Comparing CNN Inputs for Terrain Classification using Simulation

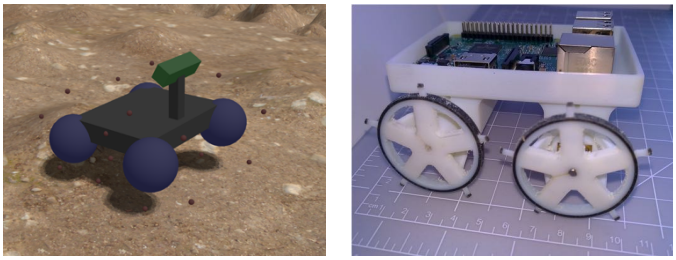
Anthony J. Clark, Jesse Simpson, and Jared Hall
Computer Science Department
Missouri State University
Springfield, Missouri, USA
anthonyclark@missouristate.edu

***Abstract*—Mobile robots frequently operate in rough, uneven terrain. One way for them to identify easier to traverse paths is to use deep learning methods, such as a convolutional neural network (CNN). It is not clear, however, what input should be provided to the CNN to best enable it to classify different terrain. In this study, we investigate and compare several input formats for improving terrain classification using a CNN. All experiments take place in simulation, where we have complete control over terrain (e.g., shapes and textures) and information about our robot. Our experiments lead us to the following conclusions: (1) input formats should prefer grayscale over color images as color has a tendency to overfit the training data and (2) disparity maps also improve classification compared with raw image data. These results can be used to improve the performance of terrain classification; particularly as they apply to transformable-wheel robots.**

***Index Terms*—mobile robotics, transformable robotics, deep learning**

I. Introduction

Making decisions based on visual input is a challenge for mobile robots. It is not always obvious how to best use pixel data to choose a path or decide a robot's next action. In this paper, we investigate how a convolutional neural network (CNN) can be used with pairs of images from a stereo camera to make decisions about its locomotion mode. Specifically, a CNN will output a terrain classification that can then be used to transform the wheels of the robot pictured in Figure 1.



(a) Simulation (b) Prototype

Figure 1. A mobile robot with transformable wheels.

The pictured robot can extend "legs" from the outer rim of each wheel. On even terrain, legs are fully retracted so that camera and other sensor data is not affected by vibrations or bumps. On rough ground, legs can be fully extended so that the robot can climb obstacles. Legs can be extended by any distance between 0 and the wheel radius, enabling a fine control over the balance between mobility and vibrations. To use wheel-legs effectively, we require a method for deciding how far they should be extended.

Despite a wide variety of transformable-wheel devices in existing literature [1], to date little research has been conducted on how to best decide when to transform the wheels. Most current methods rely on detecting poor mobility after it occurs. For example, measuring wheel slippage with encoders or using GPS to measure *actual* speed and comparing with *expected* speed. More recently, researchers have been using vision data to predict when a wheel should be transformed, rather than react to an already poor situation [2], [3].

Deep learning techniques have produced state-of-the-art results for many vision related tasks. Such techniques have been applied to terrain classification [4]–[6], though, not in the context described in this study. As a first step in applying deep learning to the problem of transformable-wheel robotics, in this study we investigate using stereo camera images as input and terrain classifications as output. However, before we can train and deploy a CNN-based terrain classifier, we first need to answer the following question: **what input format should we provide to a CNN such that it can make accurate predictions about the current terrain?** For example, we can input three-channel RGB images, disparity maps, grayscale images, etc., or some combination of these formats.

We use Gazebo [7] to simulate and collect image data. We label images based on the simulated robot's ability to navigate three different types of terrain: *high*, *medium*, or *low* mobility. These labeled images are then fed to a CNN, which is responsible for classifying terrain and making decisions regarding wheel-leg extensions.

Our results show that a combination of disparity maps and grayscale images should be used to classify terrain (in terms of traversability) for a transformable-wheel robot. More specifically, we find that using RGB images leads to better validation accuracy, but significantly worse accuracy on a testing set in which images are generated

using a second simulation environment. Simulation and deep learning code for this study can be found here: https://github.com/anthonyjclark/terrain-classification.

II. Related Work

A. Deep Learning Terrain Classification

Mobile robots need the capacity to determine traversability of terrain to navigate in complex real-world environments. This is especially true for transformable robots that can change their abilities to better handle uneven ground. Dongshin Kim *et al.* [4] compare the performance of patch, pixel, and super-pixel segmentation for terrain traversability classification, and while their super-pixel approach is an improvement over existing techniques, the output is still difficult to interpret. Deng *et al.* [8] propose a method for finding traversable regions from a mobile robot's camera. They used a vanishing point method with self-supervised learning to find traversable paths.

More recently, many research groups have turned to neural networks for classifying terrain. Iwashita *et al.* [6] propose two novel deep learning-based terrain classification methods: TU-NET and TDeepLab. These architectures combine the use of visual features with thermal features to provide robust classification of terrain by taking advantage of thermal differences. In [9], Zhang *et al.* integrate a CNN model with a near-to-far learning strategy to improve the accuracy of terrain segmentation and make it more robust against wild environments. Kim *et al.* [5] propose a novel multimodal CNN architecture comprising two input streams: 2D images and 3D point clouds from LiDAR. The combination of these two inputs provides improved accuracy when compared with image-only data. In [10], Chavez-Garcia *et al.* present a neural network architecture, which when given an image and a height map, classifies the traversability of the terrain and yields the path the robot should take. The network takes aerial images as input (as opposed to a view from the robot itself as shown in Figure 2(b)).

Although these works are related to that which we present here, they are focused on finding even terrain that any wheeled robot can traverse. In this study, we are concerned with classifying terrain with differing degrees of traversability. Consider rough terrain that a wheeled robot cannot handle; a transformable-wheel robot will be able to traverse such ground by extending its legs.

B. Perception for Transformable-Wheel Robots

At the cost of adding actuators and additional control complexity, transformable-wheel robots gain the benefits of both smooth wheel and legged-wheel locomotion [11]. Specifically, on even terrain they operate without the vibrations caused by wheel-legs, and on uneven terrain they can climb obstacles. To date, however, there has been limited research into classifying terrain for transformable-wheel robots. Wang *et al.* [3] used a CNN with image data to select appropriate gaits for the TurboQuad-V vehicle. They relied on hand-labeled data. Xu *et al.* [2] proposed a novel system using binocular vision, ultrasonic sensors, and an IMU to detect if the current ground is a plane, step, or incline. In contrast to these two studies, here we present a method for automated training data labeling using simulation that works in a more complicated environment. On the other hand, we do not present results from real-world experiments.

III. Methods

This research study is meant to address an important gap in the current literature for mobile robots: *what input formats are appropriate for classifying degrees of traversability.* Current terrain classifying techniques do not consider transformable wheels (i.e., robots that can traverse rough terrain), and most studies do not compare different input formats for a CNN. We are starting with a simulation-based study so we have full control over terrain and camera data, and so we can create an automatic labeling process using simulated velocity measurements.

A. Transformable-Wheel Robot

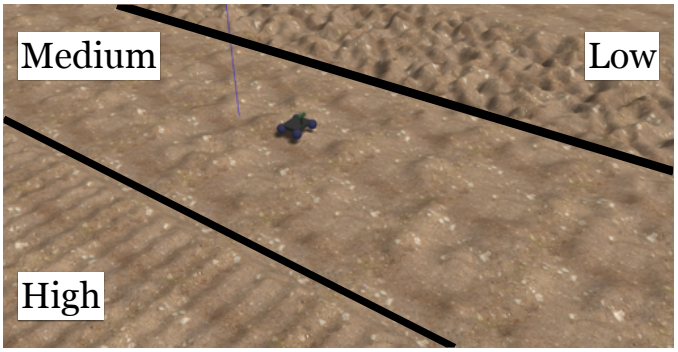
Motivation for this study comes from our prior work with the transformable-wheel robot in Figure 1. Specifically, it is difficult to decide when the wheel-legs should be extended. Typical forms of sensing are too course-grain (GPS) or too sensitive to noise (IMU). Moreover, they can only **react** to poor mobility, whereas in this study we develop a vision-based system for **predicting** appropriate wheel-leg extensions ahead of time. The current prototype is 8 cm long by 10 cm wide, and it operates using a skid-steer drive. More details about this device are presented in [11].

B. Gazebo Simulation Environment

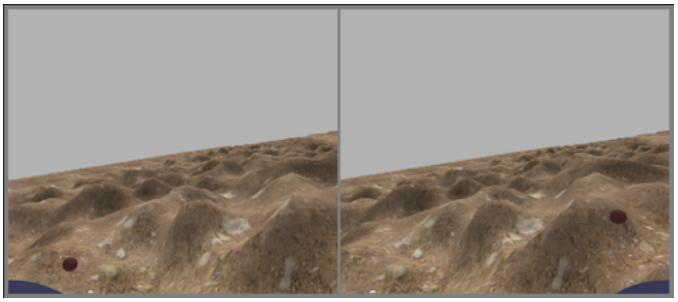
Similar to Chavez-Garcia *et al.* [10], we rely on the Gazebo simulation environment [7] for generating training data. Gazebo is specifically made for robotics research, and as such it has support for complex terrain and sensing capabilities. Figure 2 shows our robot in a custom simulation environment. In the left-hand image you can see a view of the three different terrain types: high, medium, and low mobility. These names refer to our robot's ability to traverse the terrain. For example, our robot can easily cover the terrain labeled "high" without use of its wheel-legs, but it has significant trouble on the terrain labeled "low" even with the legs fully extended. The "medium" mobility terrain requires use of the wheel-legs to traverse.

Figure 2(b) shows the robot's view. These are the images being fed to the CNN in different formats. Figure 1 shows the placement and angle of the camera. For this study, the robot was only allowed to operate in one of three modes: legs fully retracted, half-way extended, or fully extended.

To generate training and validation images, we created three environments. One in which each of the terrain types was placed in the middle. In doing so, we have images in which the robot is traversing each terrain type while having



(a) Gazebo Environment



(b) View from the Stereo Camera

Figure 2. (a) A view of the different terrain types in our Gazebo simulation environment. The black lines and annotations have been added for presentation here. (b) A view of the stereo camera output as seen from the robot's perspective.

each of the other terrain types appear in the frame on the left and right peripheries. Figure 2 depicts this setup for the case of having high-mobility terrain in the middle. The simulated robot also traversed the terrain laterally across all three types. These different traversals provide a wide-variety of images with which we can train a CNN.

To label each image, we took the current speed of the robot and compared it with the expected speed. When the current speed was less than the expected we knew that the terrain must belong to the medium or low mobility class depending on how much the wheel-legs were extended. We used all possible combinations of wheel-leg extensions, current terrain, and terrains that can be seen in the stereo camera's peripheral vision. In total, we gathered 4688, 3866, and 3218 images for the high, medium, and low mobility terrains, respectively (11 772 images in total).

C. Deep Learning with fastai

We use the *fastai* library [12] to train our CNN. We use the ResNet34 architecture [13] since fastai has a version that includes state-of-the-art techniques for regularization and optimization. Additionally, fastai has support for cyclical and progressive learning rates. Our assumption for this study is that changing the architecture will not change the relative results between the different types of input formats we feed through the network. Details regarding data augmentation, validation, batch sizes, and other hyper-parameters can be found in the linked repository.

IV. Discussion and Results

According to recent work by Gowda *et al.* [14], the RGB color space provides *good* results for most applications. Thus, here we focus on the following image input formats: RGB images, grayscale images, and disparity maps generated from the grayscale images. Similar to Iwashita *et al.* [6], we combine multiple input formats and feed them to the network. Specifically, we combine the disparity maps and left/right stereo images into a single input.

Table I
Training Results for Different CNN Input Formats

		Valid (Dirt)		Test (Grass)	
CNN Input	#	%	s/epoch	%	ms/image
RGB	3	97.8	21	33.8	7
Stack-RGB	6	95.8	32	33.8	9
Grayscale	1	97.5	17	46.6	6
Stack-Gray	2	95.0	20	51.4	7
Disp. Maps	1	74.0	17	58.8	6
Stack-RGB + Disp.	7	95.1	33	33.8	10
Stack-Gray + Disp.	3	94.5	20	58.8	7

Single Image in RGB. In our first experiment, we use only images from the left lens of the stereo camera (we conducted the same experiment for images from the right lens with identical results). Results from this and all subsequent experiments are shown in Table I. Validation accuracy after 10 epochs reached 97.8 % (we set aside 20 % of the training data for validation). However, this model performs at the same level as a random classifier when provided inputs generated with a grass texture (see Figure 3) instead of a dirt texture. Therefore, we can infer that the model relies heavily on color information. This experiment provides a baseline against which we can compare other input formats.

Stacked Images in RGB. For the second model, we took corresponding pairs of images and "stacked" them to create a six-channel input (two sets of RGB channels). The second column of Table I indicates the number of input channels for each model. This model provides similar results to the previous: high performance on the validation set and poor performance on the test set. Although performance was similar, this model took 50 % longer to train.

Single Image in Grayscale. Since the first two models rely heavily on color, we used grayscale images for the third. This model was the quickest to train and requires the least amount of time to execute when deployed on the actual device; as shown in columns four and six of the table. Columns four and six refer to the training time per epoch and the prediction time per image, respectively. In addition to quicker processing, this model performs better than random (46.6 %) at classifying terrain.

Stacked Images in Grayscale. For the fourth model, we stacked left/right pairs of grayscale images to create a set of two-channel inputs. At a slight cost of processing time we improve accuracy by a small margin. We were surprised that the extra information provided by stereo images did not lead to even greater increases in testing accuracy. We hypothesize that increasing the training data size (and variety) will further the gap between grayscale and two-channel grayscale inputs.

Disparity Maps. Inputs to the fifth model are generated by creating disparity maps from the grayscale image pairs. We tested several algorithms (and algorithm settings) for producing disparity maps, and we achieved our highest accuracy using a method developed by Hirschmuller [15]. Figure 3 shows an example pair of stereo images and the resulting disparity map.

Figure 3. Left and right image pairs from the robot's simulated stereo camera and the resulting disparity map. The left and right images are first converted to grayscale.

Although this model had the poorest training accuracy, it had the highest testing accuracy. The poor training accuracy can largely be attributed to the required tuning of the disparity map algorithm. The algorithm includes several parameters that drastically affect performance, and changing these parameters can have different affects on different pairs of images. With a larger dataset and a higher variety of textures in the scene, we would expect creating disparity maps to be even more difficult.

Examining the confusion matrix for this model (see Figure 4) shows that, while it has lower accuracy, it mostly confuses medium and high mobility images, which are the most visually similar. Specifically, the model has an accuracy of 89 %, 62 %, 73 % on low, medium, and high mobility images, respectively. If we combine the medium and high classes, the model has an accuracy of 95 %. Again, we hypothesize that an increase in training images will improve accuracy and enable the model to differentiate between medium and high mobility terrain.

Stacked Images in RGB with Disparity Maps. The next model combines disparity maps with the stacked RGB images, resulting in seven-channel inputs. Validation and testing accuracies are similar to our prior models using RGB information, showing that the superior validation performance of RGB data is dominating the CNN's output. Specifically, the disparity map channel appears to be ignored during training since it would lead to a lowered validation (and training) accuracy.

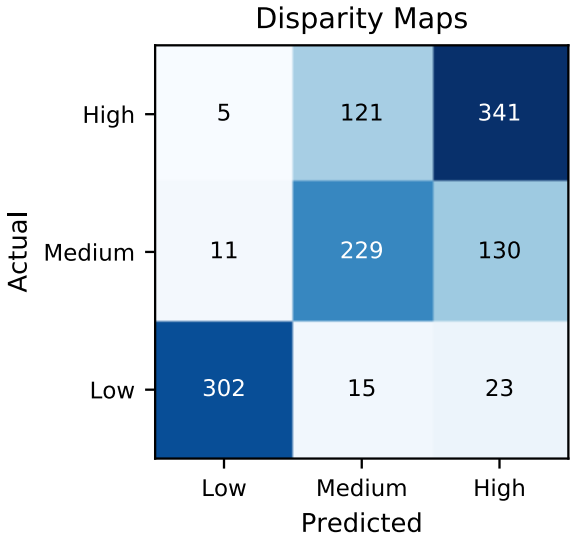


Figure 4. A confusion matrix for the disparity map model on the validation dataset. Values in the diagonal show cases of correctly classified images.

Stacked Images in Grayscale with Disparity Maps. The final model includes disparity maps and grayscale images for a total of three channels. This model performs as well as the disparity map model. Interestingly, the grayscale images do not appear to override the disparity maps for this model. Specifically, the model has both a relatively high validation accuracy (94.5 %) and the highest testing accuracy (58.8 %). This indicates that a combination of grayscale images and disparity maps are a good combination for our classification task. This model can combine the benefits of disparity maps with those of using raw image data.

V. Conclusions

In this study, we compared seven different input formats for classifying terrain as high, medium, or low mobility for our transformable-wheel robot. Our results lead to the following conclusions: (1) color in RGB images can lead to good validation accuracy, but over-reliance on color information can lead to poor testing accuracy; (2) validation accuracy does not correlate with testing accuracy when you expect to encounter different types of terrain (even if they only differ in color); (3) disparity map results can likely be improved by further tuning the algorithms parameters; and (4) results using stacked images and disparity maps will likely improve by generating more training data.

Acknowledgments

We gratefully thank members of the ARCS Lab for their time and effort developing the transformable wheel

robot, and the NVIDIA Corporation for their donation of a Quadro P6000 GPU.

References

[1] P. Moubarak and P. Ben-Tzvi, "Modular and reconfigurable mobile robotics," *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1648–1663, Dec. 2012. DOI: 10.1016/j.robot.2012.09.002.

[2] Q. Xu, W. Guo, M. Sherikar, C. Wu, and L. Wang, "Environment Perception and Motion Strategy for Transformable Legged Wheel Robot on Rough Terrains," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2018, pp. 2153–2158. DOI: 10.1109/ROBIO.2018.8665300.

[3] T. Wang, D. Sung, and P. Lin, "Terrain Classification, Navigation, and Gait Selection in a Leg-Wheel Transformable Robot by Using Environmental RGBD Information," in *International Automatic Control Conference*, Nov. 2018. DOI: 10.1109/CACS.2018.8606730.

[4] Dongshin Kim, Sang Min Oh, and J. M. Rehg, "Traversability classification for UGV navigation: A comparison of patch and superpixel representations," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2007, pp. 3166–3173. DOI: 10.1109/IROS.2007.4399610.

[5] D.-K. Kim, D. Maturana, M. Uenoyama, and S. Scherer, "Season-Invariant Semantic Segmentation with a Deep Multimodal Network," in *Field and Service Robotics*, vol. 5, 2018, pp. 255–270. DOI: 10.1007/978-3-319-67361-5_17.

[6] Y. Iwashita, K. Nakashima, A. Stoica, and R. Kurazume, "TU-Net and TDeepLab: Deep Learning-Based Terrain Classification Robust to Illumination Changes, Combining Visible and Thermal Imagery," in *2019 IEEE Conference on Multimedia Information Processing and Retrieval*, San Jose, CA, USA, Mar. 2019, pp. 280–285. DOI: 10.1109/MIPR.2019.00057.

[7] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, Sendai, Japan: IEEE, 2004, pp. 2149–2154. DOI: 10.1109/IROS.2004.1389727.

[8] F. Deng, X. Zhu, and C. He, "Vision-Based Real-Time Traversable Region Detection for Mobile Robot in the Outdoors," *Sensors*, vol. 17, no. 9, p. 2101, Sep. 2017. DOI: 10.3390/s17092101.

[9] W. Zhang, Q. Chen, W. Zhang, and X. He, "Long-range Terrain Perception Using Convolutional Neural Networks," *Neurocomput.*, vol. 275, pp. 781–787, Jan. 2018. DOI: 10.1016/j.neucom.2017.09.012.

[10] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti, "Learning Ground Traversability From Simulations," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1695–1702, Jul. 2018. DOI: 10.1109/LRA.2018.2801794.

[11] A. J. Clark, K. A. Cissell, and J. M. Moore, "Evolving Controllers for a Transformable Wheel Mobile Robot," *Complexity*, Dec. 2018. DOI: 10.1155/2018/7692042.

[12] J. Howard *et al.*, *Fastai*, 2018.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*, Dec. 2015.

[14] S. N. Gowda and C. Yuan, "ColorNet: Investigating the importance of color spaces for image classification," *arXiv:1902.00267 [cs]*, Feb. 2019.

[15] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, Feb. 2008. DOI: 10.1109/TPAMI.2007.1166.