

Image Classification for Ground Traversability Estimation in Robotics

R. Omar Chavez-Garcia, Jérôme Guzzi,
Luca M. Gambardella, and Alessandro Giusti

Dalle Molle Institute for Artificial Intelligence (IDSIA),
USI-SUPSI,
Lugano, Switzerland
omar@idsia.ch

Abstract. Mobile ground robots operating on uneven terrain must predict which areas of the environment they are able to pass in order to plan feasible paths. We cast traversability estimation as an image classification problem: we build a convolutional neural network that, given a square 60×60 px image representing the heightmap of a small 1.2×1.2 m patch of terrain, predicts whether the robot will be able to traverse such patch from bottom to top. The classifier is trained for a specific robot model, which may implement any locomotion type (wheeled, tracked, legged, snake-like), using simulation data on a variety of training terrains; once trained, the classifier can be quickly applied to patches extracted from unseen large heightmaps, in multiple orientations, thus building oriented traversability maps. We quantitatively validate the approach on real-elevation datasets.

Keywords: image classification, traversability estimation, robotics, convolutional neural networks, simulation

1 Introduction

A defining feature of autonomous mobile robots is their ability to plan a path to reach a given target position. In most indoor scenarios, the environment is trivially partitioned in traversable areas (clear floor) and obstacles (i.e. walls, objects, and other areas through which the robot can't pass); once traversable and non-traversable areas are known, path planning is solved using well-known algorithms [1].

In scenarios with uneven terrain, such as outdoors, segmenting the environment in traversable and non-traversable areas is not as trivial. For example, some areas may be traversable only in a few specific directions: a wheeled robot with limited power could be able to descend but not to ascend a steep slope; a powerful, long and narrow robot with an high centre of mass could be able to traverse the same environment both uphill and downhill, but may capsize if traversing perpendicularly to the steepest direction; a bicycle can hop up a side-walk as long as it is proceeding more or less perpendicular to the step, but

will crash when approaching the step from an oblique angle. Moreover, criteria for traversability may not be intuitive or easy to model a priori: a legged robot may be able to negotiate very challenging terrain but get stuck on flat ground with deep holes of comparable size as its feet ; vacuum cleaner robots get stuck over power cords laying on the ground; a car with a low clearance may get stuck on a speed bump.

We consider the problem of estimating where and in which directions a given 3D terrain is locally traversable by a specific ground robot, using a general approach based on machine learning that applies regardless on the robots locomotion method (wheeled, tracked, legged, snake-like), physical characteristics (size, motor torque), and low-level controller (anti-skid algorithms for wheeled robots, foothold selection and gait selection algorithms for legged robots). In the following we define that a given position of a terrain is *locally traversable* in a given direction if the robot, placed in such position and direction, can proceed straight for at least a short distance when driven by its low-level control algorithms.

Traversability is only affected by the characteristics of the terrain around the robot's position: therefore, for a given robot pose X^{robot} with position p and orientation θ , we consider as input a heightmap patch centred in p and rotated in such a way that the robot is pointing towards the top of the patch. This information is a spatial grid of height values, which can be interpreted as a grayscale image: our task is cast as a binary image classification problem (with classes *traversable* vs *non-traversable*), and solved by training a simple convolutional neural network.

Training data for the classifier is generated by simulating the robot on many predetermined training terrains, some of which represent synthetic scenarios (with a variety of obstacles such as ramps, steps, bumps, with different characteristics) and some of which represent realistic rugged terrains; during such simulations, the robot is spawned in random positions and orientations, and instructed to proceed straight ahead; its progress is monitored and instances for traversable (where the robot successfully proceeds) and non-traversable (where the robot can't proceed) heightmap patches are continuously recorded. Once the model is learned from such training data, it can be applied densely on any large unseen heightmap. Because accurate physical simulation is expensive, evaluating the classifier on many points and orientations of a test terrain is orders of magnitude faster than simulating the robot on these poses.

This approach only requires that an accurate simulation model for the target robot is available, and generates an ad-hoc classifier that only applies to such robot. If the training terrains are sufficiently representative and varied, the resulting classifier captures the specific characteristics of the robot, including the maximum steepness that can be negotiated and many non-trivial aspects (such as the risk of getting stuck over bumps).

The **main contribution** of this paper is this novel approach for traversability estimation based on image classification, detailed in Section 3. Experimental

validation and results are described in Section 4. Limitations and extensions of our approach are discussed in Section 5.

2 Related Work

From the early days of mobile robotics, the task to estimate terrain traversability has been of central importance. Robotics literature considers traversability to be an *affordance*, that is an action that a robot can perform on the environment [2]. Estimating traversability is a fundamental capability for many animals and for autonomous mobile robots, because most of their actions depend on their mobility. In robotics, using a simplified (possibly learned) model to estimate traversability is a common approach, because modelling the terrain, the robot and their interaction accurately is difficult and expensive. Several approaches have been proposed to measure traversability and to gather training data [3]: For instance, a robot may label a terrain as difficult to traverse when sensing excessive vibrations [4,5]; human experts can provide clues like preferred paths in a given terrain that avoid possible non traversable regions [6]; the robot may learn a traversability classifier on the spot, directly from experience.

The outdoor robotics community has researched traversability estimation from a variety of sensing sources, such as frontally mounted stereo cameras [7], thermal cameras used to infer soil density [8], laser scanners [9], and time of flight cameras [10].

A common approach to estimate traversability consists of two phases. First, from local sensory data, an elevation map is derived, which is a convenient spatial representation for ground robots [11]. Then, traversability estimates of parts of the elevation map are computed from geometrical features like *slope*, *roughness*, and *step height* using pre-modelled functions specific to the robot’s locomotion and size [12].

Another major approach relies on the use of visual texture to classify the terrain type, e.g. discriminating between rock, sand, and grass; then it derives a traversability score out of the assigned label [4]. Terrain classification is a classical application for supervised machine learning techniques based on generic visual features, transformation-invariant descriptors, texture features, or convolutional neural networks (CNNs) [13,14], which learn to extract problem-specific visual features. DARPA’s project *Learning Applied to Ground Robots project* [15] has advanced learning techniques for terrain traversability classification, including some applications of deep learning [16,17].

3 Traversability Estimation

Figure 1 illustrates the proposed pipeline for traversability estimation. In the following sections, we describe how the simulation environment is set-up (Section 3.1), detail the process for generating training and evaluation datasets (Section 3.2), and finally describe our classification approach (Section 3.3).

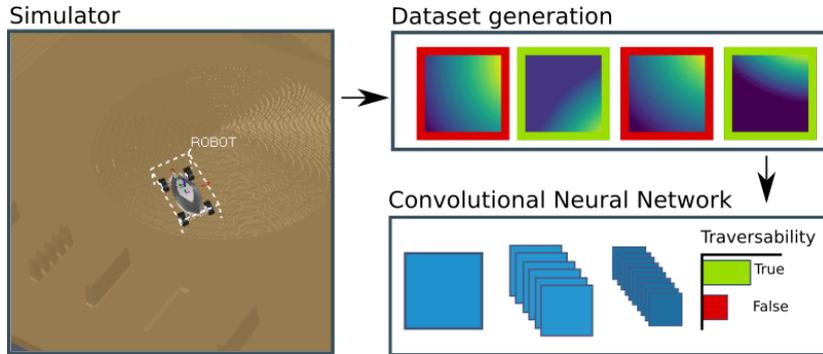


Fig. 1. We run simulations with a given robot model on synthetic terrains (left) to generate datasets linking heightmap patches with their traversability (top right); on these datasets we train and evaluate classifiers to estimate the probability that a given heightmap patch is traversable or not

3.1 Traversability from Simulation

We employ the V-REP simulator, which uses the Bullet physics engine [18] for accurate physical simulation. We simulate a differential-wheeled rear-traction rover with size $30 \times 82 \times 27$ cm (illustrated in Figure 2) moving forward at constant velocity on an uneven terrain whose shape is determined by an input heightmap. The robot’s trajectory on the terrain is captured to extract traversability information.

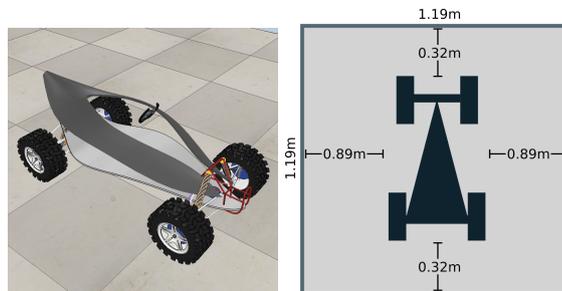


Fig. 2. Robotic platform (left) and its visual signature over a heightmap patch (60×60 px, approx 1.14×1.14 m). The patch is marked as traversable if the robot reaches its top edge within 1 second.

Heightmaps are grayscale images that represent elevation data from lowest (black) to highest (white). Figure 3 shows some examples together with 3D renderings of the respective surfaces. The leftmost heightmap was manually designed to include a variety of obstacles and terrain changes. The rest of the

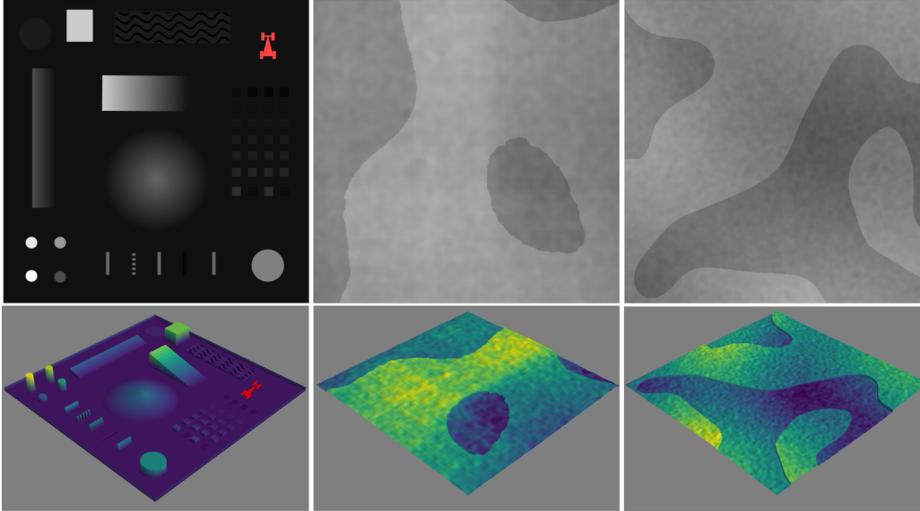


Fig. 3. Examples of heightmaps for generating traversability datasets. Top row shows a 2D top-view of the heightmaps in grayscale where white indicates maximum height. Bottom row shows the corresponding surfaces coloured by height.

heightmaps were randomly generated as a sum of Perlin noise and a curved step surface¹. The size of each generated heightmap is 512×512 px that, when simulated, are scaled to represent surfaces of 10×10 m (≈ 1.2 cm/px resolution) with a maximum height difference of 1m.

Once the simulation is initialized with a surface from a heightmap, the robot is set to a random pose (position and orientation) on the map and moves forward at constant velocity without steering. After the robot reaches the edge of the map or gets stuck for some time, it is re-spawned to a different pose to generate a new trajectory.

For each trajectory, the robot pose and its corresponding heightmap patch are extracted at 20 Hz; the heightmap patch associated to a pose is centred on the robot’s position and oriented in such a way that the robot is facing towards the top of the patch (see Fig. 2-right). If and only if the distance between the current pose $X^{\text{robot}}(t)$ and a future pose $X^{\text{robot}}(t+T)$ is greater than a threshold d and aligned with the robot’s orientation, then the patch is labelled as traversable.

Figure 4 illustrates the traversability labelling for patches along two trajectories in the left-most heightmap of Fig. 3 with $T = 1$ s and $d = 0.3$ m. For the trajectory on the left, the first patches that face the hill-like obstacle at the centre are labelled as traversable. Thereafter, where the robot slightly slips along the hillside, the patches are marked as not traversable while the rest of the trajectory as traversable. In the trajectory on the right, the patches intersecting

¹ data and code to reproduce our results are available online: https://github.com/romarcg/traversability_estimation

with the obstacle are clearly identified as not traversable. As the robot keeps trying moving forward, interaction with the obstacle makes it deviate from its original orientation towards a traversable set of patches.

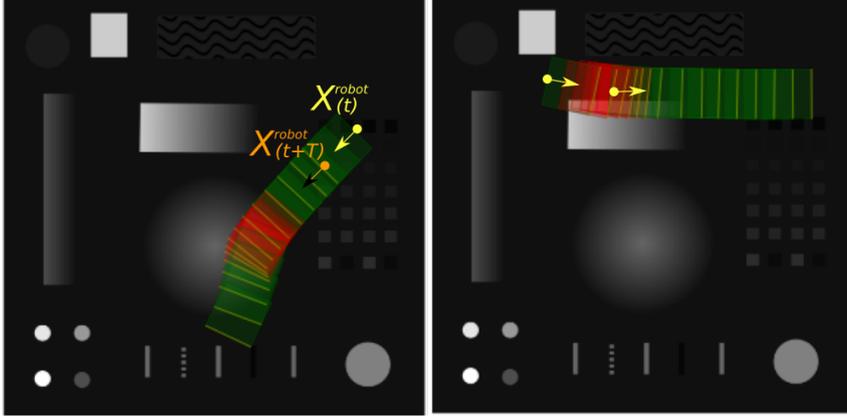


Fig. 4. Examples of trajectories extracted from simulation. Yellow arrows indicate the starting pose and initial direction of the corresponding trajectory. Green patches represent a positive traversability label while red patches indicate the opposite. The size of each heightmap is 10×10 m. Traversability threshold $d = 0.3$ m.

3.2 Dataset Generation

From each simulated trajectory (on average 12 s), we sample the robot pose and the corresponding heightmap patch at 20 Hz. The patch and its traversability label represents an instance in the dataset.

Three datasets were generated: D_{train} , a training dataset from synthetic data (40k samples); $D_{\text{eval, syn}}$, an evaluation dataset from synthetic data (10k samples); and $D_{\text{eval, real}}$, an evaluation dataset from real elevation maps (5 k samples). The maximum height of the heightmaps is 1 m for D_{train} and $D_{\text{eval, syn}}$; 3 m for $D_{\text{eval, real}}$. All datasets were generated using the robot described in Sec. 3.1, a patch size of 60×60 px, $T \approx 1$ s and $d = 0.3$ m.

Figure 5 illustrates patches extracted along the first part of the leftmost trajectory of Fig. 4. The values in each patch are offset in such a way that the centre (i.e. the robot's position) is mapped to height 0. This makes the patches independent on their absolute height on the heightmap, a feature that does not affect traversability.

Dataset $D_{\text{eval, real}}$ was generated using an elevation map from a Swiss gravel pit obtained by a flying drone [19]. The original area of this map is 0.48 km^2 , with a resolution of 5 cm/px and a maximum height of 50m. We cropped two regions of this map and scaled them down to form two heightmaps of 10×10 m

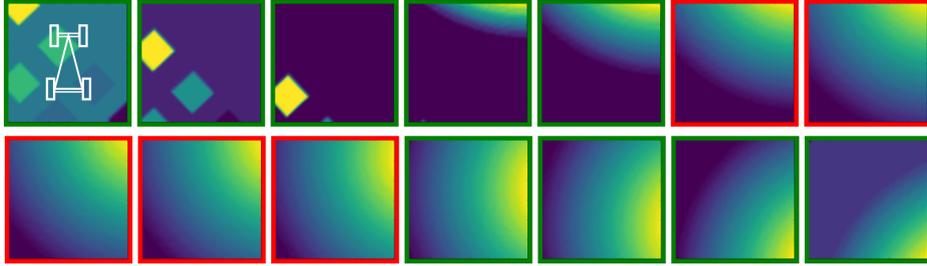


Fig. 5. Examples of labelled patches for the first trajectory in Fig. 4. Border colours indicate if a patch was labelled as traversable (green) or not (red). The colormap is used for visualization only, image data is grayscale.

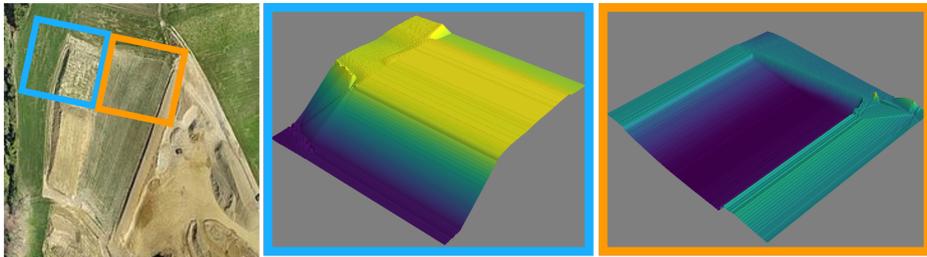


Fig. 6. Surfaces of the two heightmaps from real elevation data. Heightmaps were extracted from a mapping of a Swiss gravel pit [19].

with a resolution of 1.2 cm/px and a maximum height difference of 3 m. Figure 6 shows a reference image of the gravel pit and the surfaces of the two extracted heightmaps.

3.3 Training Traversability Classifiers

We cast the problem of estimating terrain traversability as a classification problem on heightmap data. We compare two alternative approaches: extracting descriptive features from each heightmap patch and then applying standard statistical classification techniques [20], or adopting Convolutional Neural Networks, a now-standard deep-learning approach [21] which operates directly on the raw input data. In either case, the output of the classifier indicates whether a patch is traversable.

For the **feature-based approach** we wanted to extract from the input heightmap patch some quantities that are indicative of whether the patch is traversable or not; such features may be for example the average terrain steepness in the robot’s motion direction (i.e. from the bottom of the patch to the top), or the maximum height of any steps in patch. Therefore, in our solution, we compute the Histogram of Gradients (HOG) of the heightmap patch, which includes these pieces of information; note that the gradient of a heightmap corre-

sponds to the local steepness of the terrain. Computing HOG over 6 orientations, 8×8 px per cell, and a block of 3×3 cells, results on a descriptor with 324 features that we classify by means of a Random Forest (RF) classifier [22] with 10 trees.

In the **CNN-based approach** approach, it is expected that the network autonomously learns meaningful, problem-specific features; because the input shape is high-dimensional and no prior knowledge of the problem is provided to the model, this approach requires more training data. Our CNN is built on the Keras [23] frontend powered by TensorFlow [24], and implements a 60×60 px input layer, followed by: a 3×3 convolution layer with 5 output maps; a 3×3 convolution layer with 5 output maps; a 2×2 Max-Pooling layer; a 3×3 convolution layer with 5 output maps; a fully connected layer with 128 output neurons; a fully connected layer with 2 output neurons followed by a softmax layer (output). All layers implement the ReLU activation function. The network is trained for 50 epochs to minimize a categorical cross-entropy loss using the Adadelta optimizer.

4 Experimental Results

In the following, we compare the two classifiers described in Section 3.3 with a baseline dummy classifier that always returns the class most frequent in the training set.

4.1 Classification Results

Table 1 summarizes the performance of the three estimators on the two different evaluation datasets (disjoint from the training set): $D_{\text{eval,syn}}$ and $D_{\text{eval,real}}$.

| | $D_{\text{eval,syn}}$ | | $D_{\text{eval,real}}$ | |
|---------------|-----------------------|--------|------------------------|--------|
| | ACC | AUC | ACC | AUC |
| CNN | 0.9134 | 0.9756 | 0.7456 | 0.8729 |
| Feature-based | 0.7884 | 0.9128 | 0.5940 | 0.6556 |
| Baseline | 0.4956 | 0.4957 | 0.5072 | 0.4974 |

Table 1. Performance of CNN, feature-based and baseline approaches. Accuracy (ACC) and area under the ROC curve (AUC) metrics are from a synthetic dataset generated similarly as the training dataset, and a dataset from the gravel pit described in Fig. 6.

We observe that the CNN estimator outperforms both the baseline and feature-based approaches on both evaluation datasets. Performance is lower on the $D_{\text{eval,real}}$ dataset than on $D_{\text{eval,syn}}$, probably because elevation patterns in the training dataset are more similar to the latter than to the former.

4.2 Traversability Estimation on Real-elevation Datasets

We evaluate our trained classifiers on an additional map. This dataset consists on a mining quarry (see Fig. 7) of 0.51km^2 [19], with some challenging roads designed for cars and trucks. Because our robot is roughly 4.5 times smaller than a car, we isotropically downscale the heightmap to match our robot’s size (Table 2 summarizes the process).



Fig. 7. Heightmap extracted from Sensefly’s mining quarry dataset [19]. Reference top-view image and perspective view of the elevation map from the extracted red region are shown at left and right respectively.

| | Original | Cropped and Scaled |
|------------------------|----------|--------------------|
| area (km^2) | 0.51 | 0.9 |
| resolution (cm/px) | 9 | 1.9 |
| max height (m) | 165 | 10 |

Table 2. Description of the original mining quarry dataset displayed in Fig. 7 and the scaled version used to evaluate our traversability estimation model.

For this analysis, we fix a direction and iterate over the entire heightmap extracting patches of $60 \times 60\text{px}$ with a stride of 5px . This process is equivalent as translating the robot’s position over the map while keeping a fixed orientation. Figure 8 shows the traversability estimation for the mining quarry for four orientations, indicated by the arrows. Traversability is represented as a coloured overlay on the surface of the heightmap (traversable is green, not traversable is red).

We note that the estimator correctly marks the main road (≈ 1.9 m wide) as traversable in all directions, and narrow roads as traversable only lengthwise.

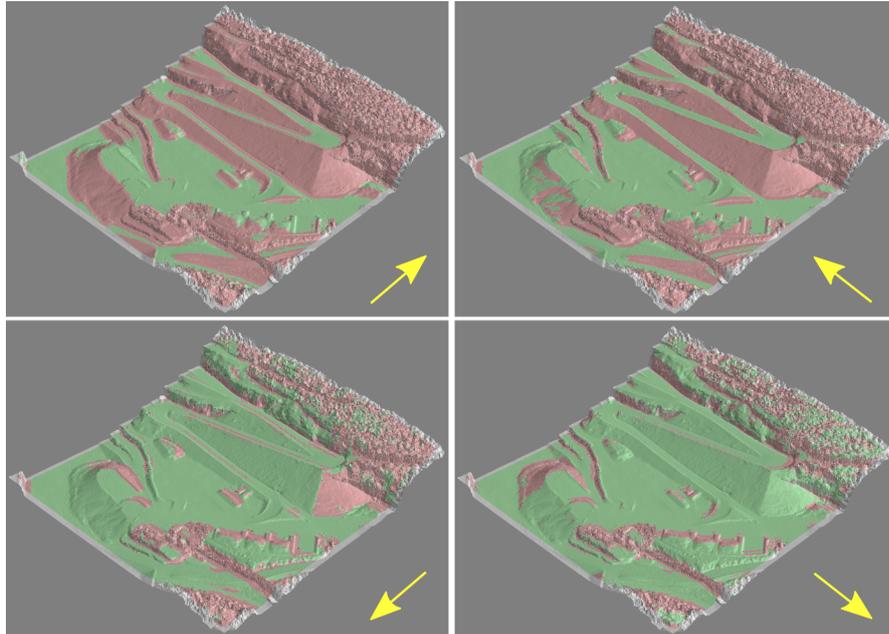


Fig. 8. Traversability estimation for the mining quarry elevation map, for four orientations of the robot (arrow).

Slopes are traversable only downhill or, sometimes, transversally. The rough surface created by the bushes at the top of the elevation map is correctly found as non-traversable in all directions. For all directions, the low-lands and plan terrain are correctly identified as traversable as it is expected. The traversability maps can be used by a planner to navigate avoiding non-traversable sections while moving towards a goal.

5 Discussion, Conclusions and Perspectives

We presented an approach for traversability estimation that casts the problem as an image classification task. Classifiers trained on simulation data capture complex characteristics of different robot models and quickly estimate traversability maps on large unseen terrains.

In this work we assumed that the terrain's 3D shape is the only factor influencing traversability; in some scenarios other factors may play a role, such as compactness, friction, and instability. Our framework could handle these factors provided they can be simulated: for example, if terrains whose 3D shape suggests a loose gravel surface were simulated with lower friction, the classifier would automatically learn that steep slopes can't be negotiated on such surfaces; in this perspective, one may add additional inputs to the classifier (such as the visual texture of the terrain) which may help differentiate the actual surface.

Another limitation of the current approach is that it does not capture the robot dynamics, such as the speed with which the robot approaches an obstacle: in fact, we limit our attention to slow robots operating on rugged terrains, where dynamic aspects have negligible impact.

Even though this paper focuses on an external perception of the world (i.e. the heightmap, which in practical scenarios is acquired by flying robots or derived by GIS data), the same approach can be used to deal with robot-centric perceptions, such as robot-mounted LIDAR, laser scanners or even standard cameras. In that case, however, the robot would only be able to estimate the traversability of the terrain in its vicinity. This application would be still very important in practice, and is a future research topic.

Acknowledgments

This research was supported by the Swiss National Science Foundation through the National Centre of Competence in Research Robotics.

References

1. Hadsell, R., Scoffier, M., Muller, U., LeCun, Y., Sermanet, P.: Mapping and planning under uncertainty in mobile robots with long-range perception. In: In Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS). (2008) 1–6
2. E., U., E., S.: Traversability: A case study for learning and perceiving affordances in robots. *Adaptive Behavior* **18** (2010) 258–284
3. Molino, V., Madhavan, R., Messina, E., Downs, A., Balakirsky, S., Jacoff, A.: Traversability metrics for rough terrain applied to repeatable test methods. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. (Oct 2007) 1787–1794
4. Brooks, C.A., Iagnemma, K.D.: Self-supervised classification for planetary rover terrain sensing. In: 2007 IEEE Aerospace Conference. (March 2007) 1–9
5. Bekhti, M.A., Kobayashi, Y., Matsumura, K.: Terrain traversability analysis using multi-sensor data correlation by a mobile robot. In: 2014 IEEE/SICE International Symposium on System Integration. (Dec 2014) 615–620
6. Silver, D., Bagnell, J.A., Stentz, A.: Learning from demonstration for autonomous navigation in complex unstructured terrain. *Int. J. Rob. Res.* **29**(12) (October 2010) 1565–1592
7. Chilian, A., Hirschmuller, H. In: Stereo camera based navigation of mobile robots on rough terrain. (2009) 4571–4576
8. Cunningham, C., Wong, U., Peterson, K.M., Whittaker, W.L.R. In: Predicting Terrain Traversability from Thermal Diffusivity. Springer International Publishing, Cham (2015) 61–74
9. Herbert, M., Caillas, C., Krotkov, E., Kweon, I.S., Kanade, T.: Terrain mapping for a roving planetary explorer. In: Proceedings, 1989 International Conference on Robotics and Automation. (May 1989) 997–1002 vol.2
10. Balta, H., Cubber, G.D., Doroftei, D., Baudoin, Y., Sahli, H.: Terrain traversability analysis for off-road robots using time-of-flight 3d sensing. In: In Proc. of the Int. Workshop on Robotics for Risky Environment - Extreme Robotics. (2013)

11. Lacroix, S., Mallet, A., Bonnafous, D., Bauzil, G., Fleury, S., Herrb, M., Chatila, R.: Autonomous rover navigation on unknown terrains functions and integration. In: *Experimental Robotics VII. ISER '00*, London, UK, UK, Springer-Verlag (2001) 501–510
12. Schwarz, M., Behnke, S.: Local navigation in rough terrain using omnidirectional height. In: *ISR/Robotik 2014; 41st International Symposium on Robotics*. (June 2014) 1–6
13. Hudjakov, R., Tamre, M.: Aerial imagery terrain classification for long-range autonomous navigation. In: *2009 International Symposium on Optomechatronic Technologies*. (Sept 2009) 88–91
14. Långkvist, M., Kiselev, A., Alirezaie, M., Loutfi, A.: Classification and Segmentation of Satellite Orthoimagery Using Convolutional Neural Networks. *Remote Sensing* **8**(4) (2016)
15. Jackel, L.D., Krotkov, E., Perschbacher, M., Pippine, J., Sullivan, C.: The darpa lagr program: Goals, challenges, methodology, and phase i results. *Journal of Field Robotics* **23**(11-12) (2006) 945–973
16. Shneier, M., Chang, T., Hong, T., Shackelford, W., Bostelman, R., Albus, J.S.: Learning traversability models for autonomous mobile vehicles. *Autonomous Robots* **24**(1) (2008) 69–86
17. Hadsell, R., Sermanet, P., Ben, J., Erkan, A., Han, J., Muller, U., LeCun, Y.: Online learning for offroad robots: Spatial label propagation to learn long-range traversability. In: *Proceedings of Robotics: Science and Systems, Atlanta, GA, USA* (June 2007)
18. Rohmer, E., Singh, S.P.N., Freese, M.: V-rep: A versatile and scalable robot simulation framework. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. (Nov 2013) 1321–1326
19. SenseFly: Elevation datasets. <https://www.sensefly.com/drones/example-datasets.html> (2016)
20. Bishop, C.M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
21. Schmidhuber, J.: Deep learning in neural networks: An overview. *Neural Networks* **61** (2015) 85–117
22. Breiman, L.: Random Forests. *Machine Learning* **45**(1) (2001) 5–32
23. Chollet, F., et al.: Keras. <https://github.com/fchollet/keras> (2015)
24. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015) Software available from tensorflow.org.