifier.



(a) (b) (c) (d)

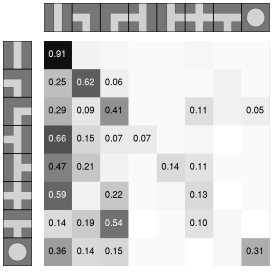
Figure 3: Example of basic patch classification of an image (a) using only appearance (b), additional depth map features (c), and MRF-based smoothing (d).

makes the problem quite hard, as the spatial resolution for the discriminating parts (with the rest of the junction even farther away) is very low: there are usually only around 5 rows of patches, *cf*. Fig. 1.

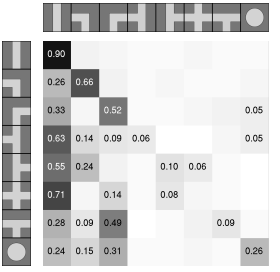
Additionally to the road type, flags were set to indicate an object's presence. Again, pedestrian crossings are annotated as soon as they are closer than $\approx 20\,\mathrm{m}$ and $\approx 10\,\mathrm{m}$ for cars. Pedestrians are split into two sets: one directly in front of the car (*e.g.*, at a pedestrian crossing), and pedestrians on the side with a height of $\approx 20\%$ of the image height. Due to their rather different features, we trained them separately. However, we report the figures for both pedestrian classes as one. Note that for cars and pedestrians, it is not our goal to obtain state-of-the-art performance, we rather demonstrate that such classes can be added without any necessary change to the system's architecture. The scene classifier was trained using 5-fold cross validation on connected subsequences of Seq. DAY.

6 Results

Patch Classifier. To assess the quality of the basic patch classification stage, we report confusion matrices for several variations of the system in Fig. 2, where each row reports how the classifier voted for a specific class. Entries are row-normalized. We compare a purely appearance-based feature set (a) with one that uses additional features from a depth map (b). As can be seen in the lower right quadrant of the matrix, including depth helps classification of object structures, such as cars. However, it does not help in distinguishing cars from pedestrians due to the very local nature of the classification. The effect of including neighborhood constraints via a Markov Random Field (MRF) is shown in (c) (based on the



(a) SUTSU



(b) GIST

1-error			
SUTSU	0.86	0.97	0.84
GIST	0.80	0.93	0.80
precision			
SUTSU	0.88	0.96	0.79
GIST	0.75	0.90	0.65
recall			
SUTSU	0.79	0.85	0.47
GIST	0.72	0.67	0.35

(c) Objects

Figure 4: Confusion matrices for the road types on Seq. DAY using the proposed Segmentation-Based Urban Scene Understanding (SUTSU) and GIST classifiers (a,b). Performance on object classes (c).

output of (a)). While this gives a certain performance improvement (*e.g.*, road structures are less confused with upright structures, sidewalk is recognized better), we will mostly restrict ourselves to the appearance-only classifier without MRF, due to reasons of computational efficiency. A visual example of the three stages can be seen in Fig. 3.

Scene Classifier. In a first quantitative experiment on Seq. DAY, we compare our two-stage classifier with a classifier directly based on GIST features [19], Fig. 4. To compare multi-class performance, we again use a confusion matrix, along with its characteristic numbers of accuracy (AC), defined as the total proportion of correct classifications, and average precision over all classes (AP). For the objects, being binary classifications, we report the number of errors, as well as precision and recall.

As can be seen, both methods manage to tell the first three classes (straight and curves) apart quite well but have problems with all the different types of junctions, which is mostly due to the poor resolution at high distances. Incoming junctions are often mapped to straight streets, where junctions going to the right are recognized slightly better due to the vehicle driving in the right lane. The class "place" is also identified rather reliably, whereas crossings and T-junctions are hardly recognized, with T-junctions often assigned to a "right curve", probably again due to the fact that in Europe we are driving on the right lane, which makes recognizing the left part of the street hard. Also note that the confusion matrices indicate similar problems for both methods, a feature combination would thus not bring much. In general, reasoning works comparably well on global classes (AC: 0.57 (SUTSU) vs. 0.56 (GIST), AP: 0.45 vs. 0.42), also corroborating the effectiveness of GIST as a global scene descriptor. However, SUTSU achieves considerably better performance on object classes, Fig. 4 (c). This is due to the fact that its feature set is largely invariant to the positioning of an object class and also has a direct notion of periodic structures, such as pedestrian crossings. Note again that our goal is not to train an object detector that can localize pedestrians or cars, our system merely detects the existence of the class.

A few example images from both sequences are shown in Fig. 5 and Fig. 6. For each image, we plot the patch classification as well as the scene classification in the lower right corner. The images also show a few results that were obtained by applying SUTSU without retraining either classifier to Seq. DUSK. As can be seen, it performs qualitatively similar. One typical failure case is due to the headlights' reflectance on the car in front, causing false "street marking" patches and hence the flag "pedestrian crossing". Training the patch

Figure 5: Example images from a 30-minute sequence, Seq. DAY. For each image, the bottom left shows the patch classification output, as well as the scene classification (road type, present objects). Result video available at http://www.vision.ee.ethz.ch/~aess/. Figure is best viewed in color.

Figure 6: Example images from Seq. DUSK.

classifier on dusk conditions should alleviate such a problem. The failure can be seen, along with some other typical ones, in Fig. 7. Apart from some obvious mistakes, it is often even difficult for humans to select the right class for an image.

Figure 7: Typical failure cases (from left to right, top to bottom): incoming junctions are ignored due to low resolution; patch classifier mistakes car for pedestrian; headlight's reflectance confuses the patch classifier; sidewalk is difficult to distinguish from road; road geometry not in set of classes; too complicated/ambiguous road geometry.

Performance. The mixed C/C++ and Matlab implementation currently takes about 1 s for the patch classifier (C/C++) and another 1–2 s for the scene classification. Most of the system is parallelizable, and should thus be amenable for real-time implementations.

7 Conclusion

We presented a two-stage method for inner-city street scene classification. Based on a patch-wise scene classification into 13 urban texture classes, we construct a pool of intermediate features that are then use to classify both road typologies, as well as detect the presence of relevant objects. The approach was tested on two challenging sequences and shows that while a state-of-the-art scene classifier can keep global classes such as road types, similarly well apart, a manually crafted feature set based on a segmentation clearly outperforms it on object classes.

We believe that this system offers exciting possibilities for future work. On the one hand, the components of the system can clearly be improved: the texture classifier could benefit from more features, *e.g.* based on 3D points (*cf.* [2]) or optic flow. The scene classifier also could directly include the vehicle's ego-motion as well as trajectory information to reason about the scene. Going beyond vision-based sensors, the fusion with GPS map data is another challenge. On the other hand, the set of classes can be expanded, and the application of the output to tracking or autonomous driving investigated.

Acknowledgements

This project has been funded in parts by Toyota Motor Corporation and the EU projects DIRAC (IST-027787) and EUROPA (ICT-2008-231888).

References

[1] A. Broggi, P. Cerri, P. Medici, P. P. Porta, and G. Ghisio. Real-time road signs detection. In *IVS*, 2007.

[2] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008.

[3] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. In *ICCV*, 2007.

[4] J. P. Da Costa, F. Le Pouliquen, C. Germain, and P. Baylou. New operators for optimized orientation estimation. In *ICIP*, 2002.

[5] DARPA. DARPA urban challenge rulebook. http://www.darpa.mil/GRANDCHALLENGE/docs/Urban_Challenge_Rules_102707.pdf, 2008.

[6] A. Ess, B. Leibe, K. Schindler, and L. van Gool. A mobile vision system for robust multi-person tracking. In *CVPR*, 2008.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html, 2008.

[8] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *CVPR*, 2004.

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70:41–54, 2006. Available from http://people.cs.uchicago.edu/~pff/bp/.

[10] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[11] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:23–37, 1995.

[12] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *IJCV*, 73:41–59, 2007.

[13] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005.

[14] Y. Hel-Or and H. Hel-Or. Real-time pattern matching using projection kernels. *PAMI*, 27:1430–1445, 2005.

[15] D. Hoiem, A.A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005.

[16] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, 2006.

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[18] J. C. McCall and M. M. Trivedi. Video-based lane estimation and tracking for driver assistance: survey, system and evaluation. *ITS*, 2006.

[19] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[20] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[21] F. Schroff, A. Criminisi, and A. Zisserman. Object class segmentation using random forests. In *BMVC*, 2008.

[22] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[23] K. Tieu and P. Viola. Boosting image retrieval. In *CVPR*, 2000.

[24] C. Wojek and B. Schiele. A dynamic CRF model for joint labeling of object and scene classes. In *ECCV*, 2008.