Enabling Learning From Large Datasets: Applying Active Learning to Mobile Robotics

Cristian Dima, Martial Hebert and Anthony Stentz
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Email: [cdima,hebert,axs]@ri.cmu.edu

***Abstract*— Autonomous navigation in outdoor, off-road environments requires solving complex classification problems. Obstacle detection, road following and terrain classification are examples of tasks which have been successfully approached using supervised machine learning techniques for classification. Large amounts of training data are usually necessary in order to achieve satisfactory generalization. In such cases, manually labeling data becomes an expensive and tedious process.**

This paper describes a method for reducing the amount of data that needs to be presented to a human trainer. The algorithm relies on kernel density estimation in order to identify "interesting" scenes in a dataset. Our method does not require any interaction with a human expert for selecting the images, and only minimal amounts of tuning are necessary.

We demonstrate its effectiveness in several experiments using data collected with two different vehicles. We first show that our method automatically selects those scenes from a large dataset that a person would consider "important" for classification tasks. Secondly, we show that the labeling of only few of the images our method selects leads to classification performance that is comparable to the one obtained after labeling hundreds of images from the same dataset.

I. Introduction

Several important aspects of outdoor mobile robotics can be reduced to solving classification problems. Obstacle detection can be seen as the problem of using sensory data to classify regions of space around a robot as traversable or not. Road following is another example in which we need to use the sensors and classify the space around an autonomous vehicle into the road/non-road classes. Developing classifiers that perform well is thus an important part of fielding a successful robotic vehicle.

Some of the more successful classifiers that were designed for these type of problems use machine learning techniques. The outdoor off-road environment is complex and the data produced by the multitude of sensors on a modern robotic platform is often high-dimensional. As a result, manually deriving good algorithms that work well in a multitude of situations can be very challenging.

In 1992, Pomerleau [1] demonstrated the first successful application of machine learning methods to the problem of mobile robot navigation: a neural network used image data in order to choose a steering angle. Learning quickly became a preferred solution for handling the real-world complexity in autonomous vehicle applications (see for example [2]–[5]).

Fig. 1. The two robotic vehicles used for the experiments described in this paper: the CMU autonomous tractor (left) and GDRS XUV (right).

While autonomous vehicles were able to demonstrate good performance in many specific test cases, there are important practical aspects that have not been addressed. Almost all the systems use supervised learning, which require labeled data. For certain problems obtaining labeled data is inherent to the data collection process and is relative cheap (see for example [6]). However, the most common approach is to have a human expert manually label data that is representative of the operating environment. Since the outdoor off-road environment is highly unconstrained, obtaining a system with good generalization properties requires using a large amount of data, which is both expensive and tedious to label.

We believe that in order to make learning for autonomous navigation applicable to real-world problems, the need for manual inspection and labeling of sensor data needs to be significantly reduced. This paper describes an active learning technique that can be applied to large unlabeled datasets in order to select only the "interesting" scenes that should be presented to the human expert for labeling. Essentially, we would like to have a data filter that can take as input datasets of thousands of images (in case we use image data) and only present the human expert with 10-20 images that are really worth labeling. This is the typical application for active learning techniques.

Active learning is a research area that had many success stories (see [7] and [8] for short but informative reviews). Some of the better known applications are related to data mining text information [8] , astronomical data or large company records. Robotics has also seen some important applications, mostly in the control domain ([9], [10]).

Our long term goal is to make learning practical for large,

real-world robotics applications by adapting promising techniques from the data mining field to robotics. The approach we describe in this paper is intuitive, well founded theoretically and produces good results. As we will show later in the paper, this technique can be used by itself or as a first stage of more complex active learning systems that require human interaction.

In section II we describe the type of data that we use and we cover the theoretical foundations of our method. In section III we present some of our experimental results. We describe two types of experiments we performed using data from two different robots. Finally, we conclude in section IV.

II. Approach

Before we present details about our algorithm we briefly describe our application and data.

A. Problem Setup and Data Representation

The technique described in this paper has been applied in the context of terrain classification and obstacle detection for autonomous outdoor robots. In particular, we are using range data from laser range finders, color, infrared and texture information captured using two autonomous robots that will be described in more detail in section III. The sensors on the two autonomous vehicles (color cameras, infrared camera and laser range finders) are calibrated with respect to each other, meaning that under very mild assumptions about the geometry of the scene we can obtain color and infrared information for every three dimensional (3-D) point returned by the laser range finder that is in the field of view of our cameras. Similarly, 3-D points from the laser can be projected into any one of our images. The assumption that needs to be made is that the imaging and range sensors are not far from each other compared to the distance to the imaged scene. Even if this assumption is violated, problems will only occur when occlusions are present.

Both our vehicles can record large amounts of laser and image data. Each time an image is captured, the recently recorded laser data is projected into it. The image is divided into a grid of rectangular patches and several features are extracted from each data modality. Each data log will contain many images, each image will be contain many patches (1200 in our case) and several features (36 in the experiments presented in this paper) will be extracted for each image patch.

Once the features are extracted, classifying the image patches as obstacles/non-obstacles or road/non-road can be achieved with any standard learning algorithm such as neural networks, support vector machines, decision trees, etc. The standard data labeling procedure consists in navigating through the entire data log, manually selecting images that a human expert considers interesting and labeling regions of the images as belonging to specific classes.

Our method is a non-interactive technique that analyzes the features associated with each one of the images in the dataset in order to detect images that are considered "surprising" given the probability distribution of the rest of the data. It is important to note that this is done *before* any data is labeled by the human expert. For this reason we refer to our method as "unlabeled data filtering". This contrasts our method with better known active learning techniques such as confidence based query selection or voting based query selection (see [8], [11]–[13]) which require a small amount of labeled data to begin with and then interactively present more data to the human expert for labeling. Using our method does not however exclude the use of some other interactive active learning technique. On the contrary, our approach can be used to obtain a good small dataset for jump-starting the other interactive methods.

B. Kernel Density Estimation

The core of our method consists in repeatedly estimating the probability density function over the space in which our data points live. Since we had no reason to assume that our data obeyed any particular probability distribution we opted for a non-parametric method such as mixtures of Gaussians or kernel density estimation (KDE). While slower than mixtures of Gaussians, kernel based methods have the advantage that they have less problems with local minima and that roughly only one parameter (the bandwidth of the kernel) needs to be selected. Since our method is running off-line and does not require human interaction speed was not an important factor and as a result we used kernel density estimation.

KDE is probably the best known method for estimating probability density functions non-parametrically, and is covered extensively by most statistics, machine learning or pattern classification books. We will only present here the basics and refer the reader to [11], [14], [15] for excellent discussions on kernel density estimation.

Assuming that N data points $x_1, \ldots, x_N$ from $\mathbf{R}^d$ are available, the KDE estimate for the probability of observing a pattern x when using a Gaussian kernel is given by

$$\hat{p}(x) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{(2\pi h^2)^{d/2}} e^{-\frac{\|\mathbf{x}-\mathbf{x}^n\|^2}{2h^2}} \tag{1}$$

where h is the bandwidth of the kernel. Essentially, this function computes a count of the neighbors of x and weights them through the Gaussian function. Kernel density estimation can use many other kernels other than the Gaussian, but in practice the choice of a particular kernel is not nearly as important as choosing the right bandwidth. The bandwidth is a smoothing parameter and choosing it is related to addressing the well known bias-variance trade-off: a bandwidth that is too small will result in a noisy estimate while choosing one that is too large with result in an over-smoothed, high bias estimate of the probability density function. Our algorithm uses k-fold cross-validation in order to search for the bandwidth that maximizes the likelihood of the data.

One of the serious problems that affect kernel density estimation is high dimensionality. It can be shown (see for example [14], [15]) that as the dimensionality of the input space increases, the likelihood of having any data points close

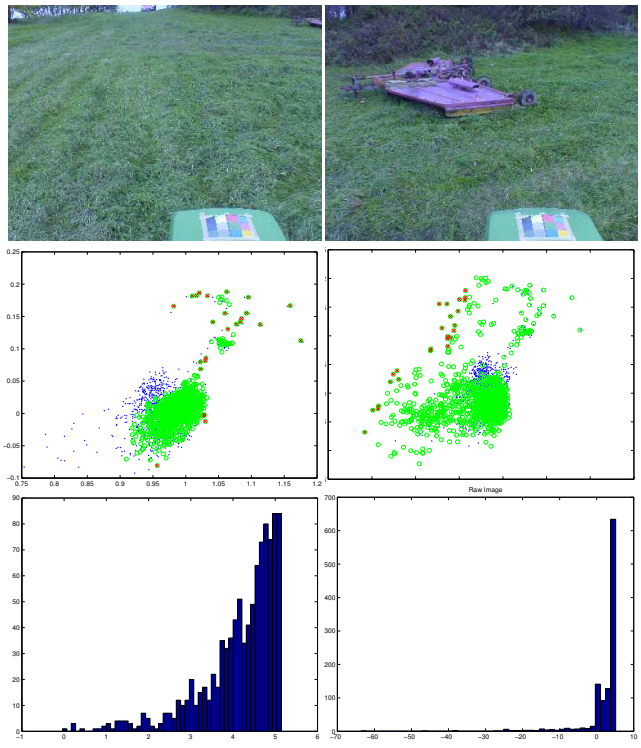

Fig. 2. The basic idea behind our algorithm: a "common" scene on the left column, an "interesting" one on the right. The second row is a 2D representation of the features space (obtained with PCA). The blue (dark) points represent that patches that are already in the selected set S. The green circles and the red crosses represent the data points coming from the currently analyzed scene (the red crosses are the 25 least likely patches). For the common scene there is a lot of overlap between the its patches and the set S, while a significant number of patches from the interesting scene are in currently empty regions of the feature space. As a result, the likelihood of the least likely patches in the interesting scene will be much lower then the one for the common scene. This can be seen on the two histograms on the third row which represent the likelihoods of the patches corresponding to each image estimated based on the current set S.

to a query point decreases dramatically. In our experiments we present two types of results: one in which we only consider a three-dimensional space given by the color features of every patch and one in which principal component analysis (PCA) is used to compress our 36 features down to a more manageable five-dimensional space. We have chosen to use PCA mostly for convenience, but other dimensionality reduction methods can also be used.

C. Unlabeled Data Filtering

To understand how we use kernel density estimation to iteratively select interesting images, we first derive our score measuring for the degree of interest presented by a particular image.

Throughout our algorithm we maintain a set S of the images that are considered "interesting" (selected for labeling) and set U of images that were not selected yet. In the beginning the set S is empty and U contains our entire dataset.

Our algorithm begins by randomly selecting one of the images in U and using it to initialize the pool of interesting images S. The N patches from the selected image represent our initial and somewhat limited knowledge about the content of the dataset. The patches from the images in S can be used with KDE in order to estimate the likelihood of any other patch from remaining images in the set U.

Let us now assume that the image patches of a specific image are independent. This assumption is known not hold for natural images, but similar independence assumptions are frequently made in the image processing field for reasons related to the dimensionality of the data. Note that we do not assume that the pixels inside a patch are independent, but only that there is no correlation between the patches.

In this case, we could express the likelihood of observing image I as a function of the likelihood of its patches as

$$p(I) = \prod_{x \in I} p(x)$$

Note however that we are not necessarily interested in the likelihood of the entire image. An image containing a small pink obstacle on a grassy background is certainly more interesting –from the obstacle detection point of view– than an image containing only grass of a slightly different shade from what we have seen so far. It might be better to estimate the degree of interest of an image only by aggregating the likelihoods of its k least likely to be observed patches. If we denote the set of the k least likely patches by A_k (such that $\forall x \in A_k, y \in I - A_k \implies p(x) < p(y)$) we can re-express the likelihood of the "surprising part of the image I" as

$$p(I_k) = \prod_{x \in A_k} p(x)$$

The score function we are using for the experiments presented in this paper is simply the log of $p(I_k)$:

$$score(I) = log(p(I_k)) = \sum_{x \in A_k} log(p(x)) \qquad (2)$$

Given the set S of already selected images we can sort all the images in the U set based on our score function. We can iteratively select the "most surprising" image in U, add its patches to the set S, reestimate the probability density function and repeat the process for selecting as many images as we are interested in. Intuitively, the method tries to select those images whose patches will populate some of the regions of the feature space that have low density. We will show later that there are heuristic methods for detecting when enough images have been selected to provide good coverage.

III. Experimental Results

In order to test the effectiveness of our approach on real-world problems, we have performed two types of experiments using data from two autonomous vehicles.

In the first experiment we apply our algorithm to four datasets and assess the degree to which the selected images coincides with what a human expert would consider "important" in the original dataset.

In the second experiment we compare the performance of a classifier that is trained on the scenes selected by our filter to the performance of the same classifier trained on the entire dataset. The error rates on a separate test set are used for comparing performance.

A. Datasets

The two robotics vehicles that were used for collecting our datasets are presented in Figure 1. The CMU autonomous tractor is equipped with two Sony DFW-SX900 high resolution (1280x960) digital color cameras, a Raytheon Control IR 2000B near-infrared camera and two mechanically scanned SICK LMS-200 laser range finder units. The XUV is also equipped with color cameras, an infrared camera and a laser range finder. Both vehicles use GPS, encoders and inertial sensors for localization. Note that while the sensors on the two vehicles offer the same sensing modalities, the quality of the data is significantly different and so is the geometrical configuration of the sensors. This encourages us to believe that our experimental findings will apply to other robots and sensor suites.

The tasks we consider are obstacle detection and road following. The datasets we use for obstacle detection were collected with the autonomous tractor on a farm in Hickory, Pennsylvania and at a site close to the Pittsburgh airport that is covered with natural brush. The FARM data (see Figure 3) was collected especially for this experiment, and we tried to capture long sequences of relatively non-interesting terrain with occasional obstacles. The point of this experiment is not to detect extremely challenging obstacles but rather to see if our unlabeled data filtering system would automatically select the images containing obstacles. The vehicle was driven through a grass and weed covered field that contained other agricultural equipment, thicker vegetation that cannot be traversed, a car, a blue tarp and several small green and light grey plant pots (meant to simulate rocks). The dataset contains 900 images recorded at a rate of 2 Hz along with the corresponding position and range data.

The two other obstacle detection datasets contain much more challenging data. The HOLE dataset contains 219 images of a series of approximately 50 cm deep holes with diameters varying from 25 to 50 cm (see Figure 5). The TALLGRASS dataset contains 309 images recorded by driving through very tall weeds (approx. 1.8 m). Twice along the path, the vehicle encounters a person wearing a camouflage jacket hidden in the weeds. Some small trees and brief ares with less weeds are also present (see Figure 6).

The road detection dataset (Figure 4) was collected with the XUV at a test range in central Pennsylvania. This dataset was originally collected to test the limits of our road detection system. The aspect of the road varies significantly over the course of the data log, which contained 440 images recorded at 1 Hz.

For all datasets we extracted color, texture, IR and laser features for each patch. We used 6 color features (mean LUV values and their standard deviations), 24 textures features (FFT based), 2 IR features (mean and standard deviation for each patch) and finally 4 features based on laser data: the height of the points expressed in the vehicle frame and the standard deviations in the vertical, forward and lateral directions. Since kernel density estimation cannot be applied directly in a 36-dimensional input space we have chosen to perform our initial experiments using either the three color means or the result of compressing all the features to a five-dimensional space using PCA.

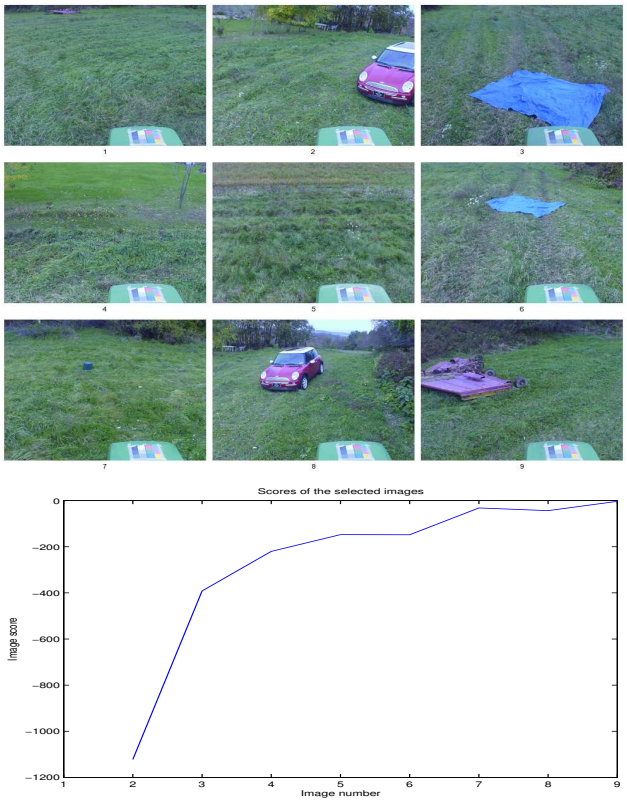

Fig. 3. Results obtained on the FARM dataset using color features. TOP: The top-left corner image is the one selected randomly for initialization. The images selected in future iterations are presented row-wise, in the order of selection. BOTTOM: the scores of the selected images (the minimal score at each iteration).

B. Experiment 1: Scene Selection

In the first experiments we applied the unlabeled data filtering algorithm to our datasets and tried to evaluate if the algorithm selects as important those scenes that a human would find interesting, such as the obstacles images or scenes containing very different types of road.

Figure 3 displays the first 9 images selected from the FARM dataset. The image patches were represented only by their color information in a three-dimensional space. The first image is always chosen randomly and then the algorithm iteratively selects the images with the lowest score according to the metric we defined in Equation 2. As we can see, an image of the car and some non-traversible vegetation and an image of the blue tarp are chosen as the two most informative images given the initial random image that only contained grass.

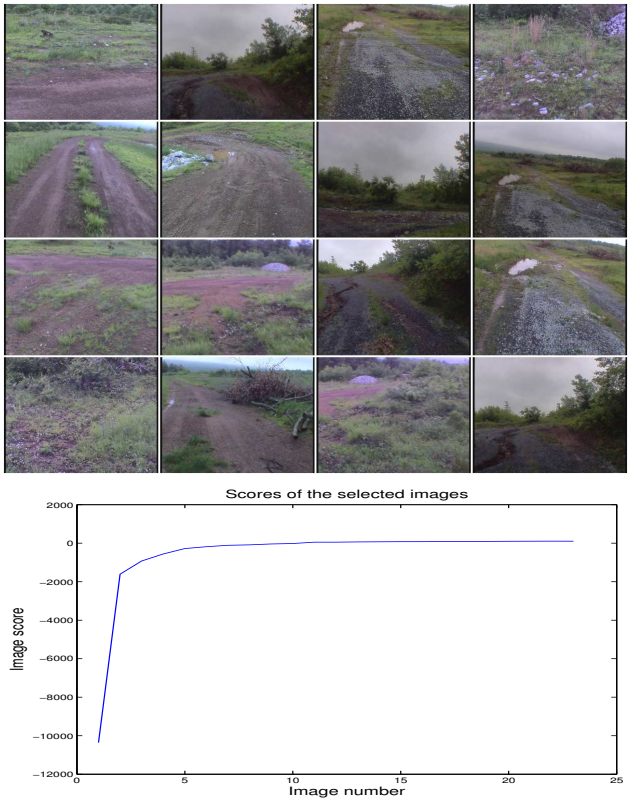

Fig. 4. Results obtained on the ROAD dataset using color features. TOP: The top-left corner image is the one selected randomly for initialization. The images selected in future iterations are presented row-wise, in the order of selection. BOTTOM: the scores of the selected images (the minimal score at each iteration).

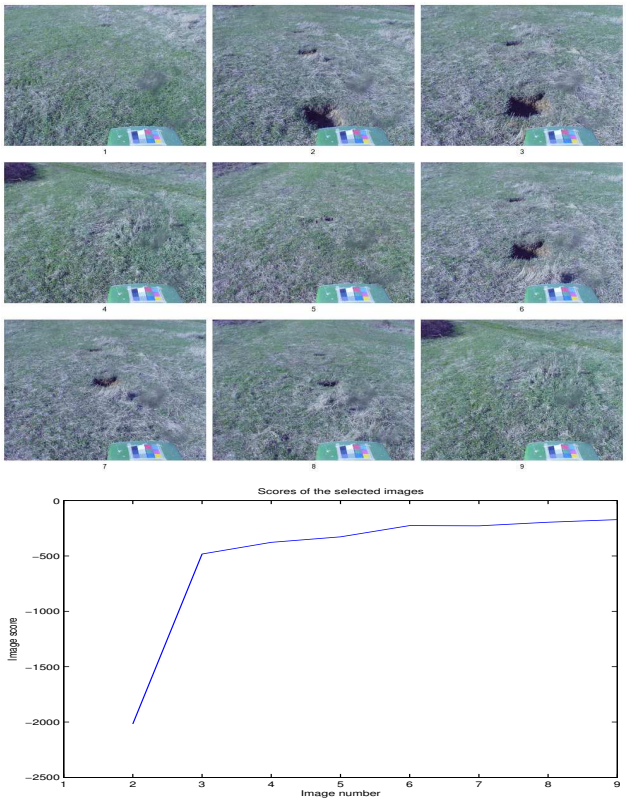

Fig. 5. Results obtained on the HOLE dataset using color, texture, IR and laser features projected to 5D using PCA. TOP: The top-left corner image is the one selected randomly for initialization. The images selected in future iterations are presented row-wise, in the order of selection. BOTTOM: the scores of the selected images (the minimal score at each iteration).

Images 3 and 4 contain views of a mowed lawn and a hay field, which are quite different in aspect from the are we drove in. It is interesting to notice that an image of the implement is only the last one of the 9 we selected; the explanation is that its color is very similar to the color of the car present in image 2.

The plot at the bottom of the figure represents the scores of the images that got selected. The score increases dramatically as the first 5 images are added and then increases at a slower rate, which suggests that the rest of the images in the dataset are a lot less "surprising" considering the distribution of the patches contained in the set S after four iterations of our algorithm. Thus, by looking at the plot of the score one can estimate the minimal number of images that need to be labeled by a human expert in order to obtain good coverage of the input space.

The same type of data is presented for the road detection dataset. Since the interesting road scenes are not as readily identifiable as it was the case with the obstacle set it is harder to estimate if the algorithm selected the "correct" images. However, as we can see in Figure 4, the algorithm selects quite varied and difficult instances of road scenes. The figure was generated using only the color features, and we can notice that the behavior from Figure 3 is repeated: after 4-5 images the color space that is covered by this dataset is adequately populated.

For the experiments on the HOLES and TALLGRASS datasets we have used all of our 36 color, texture, infrared and laser features, projected to a five-dimensional space. This slightly higher dimensional space together with the subtle nature of the obstacles made these two datasets more challenging: in the TALLGRASS dataset the presence of the camouflaged human is quite easy to miss even by a human expert.

The algorithm performed very well, selecting the kind of images one would choose to label in order to perform obstacle detection in those environments. In the HOLES dataset the first 9 images contain holes of various sizes seen at different ranges, and some of the spots where the terrain profile was changing slowly. While it might be hard to see in Figure 6, the first 9 images retrieved from the TALLGRASS dataset contained 4 instances of scenes with the camouflaged human, at different ranges and poses with respect to the vehicles (images 2,3,6,9). The other images represented transitions between short and tall vegetation, and an image dominated by the tractor self-shadow.

The good results obtained on these challenging datasets determined us to investigate if the algorithm is indeed identifying as interesting the same regions of the image that a human expert would find interesting. Looking at the selected images and determining that they contain obstacles does not

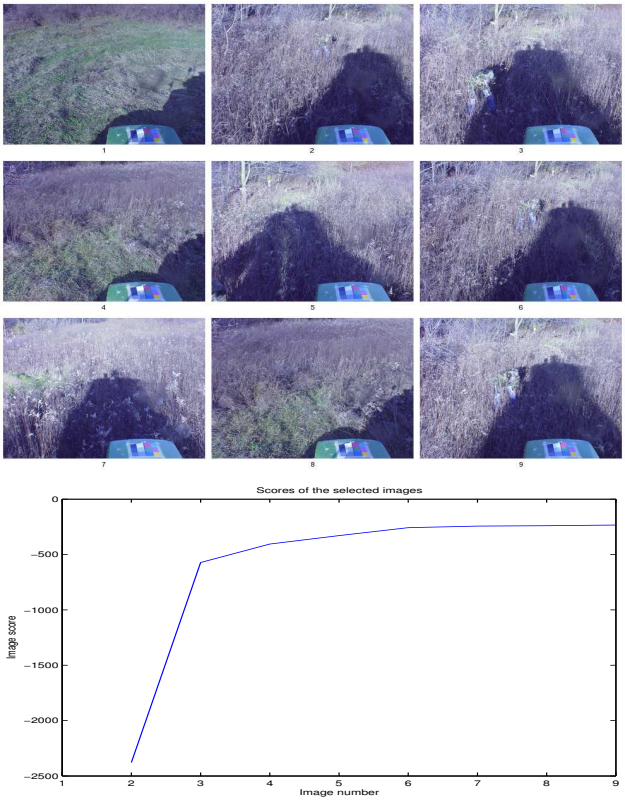

Fig. 6. Results obtained on the TALLGRASS dataset using color, texture, IR and laser features projected to 5D using PCA. TOP: The top-left corner image is the one selected randomly for initialization. The images selected in future iterations are presented row-wise, in the order of selection. BOTTOM: the scores of the selected images (the minimal score at each iteration).

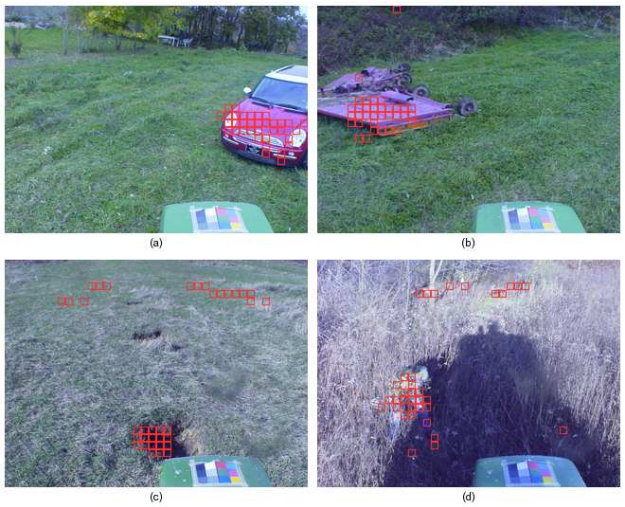

Fig. 7. A representative set of the images selected as by our algorithm, with the locations of the low likelihood patches marked in red squares. The majority of the marked patches are on the hood of the car (a), the implement (b), the hole closest to the vehicle (c) and the camouflaged person in the lower-left side of the image (d).

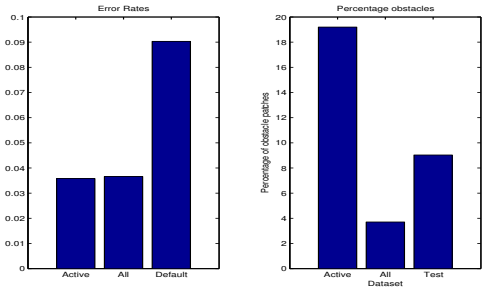

Fig. 8. Experiment 2. **Left:** Error rates of the classifiers trained on the active learning training set ('Active', 5 images) and on the entire dataset ('All', 300 images). 'Default' represents the error rate of hypothetical constant classifier. **Right:** The percentage of obstacle patches in the 5 most important images ('Active'), the 300 image dataset ('All') and the test set.

necessarily mean that our algorithm chose the images because of those obstacles.

In order to eliminate the possibility of a series of misleading coincidences, we have marked the image locations of the 35 least-likely patches used for computing the score of each one of the images that were selected. We were able to verify that they generally corresponded to our human definition of "interesting" regions. Some representative results are presented in Figure 7.

While not providing a quantitative evaluation, the experiments we presented so far indicate that the images that get selected automatically are representative of the environment in which the data is collected. We have also shown that the score curve can be used to decide on the number of images that need to be labeled in order to populate most of the representative regions of the input space.

C. Experiment 2: Classification Performance

Our goal is not just to select images that look interesting, but to actually obtain good classification performance while labeling significantly less data.

In order to verify that the images that were selected can lead to good performance we have performed a simple supervised learning experiment. We have labeled 300 images from our 900 image FARM obstacle detection dataset. We have labeled every third image in the sequence, which –given the speed at which we drove and the frame rate– essentially gave us several consecutive views of each point on the path. We have also labeled the 5 most interesting images selected by our algorithm (the randomly chosen image and the results of 4 iterations of our algorithm). We have chosen this number because according to the score curve, the first 5 images should provide an adequate coverage of the input space for the color features. As a result of this process, we have two labeled datasets: an ALL images one and an ACTIVE learning dataset.

We use the two different sets to train two neural network classifiers that take the three color features as inputs and output obstacle/non-obstacle predictions.

We have applied the two classifiers to a separate test set from the same environment but collected late in the afternoon in very different lighting conditions. The test set was manually

labeled in order to estimate error rates. The performance of the two algorithms is presented in Figure 8. The two classifiers have essentially the same error rate (approx. 3.6%). 9.03% of the test set represents obstacle patches, which means that a hypothetical classifier that would ignore all the data and always predict the most frequent class (non-obstacle) would achieve an error rate of 9.03%.

While these results might seem surprisingly good, the percentage of obstacle patches that are present in each of the training datasets (the 5 images vs. the 300 images) offers an explanation. The large training dataset was very imbalanced (only 3.71% obstacles) which means that most of the dataset contained grass. In contrast, the images selected by our method contained 19.2% obstacle patches; as a result, the classifier that was trained on our 5 image dataset got exposed to enough obstacle data to make good predictions even if its training pool was much smaller.

IV. Conclusion

We have presented a method that can be used to dramatically reduce the data labeling requirement for outdoor classification problems. Our results on datasets from very different environments confirm that the method can be used to automatically "filter" large datasets and retrieve salient images that result in a good coverage of the feature space. More importantly, our preliminary classification test has shown that in certain cases the error rates that result from using 5 informative images can be as good as the rates obtained after labeling an entire dataset of hundreds of images.

These results are an important step toward enabling the use of machine learning for the large scale classification problems that occur in outdoor robotics. Active learning approaches such as the one we describe can help in two ways: they reduce the need for costly labeled data and also reduce the amount of data that needs to be processed by the learning algorithm. As a result, they allow learning to be applied to much larger problems in the robotics field than previously possible.

Our interest in active learning expanded to several other aspects of this research area. While the largest dataset we considered so far (900 images) is already larger than what can be labelled manually , we would like to be scale up to datasets of millions of images. We are interested in using approaches such as the ones described in [16]–[18] in order to improve the speed and robustness with which we can estimate probability density functions. Furthermore, we are currently working on more standard active learning systems that are interactive and could be used as a second stage applied after the algorithm we described here.

Acknowledgment

The authors would like to thank Jeff Schneider, Carl Wellington, Nicolas Vandapel and Herman Herman for their support in various aspects related to this work.

This paper was prepared through collaborative participation in the Robotics Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0012. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

References

[1] D. Pomerleau, "Progress in neural network-based vision for autonoumous robot driving," in *Proceedings of the Intelligent Vehicles '92 Symposium*, 1992, pp. 391–396.

[2] P. Belluta, R. Manduchi, L. Matthies, K. Owens, and A. Rankin, "Terrain perception for DEMO III," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, October 2000, pp. 326–331.

[3] M. Rosenblum and B. Gothard, "A high fidelity multi-sensor scene understanding system for autonomous navigation," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, October 2000, pp. 637–643.

[4] T.-H. Hong, T. Chang, C. Rasmussen, and M. Shneier, "Feature detection and tracking for mobile robots using a combination of ladar and color images," in *Proceedings of the 2002 IEEE International Conference of Robotics and Automation*, Washington, D.C., May 2002, pp. 4340–4345.

[5] C. Rasmussen, "Combining laser range, color and texture cues for autonomous road following," in *Proceedings of the 2002 IEEE International Conference of Robotics and Automation*, Washington, D.C., May 2002, pp. 4320–4325.

[6] C. Wellington and A. Stentz, "Learning predictions of the load-bearing surface for autonomous rough-terrain navigation in vegetations," in *Proceedings of the International Conference on Field and Service Robotics*, July 2003.

[7] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," in *Artificial Intelligence*, vol. 97, no. 1-2, 1997, pp. 245–271.

[8] N. Roy and A. McCallum, "Toward optimal active learning through monte carlo estimation of error reduction," in *Proceedings of the International Conference on Machine Learning*, June 2001.

[9] A. Moore, "Efficient memory-based learning for robot control," Cambridge, UK, October 1990.

[10] ——, "Fast, robust adaptive control by learning only forward models," in *Advances in Neural Information Processing Systems*, J. E. Moody, S. J. Hanson, and R. P. L, Eds. San Francisco, CA: Morgan Kaufmann, April 1992.

[11] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley and Sons, Inc., 2001.

[12] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., 1994, pp. 3–12.

[13] H. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the Fifth Workshop on Computaional Learning Theory*, San Mateo, CA, 1992, pp. 287–294.

[14] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1997.

[15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. Springer-Verlag, 2001.

[16] A. Gray and A. Moore, "Rapid evaluation of multiple density models," in *Artificial Intelligence and Statistics*, 2003.

[17] ——, "'N-Body' problems in statistical learning," in *Advances in Neural Information Processing Systems*, T. K. Leen and T. G. Dietterich, Eds. MIT Press, 2001.

[18] A. Moore, "Very fast EM-based mixture model clustering using multiresolution KD-trees," in *Advances in Neural Information Processing Systems*, M. Kearns and D. Cohn, Eds. San Francisco, CA: Morgan Kaufman, April 1999, pp. 543–549.