Terrain Classification for Off-Road Driving
CS-229 Final Report

Kelly Shen, Michael Kelly, Simon Le Cleac'h
Stanford University
{kshen21,mkelly2,simonlc}@stanford.edu

1 Introduction

In recent years there has been a significant amount of research done on the use of computer vision for autonomous driving. However, this research has largely focused on on-road driving: there has been significantly less work focused on the off-road setting. In this setting, accurate terrain classification is crucial for safe and efficient operation.

The work that has been done on this topic has largely focused either on the use of non-vision sensor modalities for terrain categorization (Stavens and Thrun, 2012; Thomas, 2015) or on the use of classical computer vision techniques, particularly those employing local feature descriptors (Filitchkin and Byl, 2012; Khan et al., 2011). Our project will focus purely on the use of monocular vision for terrain classification, and, by using more modern techniques such as convolutional neural networks, we hope to create a more performant and flexible terrain classification system.

We first focus our efforts on the task of classifying images containing only a single terrain type. Our convolutional neural network takes as input small (100x100 pixel) images containing only one of six terrain categories and outputs a terrain label. We compare the performance of this model against that of an SVM classifier, which uses features obtained from a SURF-based Bag of Visual Words (BoVW) model.

The purpose of a terrain categorization system for off-road driving is to process images containing multiple terrain types, however. For this reason, we perform a brief investigation into the applicability of our homogeneous terrain classifiers to the mixed terrain setting. Specifically, we use a sliding window algorithm inspired by that of (Filitchkin and Byl, 2012) together with our trained classifiers to label individual pixels in mixed terrain images. Our approach generates visually intuitive results, suggesting that this approach might be a fruitful topic for further research.

2 Related Work

Prior work on the problem of terrain classification has largely been split between remote sensing applications (e.g. (Paisitkriangkrai et al., 2015; Delmerico et al., 2016)) and ground-based applications (e.g. (Filitchkin and Byl, 2012; Brooks and Iagnemma, 2012; Walas, 2015)). Research into ground-based terrain classification can be further divided by sensor modality. Non-vision modalities such as lidar (Thomas, 2015) and proprioceptive sensing (Brooks and Iagnemma, 2012; Stavens and Thrun, 2012) have been successfully used in autonomous off-road vehicles and planetary rovers. However, vision - in particular, monocular vision - offers a number of appealing advantages over other sensing modalities, such as low weight, power consumption, size, and cost, as well as high information content and excellent range (Engel et al., 2012).

Prior work on ground-based terrain classification using monocular vision has largely focused on the use of local feature descriptors such as local binary patterns (Khan et al., 2011) and SURF features (Filitchkin and Byl, 2012). (Khan et al., 2011) test a random forest classifier and a number of different local feature descriptors to classify images as one of five terrain types: gravel, asphalt, grass, big tiles, and small tiles. They employ a coarse grid-based approach to categorize multi-terrain images, however, which gives little insight into the performance of their system at the finer-grained levels useful for path-planning and control in the off-road setting.

(Filitchkin and Byl, 2012) employ an SVM classifier and a SURF-based Bag of Visual Words model to classify images of homogeneous terrain into six categories: asphalt, grass, gravel, mud, and wood-chips. They use a sliding window algorithm to label multi-terrain images at a much finer scale than (Khan et al., 2011). However, the images used to train and test their classifier were not taken while in motion, and the classification performance they report is thus likely to be overly optimistic when applied in the off-road driving setting, where vehicular motion is likely to cause some blurring of the images.

Our approach to vision-based terrain classification seeks to avoid these drawbacks. Given the marked improvement demonstrated by convolutional neural networks over classical computer vision techniques on many image classification benchmarks (Sharif Razavian et al., 2014; Krizhevsky et al., 2012), we also seek to achieve improved classification accuracy through the application of a CNN to this task.

3 Dataset and Features

We collected images of the six categories of terrains and textures relevant for driving in off-road and mixed on-road/off-road settings: dirt, grass, pavement, green vegetation, dry vegetation, and bark. Roughly 12,000 image frames were extracted from multiple videos taken of each individual terrain texture found at the Stanford Dish and at Lake Lagunita. Although the videos were recorded at a frequency of 30 Hz, we decided to extract frames at a frequency of 6 Hz, choosing only one frame out of five to avoid redundancy. The videos were taken while in motion and in partly cloudy conditions, ensuring that the dataset contained images of each terrain type in a variety of lighting conditions (e.g. alternating bright sunlight and shadow) and from a number of different perspectives. To clean our dataset, we excluded exceptionally blurry videos and videos showing inconsistent terrain / terrain that did not fit into one of our categories. To make our dataset exploitable by the convolutional neural network in a reasonable amount of time, we downsampled the frames to 100x100 pixels. This choice was the result of considering a tradeoff between the tractability of the CNN training and the identifiability of each type of terrain from the downsampled images.

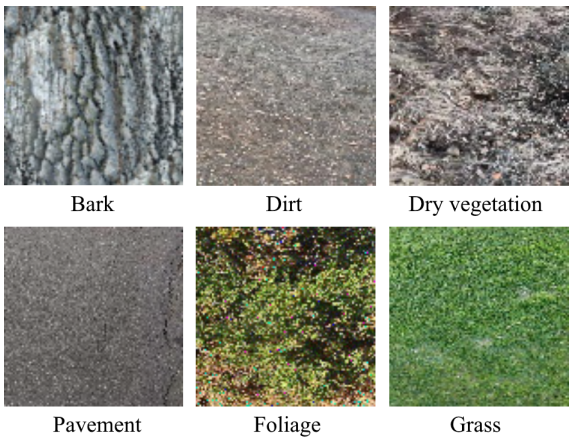

Figure 1: Samples from the 6 categories of terrains and textures downsampled to 100x100 pixel.

The second part of the data collecting process was taking pictures of mixed-terrain landscapes from the perspective of an off-road vehicle (e.g. an image of a paved road running through dry vegetation, with green vegetation in the background). These images will be fed into the mixed-terrain categorization system implemented using the sliding window algorithm, and the output will be visually assessed. These landscape images were taken from the same areas and under the same conditions as the homogeneous terrain dataset to reduce data mismatch.

The homogeneous terrain dataset was divided into training, validation, and test sets using an 60/20/20 split. A number of new training examples were then generated via various transformations of the original training examples, such as rotations, reflections, brightness/contrast adjustments, and the addition of small amounts of Gaussian noise. Our baseline model and CNN were then trained on the augmented set of these new images combined with the initial training set. This produced a total of 72150 training samples, 2408 validation samples, and 2408 test samples.

For the baseline SVM model, the images were converted to grayscale images and then expressed as 125-feature vectors representing a 125-visual word vocabulary. We determined the visual words by using the Bag of Visual Words model (BoVW) and clustering the speeded up robust features (SURF) computed on the training set. This process is described in further detail in Section 5.1. For the CNN, we inputted the normalized 100x100x3 color images without prior feature selection.

4 Methods

We employ a convolutional neural network to classify images from six homogeneous terrain textures: dirt, grass, pavement, green vegetation, dry vegetation, and bark. The performance of this CNN on this classification task is evaluated relative to that of a SURF-based bag of visual words (BOVW) model employing an SVM classifier. Specifics of each method are described in the following sections.

A central step in CNNs is that of convolution, which slides a $m*m*3$ filter w across the input volume. The filter is convolved with each subsection of the $N*N*3$ volume to produce a $(N-m+1)(N-m+1)$ 2D activation map giving the response of the filter at each spatial position. In other words, each unit x in the next layer l is derived as

$$x_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} w_{ab} x_{(i+a)(j+b)}^{l-1}$$

While the central focus of our project is on homogeneous terrain classification, we also examine the utility of our classifier as a component of a larger, heterogeneous terrain categorization system, which uses a sliding window algorithm and the homogeneous terrain classifier to label pixels in a multi-terrain image. While there is no straightforward performance metric for heterogeneous terrain classification (Filitchkin and Byl, 2012), we can establish some sense of the practical utility of our terrain classifier by the extent to which this system generates visually intuitive results.

5 Experiments/Results/Discussion

5.1 Baseline SVM Model

Much of the prior work on visual terrain classification has relied on local feature descriptors (Filitchkin and Byl, 2012; Khan et al., 2011). As such, we chose to use a Bag of Visual Words (BoVW) model with speeded up

robust features (SURF) and a support vector machine (SVM) classifier as our baseline. The Bag of Visual Words model represents an image as a multiset or "bag" of various "visual words" or image features. Each image contains some subset of the overall "vocabulary" of image features, and can be represented as a feature vector of fixed size (the size of the vocabulary), where each entry of the feature vector corresponds to the frequency of the corresponding word in the "bag"-representation of the image. (Bosch et al., 2007)

The visual vocabulary used was generated using SURF features, which are an extension to SIFT (scale-invariant feature transform) features; the SURF algorithm detects key points at unique locations in a grayscale image and then represents them in either a 64- or 128-dimensional feature vector (Khan et al., 2011). A standard approach to SURF-based BoVW involves clustering all SURF descriptors for all images in the training set using k-means, and then using the cluster centroids returned by k-means as the visual vocabulary. We generate the "bag"-representation of an image by adding to the "bag" the associated visual word (i.e. the assigned cluster centroid label given by k-means) for each SURF descriptor of the given image (Bosch et al., 2007).

64-element SURF descriptors were generated using openCV 3.3.0 (Bradski, 2000) for all images in the augmented training set and validation set using a Hessian threshold of 300. Mini-batch k-means was then used to cluster the SURF descriptors from the training set, generating a vocabulary of 125 words (i.e. the 125 centroids returned by k-means). After generating fixed-size feature vectors for all images using this vocabulary, the training data were then used to fit a multiclass SVM using scikit-learn (Pedregosa et al., 2011). Several kernels and regularization parameters were examined and the best performance on the validation set was achieved with a radial basis function kernel and regularization parameter of 1.0. The confusion matrices for the training and test sets are displayed in Figures 2 and 3, while F1 scores across the training, validation, and test sets are displayed alongside those of the CNN in Table 1 in Section 5.2.

The class-specific accuracies for the test set were 0.939 (bark), 0.728 (dirt), 0.815 (dry vegetation), 0.830 (green vegetation), 0.580 (grass), and 0.771 (pavement). While a number of these accuracies are significantly lower than the average test-set accuracy of 90% achieved in prior work using SURF features for terrain classification (Filitchkin and Byl, 2012; Khan et al., 2011), it is worth noting that our model was trained on significantly smaller images (100x100 pixels vs. 320x320 pixels and 640x480 pixels in (Filitchkin and Byl, 2012) and (Khan et al., 2011) respectively). Furthermore, unlike (Filitchkin and Byl, 2012), our training and validation images were captured while in motion, which accords with the intended application of this work (off-road driving) but caused some blurring of the images. Despite the removal of the blurriest images, small but noticeable blurring is still present in a portion of the remaining dataset, which can be expected to generally decrease the performance of SURF (Filitchkin and Byl, 2012). Considering these factors, we concluded that this model serves as a reasonable baseline against which we can compare the performance of our CNN.

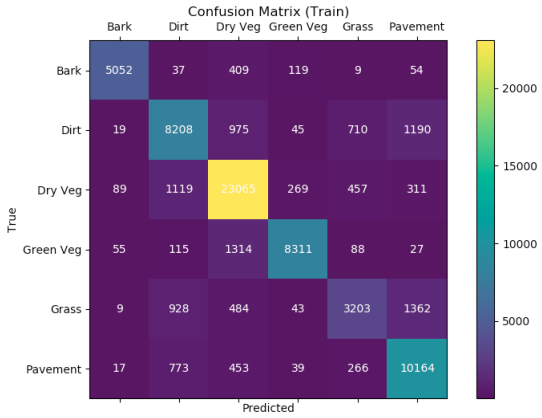

Figure 2: Confusion matrix for the BoVW-SVM model on the training set.

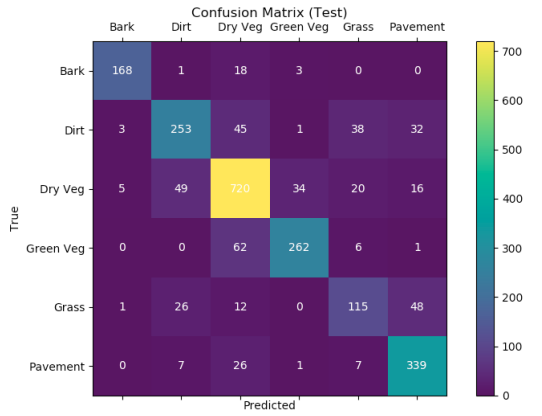

Figure 3: Confusion matrix for the BoVW-SVM model on the test set.

5.2 Convolutional Neural Network Model

Convolutional Neural Network (CNN) classifiers are known for their effectiveness in image classification tasks, due to how filters act as feature detectors that preserve the spatial relationship between pixels and mimic how humans process visual input. Each convolution layer hierarchically learns a filter that detects some construct (e.g. edges in the first layer, then combining edges and corners to higher level shapes in the second layer). On a high level, CNN's include four main operations: convolution, pooling / sub sampling, non-linearity, and classification / fully connected.

The CNN model was built with Keras using Theano

backend (Chollet et al., 2015; Theano Development Team, 2016), and took as input normalized RGB pixel values and one-hot encoded labels. The images were then fed to the first hidden layer, a convolutional layer with 32 feature maps, 5x5 filters, and ReLU activation. Next was a max pooling layer that subsampled each 2x2 window. A dropout layer followed which randomly excluded 20 percent of neurons to reduce overfitting. The matrix was then flattened and fed into a fully connected layer with 128 neurons and ReLU activation. Finally, the output layer used softmax to derive the probability for each of the 6 classes. The model was trained using cross-entropy loss and the Adam optimization algorithm, and fit with a batch size of 200 over 10 epochs. The confusion matrices for the training and test sets are displayed in Figures 4 and 5, while F1 scores across the training, validation, and test sets are displayed alongside those of the SVM in Table 1.

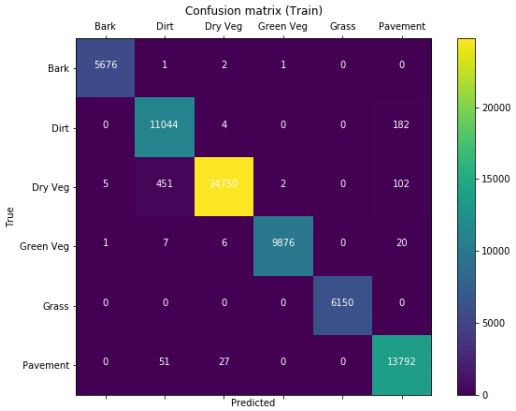

Figure 4: Confusion matrix for the CNN model on the training set.

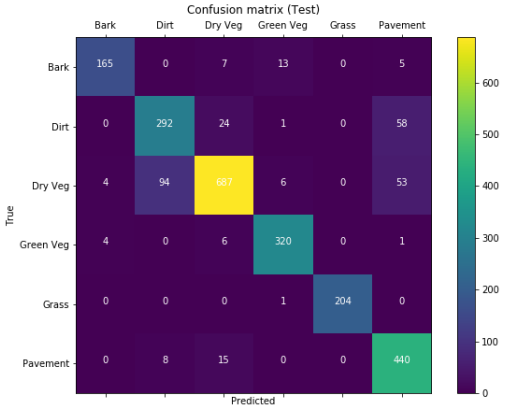

Figure 5: Confusion matrix for the CNN model on the test set.

The model concluded with a 97.70% train accuracy, 88.62% validation accuracy, and 87.54% test accuracy. The gap between train and test accuracy brings up the question of overfitting; an attempt to mitigate this was completed through reducing training iterations from 10 to 8 epochs, which resulted in 95.16% train and 86.25% test accuracy, failing to decrease the accuracy difference. Further inspection brought to attention that validation accuracy during training plateau's rather quickly, by around the fourth to fifth epoch (although minorly jumping around thereafter). This may be a sign that the optimization algorithm or objective is the culprit, and a different model (e.g. different filter / pool sizes, different number layers) with more rigorous hyperparameter search is necessary to improve performance. The current parameters were chosen through a brief and relatively ad-hoc process of comparing validation accuracies on ball-park parameter changes.

F1 Scores: SVM / CNN			
Terrain	Train	Validation	Test
Bark	92.7 / 99.9	91.8 / 89.1	91.4 / 90.9
Dirt	76.2 / 97.0	74.3 / 80.5	73.0 / 75.9
Dry Veg	89.3 / 98.8	84.6 / 87.4	83.9 / 86.8
Green Veg	89.1 / 99.8	84.0 / 95.4	83.5 / 95.2
Grass	63.8 / 100.0	64.4 / 100.0	61.9 / 99.8
Pavement	83.2 / 98.6	84.2 / 87.5	82.0 / 86.2

Table 1: F1 scores for homogeneous terrain classifiers.

The final CNN trained was one layer deep; larger models with more layers or larger inputs were ruled out simply due to local computational constraints. Despite the simplicity of the model, its performance overall is superior to the SVMs as evident in the generally lower level of confusion, particularly in the grass and green vegetation classes. Both models appear to struggle with correctly classifying dirt; the CNN in particular appears to perform weakest deciding between dirt, dry vegetation, and pavement, but is rather consistent in classifying grass, pavement, and bark.

5.3 Application to Mixed-Terrain

We employed a sliding window algorithm to label individual pixels in mixed terrain images using our homogeneous terrain classifiers. At each iteration of the algorithm, a 100x100 pixel patch of the mixed terrain image is fed to the classifier, which returns a label. Each pixel within the current patch receives a single vote for that label. The patch is then shifted slightly, and the process is repeated. After the entire image has been covered, each pixel is classified according to the category for which it received the most votes.

The images in Figure 6 suggest that the CNN's superior performance on the homogeneous classification task translates to the mixed-terrain processing task, generating smoother and more visually intuitive labeled images. Furthermore, the CNN appears to be more robust to various sources of noise, such as the shadow across the road in the second image.

Both classifiers have difficulty distinguishing be-

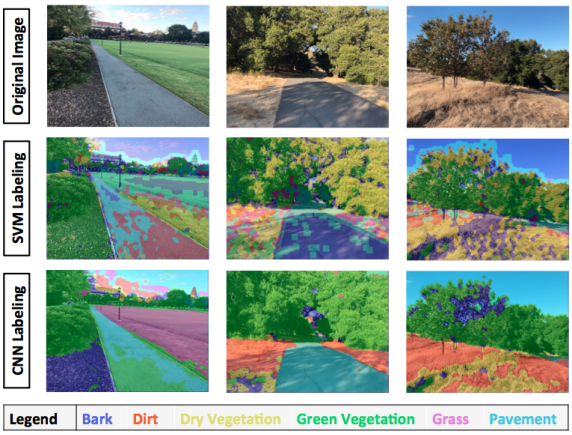

Figure 6: Multi-terrain images categorized using the sliding window algorithm paired with the SVM and CNN classifiers.

tween dry vegetation and dirt, which is unsurprising given the marked similarity between the two categories at the 100x100 pixel scale. The SVM additionally struggles to identify grass (as expected, given the poor test accuracy it achieved in this category), likely due to the color insensitivity of SURF features.

6 Code

Github repository containg all code available here: 229-terrain-classification

7 Conclusion/Future Work

In the homogeneous texture classification task, the CNN model exhibits superior performance compared to the BoVM-SVM model despite its shallow architecture. When adapted to the heterogeneous terrain classification task mirroring landscapes seen during off-road driving, the CNN model successfully maps out a generally accurate depiction of terrain boundaries, while the SVM model produces noisier and less intuitive outputs.

Future work includes several potential steps. First, training the homogeneous classifier on higher-resolution textures to enhance the distinctness of each terrain type. The 100x100 pixel inputs lose important detailed differences between classes like dirt and dry vegetation that exhibit similarities even at full resolution. Second, both models would benefit from a more rigorous hyperparameter search not constrained by time or computational power. In particular, the CNN model would likely improve in performance through adding in subsequent layers combined with larger input images. Third, the mixed-terrain categorization task requires a quantitative and task-specific evaluation metric, as the current metric of visual intuitiveness cannot transfer appropriately to evaluating off-road driving performance. Finally, the overall classification task could be bootstrapped with external training images (such as Google Image results of various terrain types) and tested on mixed-terrain landscapes taken from terrains independent of those in the training set. The current task does not generalize to terrains outside of the selected areas where data was collected.

8 Contributions

Simon: Data Processing (removing blurs, extracting frames, subsampling images)
Michael: SVM Implementation, Mixed Terrain Implementation, Data Processing (augmenting training set)
Kelly: CNN Implementation, Mixed Terrain Implementation
All: Proposal, Data Collection, Milestone, Poster, Final Report

References

Anna Bosch, Xavier Muñoz, and Robert Martí. 2007. Which is the best way to organize/classify images by content? *Image and vision computing* 25(6):778–791.

G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* .

Christopher A Brooks and Karl Iagnemma. 2012. Self-supervised terrain classification for planetary surface exploration rovers. *Journal of Field Robotics* 29(3):445–468.

François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

Jeffrey Delmerico, Alessandro Giusti, Elias Mueggler, Luca Maria Gambardella, and Davide Scaramuzza. 2016. "on-the-spot training" for terrain classification in autonomous air-ground collaborative teams. In *International Symposium on Experimental Robotics*. Springer, pages 574–585.

Jakob Engel, Jürgen Sturm, and Daniel Cremers. 2012. Camera-based navigation of a low-cost quadrocopter. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, pages 2815–2821.

Paul Filitchkin and Katie Byl. 2012. Feature-based terrain classification for littledog. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, pages 1387–1392.

Yasir Niaz Khan, Philippe Komma, Karsten Bohlmann, and Andreas Zell. 2011. Grid-based visual terrain classification for outdoor robots using local features. In *Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2011 IEEE Symposium on*. IEEE, pages 16–22.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. pages 1097–1105.

Sakrapee Paisitkriangkrai, Jamie Sherrah, Pranam Janney, Van-Den Hengel, et al. 2015. Effective semantic pixel labelling with convolutional networks and conditional random fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. pages 36–43.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. pages 806–813.

David Stavens and Sebastian Thrun. 2012. A self-supervised terrain roughness estimator for off-road autonomous driving. *arXiv preprint arXiv:1206.6872* .

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints* abs/1605.02688. `http://arxiv.org/abs/1605.02688`.

Judson JC Thomas. 2015. *Terrain classification using multi-wavelength LiDAR data*. Ph.D. thesis, Monterey, California: Naval Postgraduate School.

Krzysztof Walas. 2015. Terrain classification and negotiation with a walking robot. *Journal of Intelligent & Robotic Systems* 78(3-4):401.