



Faculty of Electrical Engineering
Department of Cybernetics

Bachelor's thesis

Transfer Learning of Ground Robot Terrain Experience

Josef Zelinka

May 20, 2022

Supervisor: Ing. Miloš Prágr

I. Personal and study details

Student's name: **Zelinka Josef** Personal ID number: **492328**
Faculty / Institute: **Faculty of Electrical Engineering**
Department / Institute: **Department of Cybernetics**
Study program: **Cybernetics and Robotics**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Transfer Learning of Ground Robot Terrain Experience

Bachelor's thesis title in Czech:

Transfer learning v p enosu zkušeností s terénem mezi rozdílnými roboty

Guidelines:

- 1) Familiarize yourself with machine learning techniques focusing on transfer learning [1], its use within the context of mobile robotics [2], and terrain characterization and traversability assessment [3,4].
- 2) Propose and develop a framework for learning terrain traversability properties and knowledge transfer between different ground robots.
- 3) Collect a dataset capturing traversability of different robots over terrains and report on the performance of the proposed solution.
- 4) Possibly deploy the proposed framework in navigation of a real mobile robot.

Bibliography / sources:

- [1] F. Zhuang et al., "A Comprehensive Survey on Transfer Learning," Proceedings of the IEEE, vol. 109, no. 1, pp. 43–76, Jan. 2021, doi: 10.1109/JPROC.2020.3004555.
- [2] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," Science Robotics, vol. 5, no. 47, p. eabc5986, Oct. 2020, doi: 10.1126/scirobotics.abc5986.
- [3] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," Engineering Applications of Artificial Intelligence, vol. 26, no. 4, pp. 1373–1385, Apr. 2013, doi: 10.1016/j.engappai.2013.01.006.
- [4] M. A. Bekhti, "Traversability Cost Prediction of Outdoor Terrains for Mobile Robot Using Image Features," Shizuoka University, 2020. doi: 10.14945/00027788.

Name and workplace of bachelor's thesis supervisor:

Ing. Miloš Prágr Department of Computer Science FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **27.01.2022** Deadline for bachelor thesis submission: **20.05.2022**

Assignment valid until: **30.09.2023**

Ing. Miloš Prágr
Supervisor's signature

prof. Ing. Tomáš Svoboda, Ph.D.
Head of department's signature

prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature



Declaration

I declare that the presented work was developed independently and that I have listed all sources of the information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, May 20, 2022

.....
Josef Zelinka



Acknowledgement

I would like to thank my supervisor Ing. Miloš Prágr, who helped me overcome the challenges of writing this bachelor's thesis and was always ready to discuss all arising problems. Secondly, I greatly appreciate the support of all members of my family, who have made it possible for me to pursue my goal with the studies of informatics and encouraged me during this journey.

Abstrakt

Pro autonomní roboty nasazené v neznámém prostředí je výhodné odhadovat prostupnost okolního terénu, jelikož správné odhadnutí prostupnosti může pomoci k naplánování lepší cesty vedoucí skrz terén. Tvorba modelu pro odhad prostupnosti pro každý dostupný robot je náročná, a proto je vhodné se pokusit znalosti sdílet mezi různými roboty. Avšak ve skupině heterogenních robotů nemůže být informace o prostupnosti sdílena přímo mezi jednotlivými roboty, protože členové skupiny mohou být vybaveni různými sensory, nebo jejich tělo může mít různou stavbu, a proto vnímají prostředí rozdílně. V této práci navrhneme postup používající přenos informací, který zajistí přenesení modelu pro odhadování prostupnosti terénu mezi heterogenními roboty. Pro každý typ robotu je vytvořen samostatný model pro odhadování prostupnosti za pomoci konvoluční neuronové sítě, která odhaduje složitost prostupu terénem díky pozorování z exteroceptivních senzorů. Předtrénované konvoluční neuronové sítě odhadující prostupnost jsou přeneseny z robotu poskytujícího znalosti na robot, který další znalosti potřebuje. Poté, co je neuronová síť přenesena, nastane opětovné trénování sítě z dat nasbíraných robotem, který síť obdržel, což zajistí přizpůsobení skutečným v cílové oblasti své práce. Navržená metoda je otestována za pomoci heterogenních robotů, s kterými byly provedeny experimenty pro ověření funkce navrženého postupu pro přenášení znalostí o složitosti prostupu terénem.

Klíčová slova: heterogenní roboty, přenos informací, odhad prostupnosti, neuronové sítě

Abstract

For autonomous robots deployed in unknown environments, it is beneficial to reason about their traversability over the surrounding terrain, which can improve robots' future path-planning decisions. Creating a model assessing traversability for each robot is a challenging task, thus making it desirable to exchange the knowledge about traversability between various robots. However, in a team of heterogeneous robots, the models assessing traversability cannot be shared directly among the members, as robots can possess different morphology or sensory equipment and experience the terrain differently. In this thesis, we propose an approach using transfer learning capable of transferring models assessing traversability between heterogeneous robots. The individual traversability assessment models used by the particular robots are created using convolutional neural networks that assess the difficulty of traversal through terrain from observations of exteroceptive sensors. Pre-trained convolutional neural networks capable of assessing traversability are transferred between the robot from the source domain to the robot in the target domain to propagate the knowledge among heterogeneous robots. After transferring the neural network between the robots, the network is retrained again using data available in the target domain to be accommodated to the different tasks. Using real heterogeneous robots, we conducted experiments validating the feasibility of the proposed knowledge transfer approach.

Keywords: heterogeneous robots, transfer learning, traversability assessment, neural networks

Contents

1	Introduction	1
2	Related work	3
2.1	Traversability assessment	3
2.2	Machine learning principles	5
2.3	Neural networks	6
3	Robotic platforms	9
3.1	Scarab II	9
3.2	Spot	10
4	Problem Statement	12
5	Method	14
5.1	Dataset collection	14
5.2	Cost assessment model	15
5.3	Knowledge transfer	16
6	Results	17
6.1	Various cost assessment methods	17
6.1.1	Transfer between slope c_s and velocity c_v cost assessment method .	18
6.2	Transfer between Scarab II and Spot	20
6.2.1	Transfer between Spot and c_s	22
7	Conclusion	24
	References	25

List of Figures

1	An overview of transfer learning between heterogeneous robots	2
2	Fully connected layers of neural network	7
3	Scarab II robot	9
4	Tripod and pentapod gait utilized by Scarab II	10
5	Spot robot	11
6	Architecture of the cost assessment model	16
7	Transfer learning of cost assessment model	16
8	Terrains in the Bull Rock Cave	17
9	Progress of the cost assessment model's training	19
10	Scarab II's predictions compared with ground truth.	19
11	Scarab II's predictions of cost from exteroceptive sensors	20
12	Spot during dataset collection.	21
13	Cost assessment model' results after transfer between heterogeneous robots.	23
14	Scarab II's predictions of cost from exteroceptive sensors after transfer from Spot	23



List of Tables

1	Results of knowledge transfer between robots with varying cost assessment methods	18
2	Influence of prolonged training	18
3	Knowledge transfer from Spot to Scarab II	21
4	Knowledge transfer from Scarab II to Spot	22

Chapter 1

Introduction

For mobile robots, the correctly assessed traversability of the surrounding terrain allows for more informed path planning decisions and enables avoidance of hard-to-traverse terrains, which can cause harm to the body of the robot. Besides the terrain itself, the traversability assessment can depend on the robotic platform. E.g., a smaller-legged robot can consider taller grass more challenging to traverse than paved surfaces as its legs can be slowed down by the grass. In contrast, a larger robot can press the taller grass down with its long legs, thus making the traversal of grassy terrain comparable with hardpacked surfaces. In addition to the problem of mobile robot interaction with the terrain is high dimensional as the environment and action space of deployed platforms. The complexity makes it difficult to create a hand-crafted function or precise plans to tackle the problems. Thus, training the model to assess traversability from the robot’s experience with the surrounding terrain is necessary.

Our work is motivated by deployments in mobile robot exploration, where it is desirable to navigate through the surroundings as quickly as possible. To accomplish the quick exploration, we strive to deploy all available robotic hardware to parallelize the task as much as possible. However, not all robots in the created exploration team possess the same build and sensory equipment, which can make the assessed traversability models directly unshareable among all the members. Such a team comprising various robotic platforms is considered heterogeneous. Hence, even though the deployed robots would greatly benefit from the ability to assess traversability, the heterogeneous platforms likely require various cost assessment approaches. The cost variance between heterogeneous robots induces the problem of creating a standalone traversability assessment model for each platform in the team, which depends on acquiring sufficient experience for each robot. Besides, in a homogenous team, where all utilized platforms are identical, wear and tear during the deployment can create heterogeneities in the previously homogenous team. The aforementioned problems signal the need for a knowledge transfer approach, which would remove the urge for manual fine-tuning, or the creation of an entirely new traversability assessment model from scratch.

In this thesis, we present an approach for transfer learning of terrain traversability assessment between heterogeneous robotic platforms. The herein studied transfer learning belongs to a group of machine learning principles and aims to improve the performance in the target domain by the experience in the source domain¹. Our adaptation of the transfer learning to the robotic domain is illustrated in Figure 1, where the transfer helps to train the cost assessment model faster as we assume that the basic knowledge for cost assessment corresponds among heterogeneous platforms. Models assessing terrain traversability are created and trained for each robotic platform. The traversability assessment model comprises a regressor in the form of an artificial neural network. A possibly colored elevation map from exteroceptive sensors enters the neural network for training purposes. The neural network then creates assessments of continuous scores describing the difficulty of given terrain’s individual segments. We transfer the already trained traversability assessment models between the heterogeneous robots to achieve better cost predictions and reduce the size of the training datasets needed to train the traversability assessment model on each robotic platform.

¹Source domain (teacher) denotes the entity providing knowledge to the individual in the target domain (student).

1. Introduction

During the transfer, the neural network of the source model is used as the initial setting of the model in the target domain as it is further trained using collected data. Besides, we address the potentially mismatched dimensions of exteroceptive sensors' observations by adding a convolutional layer to accommodate the transferred model in the target domain.

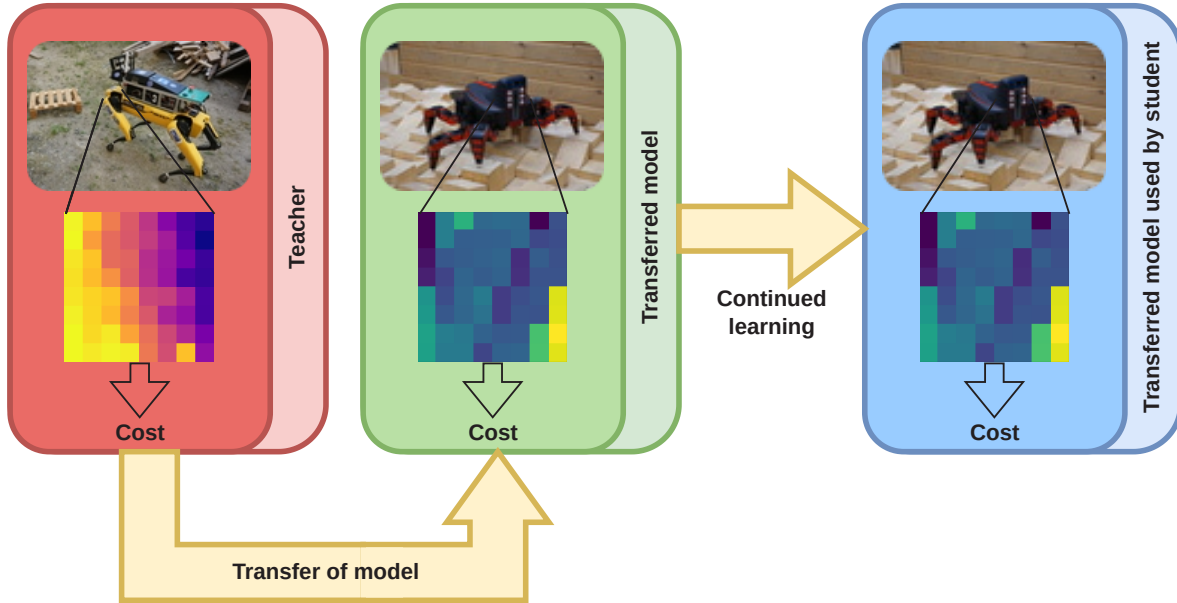


Figure 1: An overview of knowledge transfer between heterogeneous robots. Assuming that the cost assessment model for the teacher is available, the teacher’s model is transferred to the student and modified to accept the student’s observation format indicated by the different colormaps. Finally, the transferred model continues its learning using student’s observations to be informed about student’s experience.

The approach is experimentally evaluated in real scenarios with different robotic platforms. First, the approach is tested on a real hexapod walking robot with heterogeneous terrain experience simulated by various traversability assessment methods. Next, our experiments utilize robots with different morphology, where the method’s capabilities are thoroughly tested.

The rest of the thesis is organized as follows. First, the related works of traversability assessment, machine learning principles with the focus on transfer learning, and neural networks are reviewed in Chapter 2. Chapter 3 describes the Scarab II and Spot robots utilized in the experimental evaluation. In Chapter 4, the problem of traversability transfer between heterogeneous robots is presented, and Chapter 5 describes the proposed approach for the transfer of traversability assessment experience. The results of experiments that evaluate the proposed approach are shown in Chapter 6, and Chapter 7 concludes the thesis.

Chapter 2

Related work

This chapter covers the state-of-the-art research conducted on traversability assessment and machine learning topics. The topic of traversability assessment is covered in Section 2.1. The following Section 2.2 presents various machine learning principles that are relevant to the subject of knowledge transfer. In Section 2.3, we examine neural networks with a focus on their transfer between models.

2.1 Traversability assessment

The main aim of traversability assessment is to support better path planning decisions, such as avoiding impassable terrain or optimizing the path for the specific needs of the robot. The traversability over single terrain may greatly vary between different robotic platforms, such as wheeled and legged ground vehicles. Moreover, robotic platforms can have different experiences when encountering the same terrain because of changed internal states such as remaining battery capacity, reduced torques, or damage received during the deployment. Correct estimation of traversability is essential for applications where the robot encounters different terrains requiring the robot to adapt to various and often dangerous environments. Such fields could be represented by extra-terrestrial exploration [1–3], search and rescue missions [4], and agriculture or off-road driving [5].

In [6] and [7] the authors provide a thorough overview of traversability assessment methods of mobile robots. However, we aim to review recent research focusing on neural network approaches. Traversability can be classified either in discrete classes, or as a continuous score, and the information can be collected using proprioceptive, or exteroceptive sensors. Exteroceptive sensing can utilize geometry or appearance methods based on the sensory equipment present on the robot, while the proprioceptive methods utilize sensors measuring the robot's internal properties.

The most straightforward solution to traversability assessment is using a binary classifier. It provides the necessary interface to describe the most crucial terrain property, whether the robot can traverse the terrain at all. For example, in [8], the terrain's height variation is utilized to decide about the possibility of the robot's terrain traversal.

The distinction of traversability into more classes is studied in [9]. The authors propose a visual classification method according to predefined terrain types such as asphalt, gravel, and grass. The selection of appropriate behavior according to the terrain type helps the robotic platform to efficiently traverse the terrain, which is shown in [10], where the authors adjust the walking pattern (gait) of multi-legged walkers according to the perceived terrain (e.g., faster gait on asphalt, slower and higher swinging gait on grass). The data needed to create the terrain classification can be obtained using either exteroceptive [11, 12] or proprioceptive sensing [13, 14]. Proprioceptive sensing utilizes measurements of the internal states of the robotic platform. In the field of ground vehicles, the measurements often involve battery state, and vibrations endured by the platform [13]. In [15] authors take advantage of measured immediate power consumption to compute the cost of transport over terrain. The calculated cost then helps to select the optimal locomotion primitive.

The proprioceptive sensing creates requirements on the endurance of the robotic platform because to measure its states, and the robot has to be capable of enduring them. The requirements on the endurance of the platform are reduced in [16] with the help of simulation,

2.1 Traversability assessment

which brings with itself a new set of challenges. In addition to the requirements on the body’s endurance, the proprioceptive sensing on its own is not capable of providing information about future terrain. The predictions about future terrain can be obtained using exteroceptive sensing [11], which accomplishes estimation of terrain traversability for the terrain robot might consider traversing. Such assessments can allow the vehicle to make more informed decisions in path planning.

The approaches to exteroceptive perception can be further divided into geometry, and appearance-based traversability analysis [6] and often use equipment such as LiDARs and RGB-d cameras. The authors of [2] examine the 3D information about geometry using statistical processing by dividing the obtained map into individual grid cells. The properties of cells in an area, such as minimal and maximal height, the variance of height, and slope, are utilized to evaluate various probabilistic functions which yield a traversability score. When considering the statistical processing, it is beneficial to model the uncertainty and error of the traversability assessment because deterministic models do not generalize well due to errors induced by inaccurate sensory measurements. The certainty of terrain perception can be modeled by computing the number of points present in a cell and their uniformity [17]. Besides, when studying the geometry of traversed terrain, it is helpful to observe more significant features like negative gaps or edges [18]. During the examination of the geometry of the terrain, simulations aid the model to achieve better results, as shown in [19], where generated height map is utilized to create the traversability assessment models. Besides, the difference in perceived traversability can arise when approaching a height map cell from various directions. E.g., a wheeled robot with limited power can consider a sloped cell untraversable when ascending. At the same time, the descent is accomplished with ease.

The appearance-based traversability analysis can be carried out by image-processing and classification methods. In [20], the authors propose a method to classify terrain using the visual descriptor Speeded Up Robust Features in addition to Bag of Visual Words. In [21], the semi-supervised approach utilizes Generative Adversarial Networks (GANs) [22] to assess the traversability. The authors create a network where they need to provide more positive examples than negative ones as such a dataset should be easier to obtain. Positive (traversable) samples are safer to collect since the robotic platform does not have to experience untraversable terrain. Architectures of neural networks achieving predictions about traversability on future paths are presented in [23] which create virtual images from the images captured along the already traversed path. The constructed virtual images are presented to GANs introduced in [21] to predict the traversability along the estimated path. In [3], a fully convolutional neural network that replicated the VGGnet [24] architecture is utilized to locate the best possible place for the rover to land by classifying multiple terrain types and has been used for the Mars rover mission.

It can be beneficial to combine multiple traversability assessment approaches to leverage the accuracy of traversability assessment. The most apparent combination is between geometry and appearance-based methods. Such fusion of approaches was already studied in [25] where independent traversability maps are created by LiDAR and 2D camera and later fused using Bayesian rule. If the aforementioned techniques are not sufficient additional sensors such as humidity, precipitation, or temperature sensors might be employed to provide some complementary information about the environment, such as in [26], where an UGV utilizes various sensors to detect water surfaces better.

We aim to assess the traversability using geometry and an appearance-based approach and collect the ground truth data from the proprioceptive sensors. The traversability assessment is carried out using convolutional neural networks. Thus, the topic of neural networks is further studied in Section 2.3.

■ 2.2 Machine learning principles

Several problems have to be addressed when developing machine learning algorithms to be deployed on robotic platforms. First and foremost, robotic experiments testing machine learning approaches require financial resources as the robotic hardware has to be obtained and maintained. Additionally, machine learning algorithms deployed on robotic platforms have to be able to handle large datasets and perform actions in high dimensional action space, e.g., in case of legged robots, each joint increases the number of *Degrees of freedom* (DOF) correspondingly. Described issues can not always be solved using simulators, as their results can vary in experiments carried out in the natural environment.

In this section, we review information about incremental and transfer learning, focusing on traversability assessment as those approaches aim to reduce the influence of presented challenges during robotic deployments. Incremental learning is a machine learning approach that utilizes an incoming stream of data points to enhance the performance of an algorithm and is often called online learning. Incremental learning's primary goal is to produce immediate results. It is helpful in scenarios where the initial data available is scarce or non-existent and frequent attempts at improvement are desirable. The artificial neural networks, decision trees, and support vector machines are the main approaches that can be adjusted to carry out incremental learning tasks. Additionally, the challenges mentioned above make robotics a field suitable for online learning. The robot should quickly achieve satisfactory results on crucial tasks such as traversability assessment to protect its hardware. An example of incremental learning's deployment in robotics is shown in [27] where the authors utilize Fast Incremental Gaussian Mixture Network to accomplish improved traversability assessment on a ground robot from an aerial scan.

Reinforcement learning builds upon incremental learning and aims to improve performance on a given task by repeatedly attempting to solve the problem [28] by maximizing a given reward function. The reward function is the main distinction from other incremental learning approaches. Reinforcement learning can reduce the amount of training data or remove the need for manual fine-tuning of the controller. For successful reinforcement of knowledge, it is necessary to define an appropriate reward function that incentivizes the robot to take a path leading to a solution to the task.

In addition to incremental learning, transfer learning is another method of improving assessments by sharing experiences between different units in a team. Transfer learning is defined as a machine learning approach to improve the knowledge in the target domain by the transfer from the source domain [29]. Usage of transfer learning aims to improve performance in the target domain by utilizing knowledge from the source domain. In [30] the authors reduce the size of the dataset in the target domain necessary to train the convolutional neural network, thus shortening the training time of the neural network.

In [16] authors deploy reinforcement and transfer learning to improve the traversability over challenging terrain of a four-legged robot by training a motion policy. The policy is trained in simulation and then transferred to a real robot. The training is split into two parts. First, the controller is presented with labeled data; thus, reinforcement learning allows the policy to achieve high performance quickly. Moreover, the controller is deployed in a simulator having the same sensing capabilities as in a real environment. To address the differences between simulation and real-world, contact and slippage values are estimated from the past measurements rather than provided by models, as the models of contact and slippage are often inaccurate.

The costly data collection, and various tasks and builds of the robots make robotics an interesting field for the deployment of transfer learning. Although testing only in simulators, authors, in [31] utilize transfer learning to propagate experience in different scenarios of robotic

2.3 Neural networks

soccer, where they propose a solution to transferring neural networks between tasks with different inputs and action spaces. In [32] humanoid robots observe human gestures and motions to replicate them later. Another utilization of transfer learning is demonstrated in [33], where a robotic arm is trained to reach a destination of a colored block. The transfer is carried out between robotic arms with different amount of joints.

Although both incremental and transfer learning improves performance in robotic domains, incremental learning is more useful in scenarios where it is desirable to improve performance on a single task and setup. In contrast, transfer learning aims to share knowledge between varying problems. Thus, transfer learning is suited for the different robots encountered in robotic teams, while incremental learning is ideal for tweaking individual tasks. However, in comparison with transfer learning, incremental learning induces higher requirements on the build of the robotic platform because incremental learning procedures generate improvements using trials and errors [34]. Thus, the topic of robot safety discussed in [35] should be considered. Nonetheless, when using transfer learning, a problem called negative transfer can arise. Negative transfer occurs if the knowledge from the source domain hinders the performance in the target domain.

In [36] we show a transfer learning approach, which utilizes a traversability assessment model predicting the cost of the observed terrain and certainty of the cost prediction. The transfer is achieved by first transferring the traversability assessment model between the robots and then comparing certainties of the predictions of the transferred and directly trained model, choosing the cost prediction with higher certainty. This thesis builds on the cost regressor proposed in our previous work. However, instead of combining assessments of traversability models, a layer is possibly added to the regressor during the transfer, and additional retraining is employed to accommodate the transferred traversability assessment model in the target domain.

■ 2.3 Neural networks

This section provides a background on neural networks within the context of transfer learning. Deep learning is a category of machine learning principles that aims to grasp non-trivial internal properties of raw input data. It is closely linked with artificial neural networks that imitate the layers of neurons in the human brain. Neurons in artificial neural networks are represented by simple mathematical operations such as addition or multiplication. The rise of computational resources has allowed for the development of neural networks in recent decades as they consist of many parameters. The development led to many deployments of neural networks in fields such as natural language processing [37] and semantic segmentation [38].

As the typical neural networks consist of various types of layers, we aim to present a brief description of the functions of individual layers to understand the proposed neural network architecture. The weights of the individual layers are parameters that influence the outcome of each layer and, in most cases, are updated during the training of the neural network, the exception being, e.g., the batch normalization layer. The channels of the input matrix can be likened to the color channels of an image, but in neural networks, they can fulfill much broader needs.

Convolutional layer utilizes a matrix sliding over the input matrix of the layer. For each desired position of the sliding matrix called convolutional kernel on the input matrix, the sliding matrix is piece-wise multiplied with the elements in the underlying submatrix of the input matrix. The resulting matrix is then summed into one element. The elements from all desired positions of the convolutional kernel construct the output matrix. The convolutional layer weights are the individual elements of the convolutional kernel.

Maxpool layer moves a sliding window over the input matrix and fills the resulting matrix with the maximal element present in the sliding window.

Fully-connected layer shown in Figure 2 consists of connections between all input and output elements, where each connection is weighted to determine the influence between input and output.

Dropout layer helps to prevent overfitting when training deeper neural networks. The dropout layer selects inputs to be set to 0 with the selected rate and scales the remaining ones to match the sum of unchanged inputs.

Global average pooling layer computes an average for each input channel, thus acting as a flattening layer.

Batch normalization achieves the zero mean and the standard deviation of 1, which prevents overfitting, helping to create deeper networks.

Activation functions are utilized in neural networks to induce nonlinearities necessary for the network to be susceptible to more difficult tasks than linear regression. Often come in the form of tanh or sigmoid function; however, in our work, we employed *Parametrized Rectified Unit* (PReLU) given by the following equation

$$f(x) = \max(ax, x), \quad (1)$$

where a is a learnable parameter during the training.

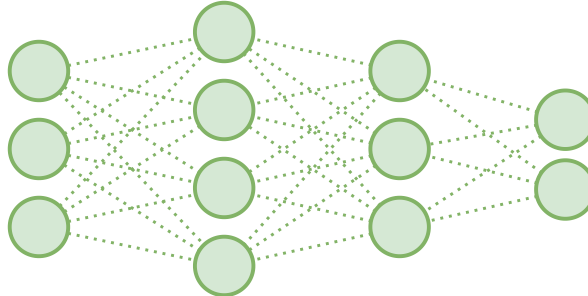


Figure 2: Fully connected layers of the neural network, where each neuron is represented by a dot of a layer pictured as vertically stacked neurons, are connected to all neurons of another layer.

Many of the constructed neural network architectures utilize convolutional layers, thus giving them the name of convolutional neural networks. The initialization of the weights is a crucial aspect when training the neural network, which might boost or slow down the training process. During the training of the network from scratch, initialization schemes, such as [39] are utilized, where the initialization function is created to cooperate with the activation function. However, a huge dataset is usually needed to train a network from scratch fully. Thus, the transfer of weights from similar tasks trained on a more extensive dataset can help achieve satisfying results with a smaller dataset [30]. After transferring weights between tasks, it is usually desirable to fine-tune the weights to better suit the needs of the task in the target domain.

There are various techniques when approaching the fine-tuning step. In the classification realm, the networks often comprise of feature extractor, which includes various convolutional

2.3 Neural networks

and max-pooling layers and classifiers consisting of fully-connected and output layers. In [40] the last fully-connected layer is reinitialized to comply with the possibly different classes in the target domain. If the source and target tasks vary greatly, an entire redesign of the classifier’s architecture is employed [41]. This way, just the feature extractor remains transferred from the source model.

During fine-tuning, selected layers of the neural network can be frozen, denoting that their weights are not updated [42]. Freezing the layers of the feature extractor allows us to fine-tune just the classifier when the wanted features are similar. When it is perceived that the feature extractor should be retweaked, it is beneficial to sequentially unfreeze the layers starting with the layer nearest to the classifier. Sequencing the unfreezing ensures that the layers extracting higher-level features are fine-tuned first, as it is more likely for the lower-level features to be shared between tasks.

The learning rate of the neural network is a factor that describes how much the network will accept the changes proposed by the gradient during the training. Thus, when fine-tuning, the learning rate is set lower, in case just slight corrections in weights are desired [41].

Due to the similarity to image processing, convolutional networks are utilized in the field of traversability assessment as described in Section 2.1, where the networks help to predict the traversability using exteroceptive sensors. Our focus is to implement and find the best possible setting for weight transfer. We assume that the transferred knowledge will boost the performance during traversability assessment for heterogeneous robots.

Chapter 3

Robotic platforms

Our experiments are carried out in real-world scenarios. Thus, the utilized heterogeneous robotic hardware is overviewed. We examine the build and equipment of the small hexapod robot Scarab II in Section 3.1, and the quadruped walker Spot in Section 3.2.

3.1 Scarab II

The Slow-Crawling Autonomous Reconnaissance All-terrain Bot [43] (Scarab II) shown in Figure 3 is a small hexapod walking robot utilized in our experiments. The miniature robot is suitable for experiments in confined areas where the larger robotic platforms would not fit.



Figure 3: Scarab II robot

The hexapod allows for single-manned deployment in the environment as the body, including the battery pack, weighs 4.8 kg and it occupies a circle of 55 cm in diameter. The diameter encompasses all possible foothold positions as the position of the legs changes during the deployment. The robot's body is manufactured by 3D printing using the PETG filament to allow for the eventual need for rapid prototyping as each experiment requires different equipment present.

The Scarab II possesses six legs, each of which consists of three joints whose motion is provided by Dynamixel XM430-W210-R servos and the rest of the structure of the leg is 3D printed. The legs of the hexapod result in 18 DOF. The tips of the legs are ended with end caps 3D printed from the flexible filament to avoid the contact of rigid PETG with smooth ground surfaces such as linoleum and ceramic tiles. The flexible end caps reduce the slippage on hard surfaces and allow for better traversability of sloped terrains. The robot utilizes an adaptive tripod gait [44] (walking pattern) to propel its body forward. The description of the

3.2 Spot

tripod gait in comparison with the alternative pentapod gait is presented in Figure 4. The tripod gait was selected as it allows for faster locomotion of the Scarab II than other types of gait, such as the reviewed pentapod.

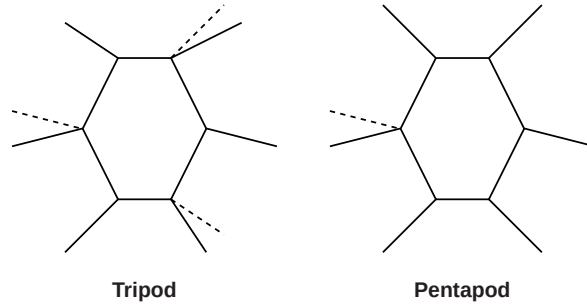


Figure 4: Comparison of tripod (left) and pentapod (right) gait. The tripod gait lifts three legs during the swing phase, denoted by the dashed lines marking the new locations of the legs. The configuration of the moving legs, as presented, allows the robot to remain statically stable during the motion of the whole body. The pentapod gait lifts at most one leg of the ground, leaving the remaining five in contact with the surface. Hence, the pentapod gait results in a slower traversal yet more stable body, suitable for the most challenging paths through the terrain.

The raised structure in the front of the robot's body, resembling the head of the robot, carries an assembly of Intel RealSense Tracking Camera T265 and RGB-d Intel RealSense Depth Camera D435, which collect information about surrounding terrain. The robot is equipped with an onboard Intel NUC computer with Intel Core i7-110710U Processor with frequency up to 4.70 GHz, and the whole robot is powered by two three-cell Li-Po 12 V, 5200 mAh batteries providing the robot with enough power to operate for roughly 1.5 h.

3.2 Spot

The Spot 5 is a four-legged robot resembling a dog developed by Boston Dynamics and is aimed at industrial customers. As industrial usage suggests, the Spot possesses a more durable body than the Scarab II. The Spot is 1.1 m long and 0.5 m wide and 0.6 m tall when standing. The total weight of Spot is 31.7 kg and is capable of carrying an additional payload of 14 kg.

Spot's four legs are actuated using 3 brushless motors, allowing 12 DOF combined. In contrast to the Scarab II, the actuator of the joint, equivalent to the knee of a dog, is located inside the body. Thus, the leg's weight is reduced, which results in a broader range of motions possible with the leg. The design of the Spot and its legs is capable of maximal forward velocity of $1.6 \text{ m} \cdot \text{s}^{-1}$, traversing slopes of up to 30° and ascending or descending most typical stairways. Like the Scarab II, the Spot's legs are ended with a rubber surface improving its traction on slippery terrain. During our experiments, we utilized the crawl gait for the Spot, where three feet always touch the ground, and one foot moves to a new foothold. The selection of such a slower crawl gait accomplishes safer navigation of challenging uneven terrain and assures the stable position of the robot.

The body of the Spot is equipped with five grayscale and depth recording Intel RealSense Depth D435 Cameras providing vision in all directions of a possible movement helping the controller decide about the occupancy of surrounding areas to assure obstacle avoidance. However, to build the environment model, we have used an Ouster OS0-128 lidar located on the attached payload to the robot. The raw data from the Ouster lidar is processed using Nvidia Jetson AGX Xavier, and the additional computations are carried out on an Intel Nuc

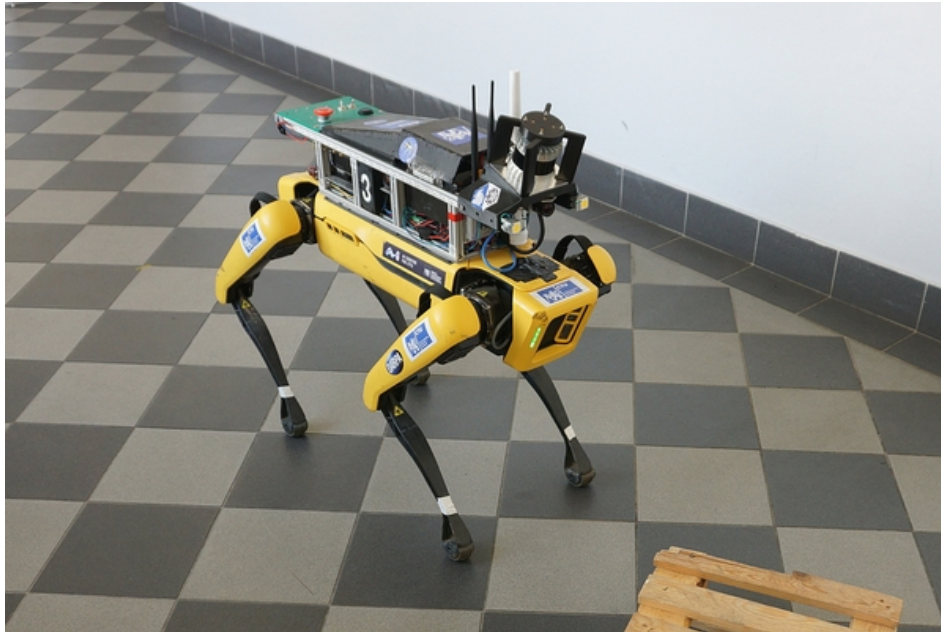


Figure 5: Spot robot with additional payload.

computer with Intel Core i7-110710U Processor located in the payload. The Spot's 58.8 V, 10 289 mAh batteries allow for roughly 1.5 h of walking time.

Chapter 4

Problem Statement

In this thesis, we address the traversability transfer learning, which is defined as follows. Let the various robots R_i perceive diverse terrains T_y during operational usage. Furthermore, let us assume that the properties of the terrain change. Thus, the terrains and their observations are not equal $T_1 \neq T_2$. As the terrains T observed by the robots are diverse, we describe the robot's experience with its traversability over the terrain T_y by the cost C , which expresses the robot's traversal experience with the terrain.

Let the environment be modeled as a 2.5D grid $\mathbb{W} \subset \mathbb{R}^2$ with cells v , where the size of the individual cell w is chosen to correspond with the size of the robot's body. Center of robot's footprint v^{robot} is discretised to be positioned onto the grid \mathbb{W} . Robot's paths ψ consist of a sequence of neighboring cells v_1, \dots, v_n visited by the robot and the collected costs C can be utilized to facilitate better path-planning decisions by finding a path with the minimal expected cost

$$\psi^* = \underset{\psi \in \Psi(v, v')}{\operatorname{argmin}} \sum_{v_i \in \psi} C_{v_i} \quad (2)$$

where Ψ is a set of all paths leading from v to v' . To enable path-planning on unvisited parts of the environment, it is needed to create a model r capable of assessing the cost C_i on previously untraversed parts of the terrains T_i observed using exteroceptive sensors

$$r(T_i) \rightarrow C_i. \quad (3)$$

Since the mobile robot traversability is considered too complex to assess directly from terrain appearance or geometry using a handcrafted function, the relation between measurements from proprioceptive sensors, which primarily help with the computation of cost C , and exteroceptive sensors providing the observations of the surrounding environment is learned from the collected terrain observations T_i accompanied by the cost measurements C_i . The costs are computed using proprioceptive sensors because of their ability to measure the external environment's direct influence on the robot's body. Hence, the goal is to find the most probable cost C_i given a terrain observation from an exteroceptive sensor T_i . The model is created using the collected dataset D_t and aims to minimize the *Root mean square error* (RMSE) between cost assessments on collected terrain observations T_i and the measured costs C_i

$$\min \sqrt{\frac{1}{n} \sum_{i=1}^n (r(T_i) - D(T_i))^2}, \quad (4)$$

where $D(T_i)$ returns the cost C_i measured by the robot.

Motivated by the creation of cost assessment model r to achieve better path-planning decisions, the robot R captures its observations of surrounding terrain T_i and assigns a value of traversal experience C_i with observed terrain into the dataset D_T^R . The utilized sensory equipment greatly influences the terrain observations T on the robotic platform R , i.e., some robotic platforms employ only a depth camera, while others support the depth images by information from a colored camera. Besides, each robotic platform possesses different morphology, which results in various perceptions of the surrounding terrain T . Thus, different

equations for the computations of cost C are utilized on each robotic platform, e.g., in the case of a small robot, it is more suitable to compare desired with achieved velocity, as the robot often gets stuck in taller grass. Whereas a bigger-legged robot would benefit more from costs C computed from the stability of the robot's body since tall grass does not slow down the locomotion of such a robot. The difference in morphology and cost computation methods highlights the need for various proprioceptive sensors measuring variables needed to calculate the perceived cost C . The robot stores information about observed and traversed terrain into a $2.5D$ grid \mathbb{W} . Observation T_i is positioned on the grid, and the maximal collected elevation is assigned as the 0.5 dimension. Furthermore, such a grid can be colored according to the viewed color when RGB cameras are present on the robot.

The larger deployments of robotic teams often involve heterogeneous robotic platforms, which possess different capabilities. Such behavior takes shape when two robotic platforms R_1, R_2 traverse through the same part of terrain T_j , however, their cost measurements are not equal $C_{j_1} \neq C_{j_2}$. We assume that an improvement of the predictions of perceived terrains by the model r can be achieved using already trained models from the different robotic platforms, as the robots could exchange information about terrain previously unobserved by them. The goal is to improve the performance by transferring the cost assessment model r_1 from platform R_1 to robot R_2 . Successful accommodation of model r_1 on platform R_2 would improve cost assessments on newly observed terrains relatively soon after the beginning of learning, assuming the learning time to be finite. Thus, after the model r_1 is transferred from platform R_1 to platform R_2 , it is essential to again retrain the transferred model to satisfy the optimization criterion 4, where model r_1 is utilized and dataset D collected on platform R_2 .

Hence, the goal of the transfer is to improve the prediction the following should be satisfied to achieve successful transfer

$$|r_1^t(T_n) - C_n| < |r_2(T_n) - C_n|, \quad (5)$$

where T_n is a new terrain observation, C_n is cost measured during the traversal of such part of terrain and r_1^t, r_2 are the utilized cost assessment models.

Chapter 5

Method

The chapter outlines the steps to transfer the knowledge of traversability between heterogeneous robotic platforms. The predictors of cost are trained on datasets collected during a robot’s terrain traversal. The datasets carry information about the robot’s experience with the surrounding environment, and the topic addressed in Section 5.1. Section 5.2 describes the creation of a cost assessment model to elevate the direct experience with perceived terrain to the surrounding terrain. Finally, the approach to the transfer of cost assessment model is shown in Section 5.3.

5.1 Dataset collection

During each data-collection run robot first stores the information from its interaction with the surrounding terrain traversed during human-operated deployment collected by the particular robot’s extero- and proprioceptive sensors. As our robotic systems are built upon the middleware Robot Operating System (ROS) [45], the utilized robots can collect and replay measurements from sensors mounted on every robotic platform. We utilized this feature to store the required measurements into `.bag` files. The storage into separate files enables us to process the measurements individually, which reduces the difficulty of the robot’s deployments.

The recorded `.bag` files are then replayed and a map of the surrounding environment [46] from the observations collected during deployment is created. The grid map of the surrounding environment consists of cells describing an area with the edge size of 7.5 cm, which corresponds with the leg of the smallest robot, and each cell contains properties of the observed terrain based on the carried exteroceptive sensors. Both the Spot and Scarab II store the maximal perceived elevation of each observed cell. The Scarab II’s camera is capable of collecting colored data. Thus, additionally `a` and `b` features of the `lab` color space are recorded for the observed cells. The `l` feature expressing lighting of the observed terrain is not utilized as it reduces the influence of varying lighting of the observed terrain. Along with the observations of the environment, the `hdf5` [47] file contains information about the robot’s trajectory accompanied by the calculated cost, which was computed using one of the following cost computation methods, where the Scarab II utilizes the Velocity, Slope and Difference methods. In contrast, the Angle method is employed by the Spot:

Velocity c_v - utilizes relative slowdown compared to the commanded velocity. The distance d covered in a fixed timeframe t results in achieved velocity v . The speed v can differ from the commanded velocity v_{cmd} by either being lower due to difficulties with some terrain properties such as tall grass. The ratio of achieved speed v and commanded velocity v_{cmd} expresses the cost c_{tmp} which is to be adjusted. The computation is summarized by the following equation

$$c_v = \frac{v}{v_{cmd}} = \frac{d}{v_{cmd} t}. \quad (6)$$

Slope c_s - the slope cost model first computes the angular distance in degrees from the leveled plane θ , and the resulting cost c_s is equal to the distance from the straight pose in degrees offset by 1 as on flat terrain robot has to make an effort to propel its mass forward. The equation

$$c_s = 1 + \frac{\theta}{2\pi} \cdot 360^\circ \quad (7)$$

results in the desired cost calculation.

Difference c_d - similarly, the slope cost model prepares the angular distance in degrees γ . However, the difference cost model computes the maximal angular distance of subsequent, thus articulating changes in the terrain’s slope. Note that the robot’s tilt is observed on both axis corresponding to the leveled plane when the robot is laid on flat ground. The difference cost model utilizes following equation

$$c_d = 1 + \frac{\gamma}{2\pi} \cdot 360^\circ. \quad (8)$$

Furthermore maximal cost $c_{max} = 10$ is utilized in addition to tanh function to shift the distribution of introduced costmodels $c_m, m \in \{v, s, d\}$ closer to origin. The equation

$$c_{am} = c_{max} \cdot \tanh \frac{c_m}{c_{max}} \quad (9)$$

is used to compute adjusted cost c_{am} for all Scarab II’s cost assessment methods. The Equation 9 helps to remove high values of cost, which in case of velocity computation method c_v occur when the robot gets stuck. The cost assessment methods are created to return strictly positive values of cost because the non positive perceived costs would result in infinite paths being the most advantageous. Thus, causing the robot not to reach its goal as wandering through terrain would be more favorable for him.

Angle c_a - the cost for Spot robot is computed as the absolute angle α of the two opposing legs from the flat surface e.g. the left front and right rear leg are considered opposing. The Spot’s cost model is expressed as

$$\alpha = 1 + \left| \tan \frac{d_z}{d_{xy}} \right|, \quad (10)$$

where d_z denotes the difference in elevation of both legs and d_{xy} the distance of the legs on a XY plane.

■ 5.2 Cost assessment model

The cost assessment model similar to the utilized in [36] is presented with $w \times w$ segments of the terrain observations T , where w is the number of cells of the observation window’s width, which is selected so that at minimum, the entire robot’s footprint is covered. The segment of terrain observation T includes possibly colored elevation grid based on the carried sensory equipment. The regressor r is constructed using neural networks, which are utilized to produce the desired predictions, as the task of cost assessment shares similarities with image processing. The architecture of the regressor is depicted in Figure 6 and is created using `TensorFlow` library [48], which provides the implementation of neural network layers. During the training, the neural network is presented with a set of terrain observations and corresponding perceived costs and is repeatedly asked to estimate the cost of the presented observations. To update the neural network’s weights we utilized the Adam optimizer. The mean absolute percentage error of the estimation is used as loss Λ to be minimized during the training, and is specified by equation

$$\Lambda = 100 \left| \frac{y_t - y_p}{y_t} \right|, \quad (11)$$

where y_t is the expected output of the neural network and y_p the predicted.

5.3 Knowledge transfer

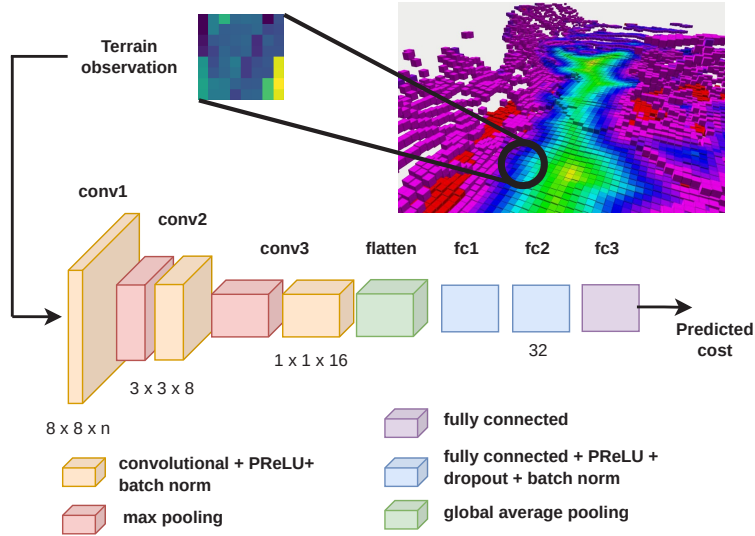


Figure 6: Architecture of the regressor, where n is the number of terrain observation's channels entering the neural network and width of observation window is $w = 8$.

5.3 Knowledge transfer

During the transfer of knowledge, which is visualized in Figure 7 the student receives the teachers cost assessment model in the form of the regressor. However, the teacher's model can be trained using differently shaped terrain observations. The width w of the input window can be prepared separately as we choose the width w accordingly to the robot receiving the model. Nonetheless, the observations of teacher and student can differ in the number of perceived channels caused by varying sensory equipment between the robots, which cannot be altered. A convolutional layer is added preceding the regressor if the shapes of the input data differ to accommodate the transferred regressor to the student's perceived data. We examine a varying number of input channels n . Thus, the convolutional layer consists of 1×1 convolutional kernel with the input and output channels corresponding to the perceived number of channels and the number of transferred regressor's input channels. Such a layer should be capable of grasping the relations between the inputs with various channels. After the convolutional layer is added, the newly obtained model is retrained using the dataset collected by the student to achieve better performance as the costs of teacher and student are assumed to be heterogeneous. Besides, during the retraining of the regressor, a specified number of layers l can be frozen, meaning the weights of those layers are not changed since it is assumed that the initial layers extract more general features, which are primarily similar between various data.

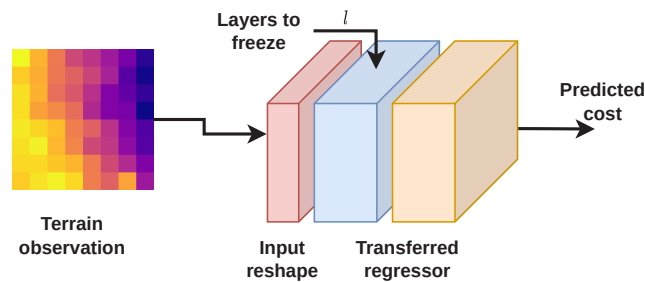


Figure 7: Setup used during the transfer of knowledge, l being frozen layers during the training.

Chapter 6

Results

This chapter reports on the experiments conducted to evaluate the proposed knowledge transfer method. The results of knowledge transfer between homogeneous robots with varying cost assessment models are demonstrated in Section 6.1. Such a scenario allows us to simulate the heterogeneities in the perception of cost while deploying only the small hexapod walker, which helps to verify the feasibility of the proposed method easier. The Section 6.2 depicts the transfer between fully heterogeneous robots.

6.1 Various cost assessment methods

The feasibility of the proposed method is first verified in a scenario where various cost assessment methods are utilized to simulate the difference in perception of heterogeneous robots. Scarab II collected the datasets in the Bull Rock cave near Brno, Czech Republic, as described in [36]. The datasets were collected in various parts of the cave system, which is shown in Figure 8, presenting various terrains to the traversing robot.

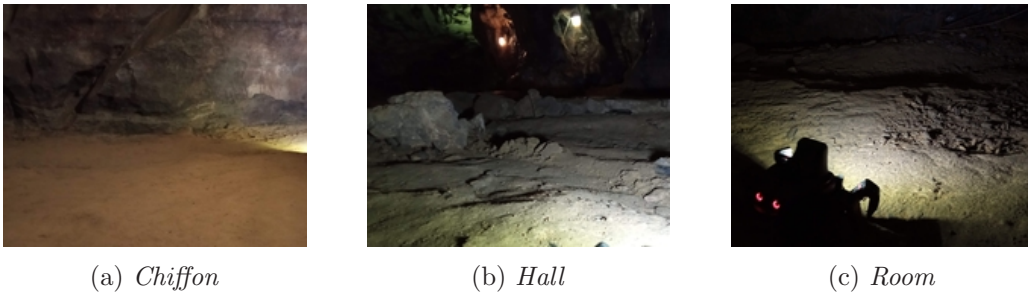


Figure 8: Different terrains of the Bull Rock Cave used in the evaluation of the proposed method.

Five scenarios are presented to test the transfer learning approach for each pair of cost assessment methods. The scenarios are prepared by randomly choosing five datasets from the collected dataset pool as testing data for the direct and transferred model. The testing datasets are removed from the datasets available to train the teacher’s and the student’s cost assessment models. From the remaining datasets, 12 are randomly drawn for the teacher and 5 for the student to create training datasets for their cost assessment models. All regressors are trained for 300 epochs during the training, and the width of the observation window is $w = 8$. Note that the teacher’s and student’s regressors can be trained using datasets collected in similar cave parts.

The results of the experiment are presented in Table 1 and show that the transfer of the model from the teacher to the student lowered the RMSE of predictions on the testing datasets. Freezing the initial four layers improved the assessment of cost. However, when eight layers are frozen, the RMSE did not as much as with the $l = 4$ frozen layers. The worse behavior of $l = 8$ frozen layers is probably caused by the fact that the additional frozen layers articulate the differences in cost perception.

In Table 2 the transfer from c_v to c_s is chosen to examine the influence of training for higher amount of epochs. Same randomly generated transfer scenarios are utilized as in

6.1 Transfer between slope c_s and velocity c_v cost assessment method

Table 1: Mean (std) of RMSE for 5 randomly generated scenarios for each pair of cost assessment methods. The columns of the table mark the performance of the utilized regressors, where the number signals the number of frozen layers l during the retraining of the transferred model.

Scenario	Direct	Transfer 0	Transfer 4	Transfer 8
$c_d \rightarrow c_s$	2.93 (1.47)	2.42 (0.53)	2.22 (0.20)	3.37 (2.43)
$c_d \rightarrow c_v$	2.42 (0.99)	1.59 (0.45)	1.63 (0.52)	1.63 (0.51)
$c_s \rightarrow c_d$	2.80 (1.18)	1.40 (0.17)	1.42 (0.27)	1.40 (0.24)
$c_s \rightarrow c_v$	1.84 (0.42)	1.53 (0.33)	1.54 (0.38)	1.68 (0.35)
$c_v \rightarrow c_d$	2.68 (1.32)	1.55 (0.20)	1.75 (0.51)	1.78 (1.00)
$c_v \rightarrow c_s$	3.76 (1.55)	2.23 (0.38)	2.08 (0.22)	2.55 (0.90)
Overall	2.74 (1.15)	1.79 (0.34)	1.77 (0.35)	2.07 (0.90)

Table 1. However, the cost assessment models are trained for 600 epochs and compared with the performance of regressors trained for 300 epochs. The results show that only the direct student’s model has improved thanks to the increased number of epochs. The RMSE of the transferred model slightly increased during the prolonged training, caused by the overfitting of training data. Overall the results suggest that the transfer helps to reduce the necessary number of training epochs by utilizing an already trained model as initialization of the training.

Table 2: Comparison of RMSE’s mean (std) between 300 and 600 epochs of $c_v \rightarrow c_s$ transfer

Epochs	Direct	Transfer 0	Transfer 4	Transfer 8
300	3.76 (1.55)	2.23 (0.38)	2.08 (0.22)	2.55 (0.90)
600	2.39 (0.45)	2.29 (0.32)	2.18 (0.34)	2.26 (0.29)

6.1.1 Transfer between slope c_s and velocity c_v cost assessment method

To better understand the knowledge transfer, we examine the transfer between student’s slope c_s and teacher’s velocity c_v cost assessment method in detail as those methods compute cost using dissimilar approaches. The robot is deployed in parts of the cave denoted as *Hall*, *Room*, *Chiffon* and *Stones*. In *Hall* and *Room*, the robot perceives a hardpacked surface, while the terrain in *Chiffon* presents the robot with a slightly sandy surface making the robot’s locomotion harder. Finally, a stony terrain is located in the part of the cave called *Stones* where the robot makes use of its legs. The student’s dataset is trained using one roll-out in all aforementioned terrain types. The teacher is trained on more roll-outs in specified terrains with the addition of a part called *Hall2*, which is similar terrain to *Hall* but located in a different part of the cave. The student’s dataset is overall a third size of the teacher’s, and hence is ideal to showcase the transfer of knowledge as there is a lot of information to be received by the student.

Both teacher’s and student’s direct baseline cost assessment models were trained for 300 epochs. After the teacher’s model was transferred to the student, it was tweaked using 300 epochs and $l = 0$ frozen layers to create the transferred model. The summary of the training carried out on the student’s platform for 300 epochs is present in Figure 9a where an initial boost during the training of the transferred model is visible. The decrement of cost during the tweaking of the transferred model stops when the training is prolonged in comparison with

6.1 Transfer between slope c_s and velocity c_v cost assessment method

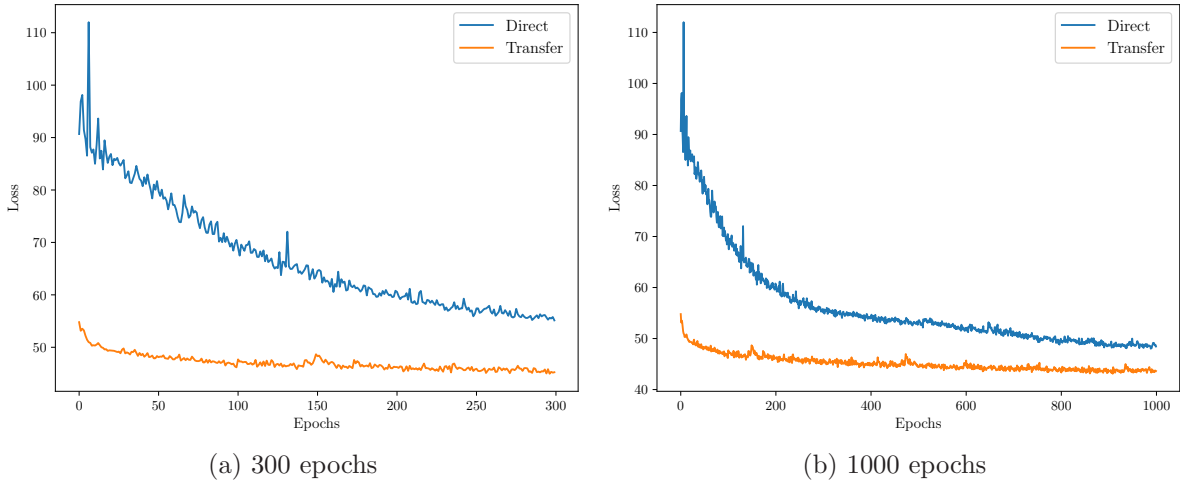


Figure 9: Progress of the cost assessment model's neural network training.

the direct model trained for 1000 epochs 9b. However, the transferred model still produces better losses than the direct model. The transferred and student's cost assessment models are tested using a dataset not used during the student's model training. The RMSE of regressors' predictions against the collected ground truth was 3.04 for the student's direct regressor and 2.09 for the transferred model from the teacher, resulting in improvement by 31.25%. The improvement in RMSE suggests that the transferred model produces better cost assessment predictions.

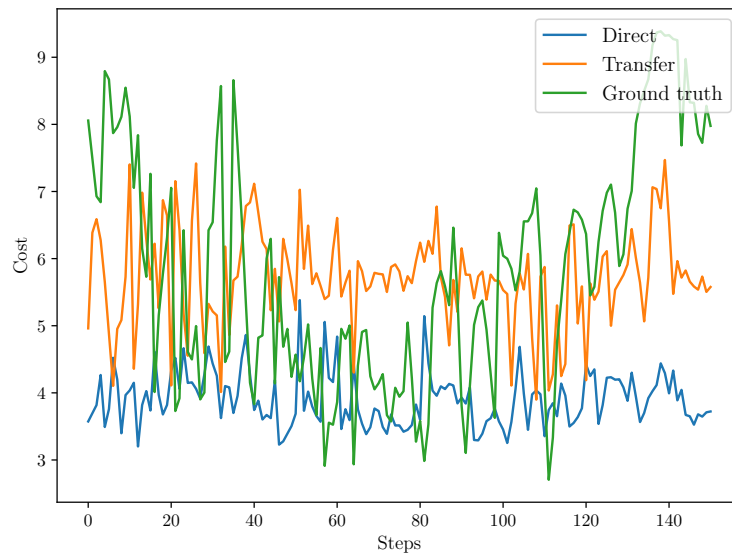


Figure 10: Predicted costs by student's and transferred neural network to be compared with Ground truth after training for 300 epochs.

As a demonstration, a single traversal in the *Hall2* part of the cave and resulting predictions are examined. Figure 10 shows the predicted costs by the student's direct and transferred model to be compared with the collected ground truth. Both direct and transferred models produce slightly noisy estimates, which can be expected since the perceived ground truth is also noisy. However, the transferred model matches the ground truth's overall difficulty more closely.

6.2 Transfer between Scarab II and Spot

In Figure 11 the cost prediction for the full observed maps by exteroceptive sensors is shown. However, the ground truth for the whole view range of the robot is not present as the robot traversed just a single path through the terrain. Thus, only manual evaluation was utilized to verify the feasibility of the assessments for future path planning. The difference in assessment is visible as the transfer model suggests a higher cost in locations where the elevation of the height map changes compared to the student’s model, which suggests that the transferred model produces better assessments than the direct model.

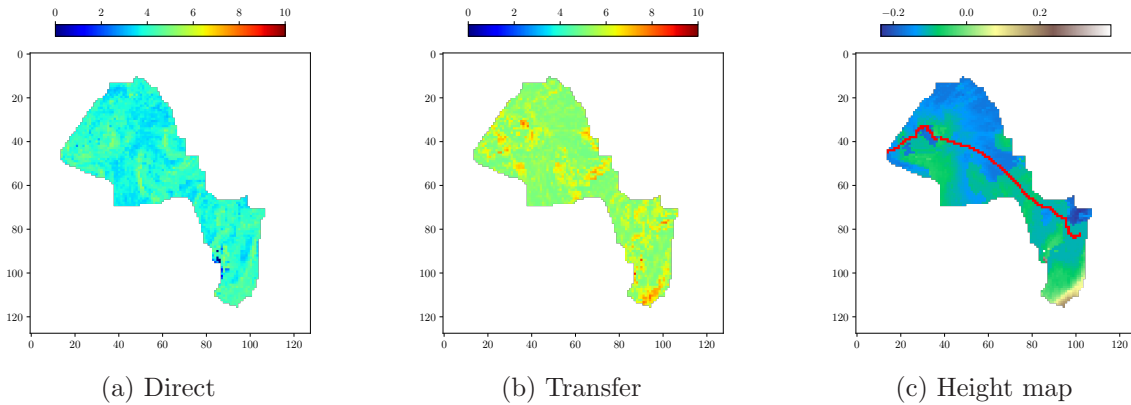


Figure 11: Cost assessments from exteroceptive sensors after training for 300 epochs and the perceived height map, where the path of the robot is marked using red color.

6.2 Transfer between Scarab II and Spot

To complete the collection of datasets for transfer learning between Scarab II and Spot, we gathered datasets using the Spot robot. The Spot’s datasets were added to the Scarab II’s collection from the Bull Rock Cave to create a setup for the possible utilization of the knowledge transfer. The Spot robot was deployed indoors and outdoors at the Czech Technical University campus at Karlovo náměstí. Indoors shown in Figure 5 the Spot moved through the halls of the building, and we modified the surfaces by partly covering them with artificial grass and spikes in the form of soundproofing material. Figure 12 captured the robot collecting a dataset outdoors, where the robot traversed various surfaces such as hard-packed soil, cobbles, and sloped grass. The Spot robot found all presented terrains relatively easy to traverse because the most challenging terrains were omitted for the robot’s safety. Additionally, the Spot’s datasets are, in general, longer than Scarab II’s as the robot is capable of faster movement. Thus, it is sufficient to use fewer datasets to train the cost assessment model. To test the presented method in heterogeneous scenarios, we examine the performance when transferring from Spot to Scarab II in Table 3 and the opposite direction in Table 4.

The transfer from Spot to Scarab II described in Table 3 is achieved using observation windows with a width of $w = 8$ cells, which is suitable for the smaller hexapod walker. Each transfer scenario comprising Spot and one of the hexapod’s cost models is evaluated using the mean and std of RMSE on five evaluation setups. For each evaluation setup, 5 datasets are randomly chosen to train the Spot’s teacher model. The Scarab II embodying the student receives randomly chosen 6 datasets, and the trained cost assessment models are tested on 5 randomly chosen datasets, which are different from the student’s training datasets. The regressors of the cost assessment models are trained for 300 epochs for both Scarab II and Spot.

The Scarabi II can observe the surrounding environment using a colored height map,



Figure 12: Spot during dataset collection.

while the Spot perceives just the height map. Thus, in each transfer scenario, we consider the student’s model to be capable of receiving input with both $n = 1, 3$ number of channels. When using the 3 channel version which perceives both the elevation and the a and b channels of the lab color space, a convolutional layer is added to accommodate the teacher’s model created by Spot having one input channel.

Table 3: Mean (std) of RMSE for knowledge transfer from Spot to Scarab II. The Transfer x signals the number of frozen layers is $l = x$ during the retraining of the regressor. The height map and color define the number of input channels as $n = 1$ for the resulting model perceiving just height map and $n = 3$ for the height map + color. The regressors are trained for 300 epochs for both Scarab II and Spot.

Scenario	Direct	Height map		Height map + color		
		Transfer	Transfer	Direct	Transfer	Transfer
		0	4	0	0	4
Spot $\rightarrow c_d$	1.92 (1.27)	1.25 (0.15)	1.26 (0.14)	1.65 (0.66)	1.33 (0.18)	1.34 (0.21)
Spot $\rightarrow c_s$	2.93 (0.88)	1.91 (0.48)	1.70 (0.13)	2.67 (1.44)	1.70 (0.13)	1.70 (0.18)
Spot $\rightarrow c_v$	2.72 (0.99)	1.79 (0.65)	1.67 (0.54)	2.24 (1.02)	2.93 (2.11)	1.88 (0.67)
Overall	2.52 (1.05)	1.65 (0.43)	1.54 (0.27)	2.19 (1.04)	1.98 (0.80)	1.64 (0.35)

The results in Table 3 show the performance of the trained models. The mean of RMSE for all transferred models is overall better than the direct model trained using just the student’s training data. However, the performance of the transferred model has not improved when modifying the teacher’s model to accept the colored height map collected by the Scarab II, although the direct model has improved when using $n = 3$ input channels. In the authors’ opinion, the likely cause is that the added convolutional layer could not sufficiently modify the input observation to achieve good performance in combination with the underlying transferred model.

To examine the transfer from Scarab II to Spot, we created transfer scenarios where each hexapod’s cost assessment method was transferred to Spot. The size of observations is set to $w = 16$ cells as the Spot’s body is larger than the hexapod’s. Each transfer scenario consists of 5 evaluation setups with 5 datasets for the teacher’s Scarab II model training, 1 Spot’s dataset for the student’s model training, and 3 datasets to test the resulting models.

6.2 Transfer between Spot and c_s

The student’s model can make decent predictions with a single training dataset as the Spot’s collected datasets are greater than the Scarab II’s and involve multiple types of terrain. The regressors’ neural networks are trained for 100 epochs for both Scarab II and Spot. To utilize Scarab II’s model with the three input channels created to assess the colored height map, we added a convolutional layer during the transfer of the model to the Spot.

In Table 4 the results show the improvement in the mean of RMSE achieved when utilizing the transfer learning. The transfer of Scarab II’s model perceiving color has achieved the best performance on the testing dataset. The authors suppose that the added convolutional layer and increased number of channels help the model to better grasp the underlying structure of the terrains’ traversal properties. Note that only one direct model is created for both the Height map and Height map + color scenario in each trail since both scenarios possess the same number of input channels n . Thus the direct models are the same.

Table 4: Mean (std) of RMSE for knowledge transfer from Scarab II to Spot. The Transfer x signals the number of frozen layers is $l = x$ during the retraining of the regressor. The height map and color define the number of input channels as $n = 1$ for the underlying model transferred from the Scarab II perceiving just the height map and $n = 3$ for the height map + color. The regressors are trained for 100 epochs for both Scarab II and Spot.

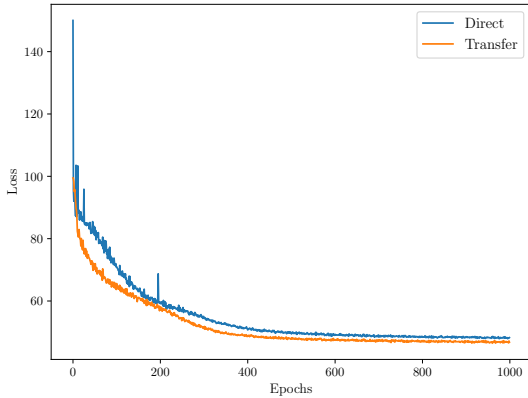
Scenario	Direct	Height map		Height map + color	
		Transfer 0	Transfer 4	Transfer 0	Transfer 4
$c_d \rightarrow$ Spot	2.14 (2.32)	0.57 (0.44)	0.65 (0.35)	0.60 (0.08)	0.92 (0.38)
$c_s \rightarrow$ Spot	0.99 (0.76)	0.26 (0.18)	0.32 (0.23)	0.85 (0.18)	0.66 (0.07)
$c_v \rightarrow$ Spot	0.84 (0.43)	2.14 (2.97)	1.48 (1.96)	0.60 (0.13)	0.60 (0.17)
Overall	1.32 (1.17)	0.99 (1.20)	0.82 (0.85)	0.68 (0.13)	0.73 (0.21)

6.2.1 Transfer between Spot and c_s

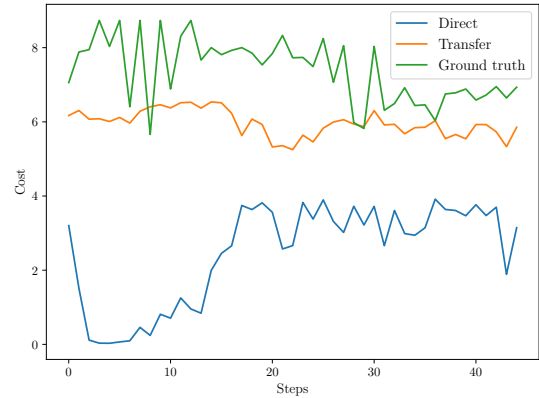
In this subsection, we created a scenario to show a more detailed overview of the knowledge transfer between the Spot and the Scarab II robot. The Scarab II utilizes the slope cost computation method c_s , and the width of the observation window is set to $w = 8$. The Spot’s teacher cost assessment model is trained using all datasets collected, while the Scarab II’s model is trained mainly on datasets from the *Hall* and the *Chiffon* part of the cave. The direct and transferred regressors are tested using datasets from the *Room* area of the cave. The regressors are trained for 300 epochs for both Scarab II and Spot, and the student’s direct model achieved an RMSE of 4.4, while the transferred and retweaked model had shown an RMSE of 1.86.

The progress of the regressors’ training after a prolonged training for 1000 epochs is depicted in Figure 13a. The initial boost received thanks to the transfer is present during the initial epochs of the training. After training for more than 400 epochs, the improvement in loss stops, and the models achieve almost similar loss until the end of the training. The RMSEs after training for 1000 epochs on the testing datasets are 2.05 and 2.01 for the direct and transferred model, respectively, favoring the transferred model, suggesting both models are somewhat overfitted to the training data after such prolonged training. Figure 13b shows the measured and predicted costs on a single dataset where we can observe the improvement achieved by the transferred model. However, even the transferred model cannot closely follow the oscillations of the ground truth perceived during the traversal through the terrain.

Figure 14 illustrates the cost assessments on all observed surfaces and the height map with



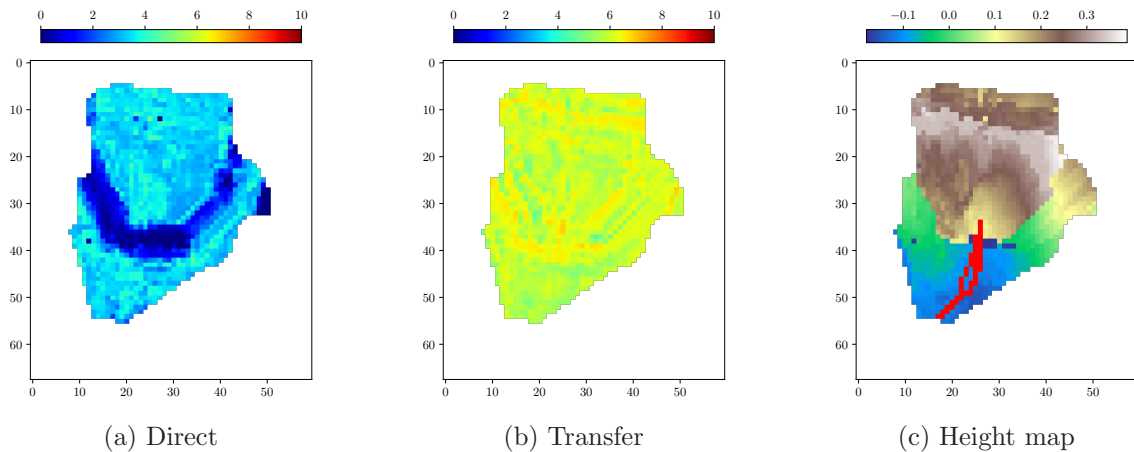
(a) Progress of the cost assessment model's neural network training for 1000 epochs.



(b) Predicted costs by student's and transferred neural network to be compared with Ground truth after training for 300 epochs.

Figure 13: Cost assessment model's results after transfer between heterogeneous robots.

a marked path of the robot during the collection of the dataset in *Room* part of the cave. However, we present only a manual evaluation of the predictions since the robot traversed only a tiny part of the observed environment. The transferred model assigns a higher cost to the terrain edge, while the student's model underestimates the traversal difficulty. Additionally, the transferred model suggests a more challenging cost in all areas of the observed environment, which likely resembles the actual perceived cost more closely. Thus, the transfer can presumably correct the predictions of the student's direct model.



(a) Direct

(b) Transfer

(c) Height map

Figure 14: Cost assessments from exteroceptive sensors after training for 300 epochs and the perceived height map. The path of the robot is marked using red color.

Chapter 7

Conclusion

This thesis presented a transfer learning approach capable of sharing traversability assessment models between heterogeneous robots. We construct a cost assessment model comprising a neural network regressor for each robot, which can assess traversability from a given terrain observation. The created cost assessment model is transferred from the teacher to the student to achieve the desired transfer between the heterogeneous robots and thus facilitate faster training of the traversability assessment model. During the transfer, a convolutional layer is possibly added to make the transferred model prepared for the number of channels perceived by the student's exteroceptive sensors. The transferred model is then further trained using the student's collected data with possibly freezing some of the neural network's layers. A dataset to evaluate the proposed method is created using data collected by the hexapod walker Scarab II and quadruped robot Spot, which differ in morphology.

The method is first tested on a scenario with various cost assessment models used by the Scarab II robot simulating the differences in traversability perception. We examine the possible improvements using different numbers of frozen layers during the retraining of the transferred model. The best results were achieved with 0 and 4 frozen layers, and the transferred neural networks outperformed those directly trained just on the student's dataset. The method's performance on a fully heterogeneous setup is tested by transferring knowledge between the Spot and the Scarab II. As the utilized platforms possess heterogeneous sensing capabilities, a convolutional layer is added to the created networks to accommodate the model's need for each robot's perception. The performance of transferred models with an additional convolutional layer is compared against the transfer setup without added layer. The networks with added convolutional layer performed better when transferring knowledge from Scarab II to Spot as they could utilize all information available more efficiently. However, both approaches outperformed the direct baseline training using just the student's dataset. We conclude that it is feasible to improve the assessment of traversability using the transfer of knowledge in the case of fully heterogeneous robots.

References

- [1] S. Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja, and K. Schwehr. Recent progress in local and global traversability for planetary rovers. In *Proceedings-IEEE International Conference on Robotics and Automation*, volume 2, pages 1194–1200, 2000.
- [2] D. B. Gennery. Traversability analysis and path planning for a planetary rover. *Autonomous Robots*, 6(2):131–146, 1999.
- [3] B. Rothrock, R. Kennedy, C. Cunningham, J. Papon, M. Heverly, and M. Ono. SPOC: Deep Learning-based Terrain Classification for Mars Rover Missions. In *AIAA SPACE*, 2016.
- [4] B. Cafaro, M. Gianni, F. Pirri, M. Ruiz, and A. Sinha. Terrain traversability in rescue environment. *IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2013.
- [5] A. Huertas, L. Matthies, and A. Rankin. Stereo-based tree traversability analysis for autonomous off-road navigation. *Proceedings - Seventh IEEE Workshop on Applications of Computer Vision*, pages 210–217, 2005.
- [6] P. Papadakis. Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Engineering Applications of Artificial Intelligence*, 26(4):1373–1385, 2013.
- [7] M. A. Bekhti. *Traversability Cost Prediction of Outdoor Terrains for Mobile Robot Using Image Features*. PhD thesis, Shizuoka University, 2020.
- [8] D. Langer, J. K. Rosenblatt, and M. Hebert. A Behavior-Based System for Off-Road Navigation. *IEEE Transactions on Robotics and Automation*, 10(6):0–7, 1994.
- [9] Y. N. Khan, P. Kamma, and A. Zell. High resolution visual terrain classification for outdoor robots. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1014–1021, 2011.
- [10] S. Zenker, E. E. Aksoy, D. Goldschmidt, F. Worgotter, and P. Manoonpong. Visual terrain classification for selecting energy efficient gaits of a hexapod robot. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics: Mechatronics for Human Wellbeing*, pages 577–584, 2013.
- [11] Y. Peng, D. Qu, Y. Zhong, S. Xie, J. Luo, and J. Gu. The obstacle detection and obstacle avoidance algorithm based on 2-D lidar. In *IEEE International Conference on Information and Automation - In conjunction with IEEE International Conference on Automation and Logistics*, pages 1648–1653. IEEE, 2015.
- [12] D. Ball, B. Upcroft, G. Wyeth, P. Corke, A. English, P. Ross, T. Patten, R. Fitch, S. Sukkarieh, and A. Bate. Vision-based Obstacle Detection and Navigation for an Agricultural Robot. *Journal of Field Robotics*, 33(8):1107–1130, 2016.

- [13] S. Martin and P. Corke. Long-term exploration tours for energy constrained robots with online proprioceptive traversability estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5778–5785, 2014.
- [14] F. L. Garcia Bermudez, R. C. Julian, D. W. Haldane, P. Abbeel, and R. S. Fearing. Performance analysis and terrain classification for a legged robot over rough terrain. In *IEEE International Conference on Intelligent Robots and Systems*, pages 513–519, 2012.
- [15] N. Kottege, C. Parkinson, P. Moghadam, A. Elfes, and S. P. Singh. Energetics-informed hexapod gait transitions across terrains. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5140–5147, 2015.
- [16] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):1–14, 2020.
- [17] D. Wettergreen, P. Tompkins, C. Urmson, M. Wagner, and W. Whittaker. Sun-synchronous robotic exploration: Technical description and field experimentation. *International Journal of Robotics Research*, 24(1):3–30, 2005.
- [18] A. Sinha and P. Papadakis. Mind the gap: detection and traversability analysis of terrain gaps using LIDAR for safe robot navigation. *Robotica*, 31(7):1085–1101, 2013.
- [19] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti. Learning Ground Traversability from Simulations. *IEEE Robotics and Automation Letters*, 3(3):1695–1702, 2018.
- [20] P. Filitchkin and K. Byl. Feature-based terrain classification for LittleDog. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1387–1392, 2012.
- [21] N. Hirose, A. Sadeghian, M. Vazquez, P. Goebel, and S. Savarese. GONet: A Semi-Supervised Deep Learning Approach for Traversability Estimation. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3044–3051, 2018.
- [22] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative Adversarial Networks: An Overview, 2018.
- [23] N. Hirose, A. Sadeghian, F. Xia, R. Martin-Martin, and S. Savarese. VUNet: Dynamic Scene View Synthesis for Traversability Estimation Using an RGB Camera. *IEEE Robotics and Automation Letters*, 4(2):2062–2069, 2019.
- [24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [25] J. Sock, J. Kim, J. Min, and K. Kwak. Probabilistic traversability map generation using 3D-LIDAR and camera. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume June, pages 5631–5637, 2016.
- [26] L. Matthies, M. Maimone, A. Johnson, Y. Cheng, R. Willson, C. Villalpando, S. Goldberg, A. Huertas, A. Stein, and A. Angelova. Computer Vision on Mars. *International Journal of Computer Vision*, 75(1):67–92, 2007.
- [27] M. Prágr, P. Čížek, and J. Faigl. Incremental Learning of Traversability Cost for Aerial Reconnaissance Support to Ground Units. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11472 LNCS, pages 412–421. Springer Verlag, 2019.

- [28] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*, 32(11):1238–1274, 2013.
- [29] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.
- [30] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal. Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials*, 157:322–330, 2017.
- [31] M. E. Taylor, S. Whiteson, and P. Stone. Transfer via Inter-Task Mappings in Policy Search Reinforcement Learning. *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems - AAMAS '07*, 2007.
- [32] N. Makondo, M. Hiratsuka, B. Rosman, and O. Hasegawa. A Non-Linear Manifold Alignment Approach to Robot Learning from Demonstrations. *Journal of Robotics and Mechatronics*, 30(2):265–281, 2018.
- [33] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine. Learning modular neural network policies for multi-task and multi-robot transfer. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2169–2176, 2017.
- [34] J. Hua, L. Zeng, G. Li, and Z. Ju. Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning, 2021.
- [35] M. Pecka and T. Svoboda. Safe exploration techniques for reinforcement learning – an overview. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8906, pages 357–375. Springer Verlag, 2014.
- [36] J. Zelinka, M. Prágr, R. Szadkowski, J. Bayer, and J. Faigl. Traversability Transfer Learning Between Robots with Different Cost Assessment Policies. In *2021 International Conference on Modelling and Simulation for Autonomous Systems*, pages 333–344. Springer, Cham, 2022.
- [37] P. Li, J. Li, and G. Wang. Application of Convolutional Neural Network in Natural Language Processing. In *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP 2018*, pages 120–122. Institute of Electrical and Electronics Engineers Inc., 2019.
- [38] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding Convolution for Semantic Segmentation. In *Proceedings - IEEE Winter Conference on Applications of Computer Vision*, volume 2018-Janua, pages 1451–1460, 2018.
- [39] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- [40] H. C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging*, 35(5):1285–1298, 2016.

- [41] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, 2016.
- [42] R. Ribani and M. Marengoni. A Survey of Transfer Learning for Convolutional Neural Networks. In *Proceedings - 32nd Conference on Graphics, Patterns and Images Tutorals*, pages 47–57, 2019.
- [43] M. Forouhar, P. Čížek, and J. Faigl. SCARAB II: A small versatile six-legged walking robot. In *5th Full-Day Workshop on Legged Robots at IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–2, 2021.
- [44] J. Faigl and P. Čížek. Adaptive locomotion control of hexapod walking robot for traversing rough terrains with position feedback only. *Robotics and Autonomous Systems*, 116:136–147, 2019.
- [45] Stanford Artificial Intelligence Laboratory et al. Robotic Operating System. <https://www.ros.org>, 2018.
- [46] J. Bayer and J. Faigl. Decentralized topological mapping for multi-robot autonomous exploration under low-bandwidth communication. In *10th European Conference on Mobile Robots - Proceedings*. Institute of Electrical and Electronics Engineers Inc., 2021.
- [47] The HDF Group. Hierarchical Data Format, version 5. <https://www.hdfgroup.org/HDF5/>, 1997-2022.
- [48] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems. <https://www.tensorflow.org/>, 2015.