# Terrain Classification for Small Legged Robots Using Deep Learning on Tactile Data

Hojung Choi
Stanford University
hjchoi92@stanford.edu

Rachel Thomasson
Stanford University
rthom@stanford.edu

## Abstract

*For small legged robots operating in diverse environments, the details of ground contact interaction play a huge role in the dynamics and efficiency of locomotion. Adapting the robot's gait parameters to suit the terrain improves performance significantly. In this work, we present a deep neural network classifier that predicts terrain given data from a capacitive tactile array attached to the C-shaped legs of SAIL-R, a small mobile robot. The classifier employs a 1D convolutional neural network (CNN) to extract relevant features from spatio-temporal tactile sensor data. We compare results to an SVM baseline and train on different subsets of hand-designed features to highlight how the classification results are correlated to the information provided. Results show that the use of deep learning bolsters accuracy significantly when compared to SVM and that including hand-designed features, particularly those that cannot be inferred from raw data, is beneficial.*

## 1. Introduction

During locomotion, the interaction between the ground and limb can have a significant impact on the efficiency and the robustness of mobility especially for small ground robots and animals. For instance, when running quickly on sand, the deformation of the sand causes dissipation of energy and increases the cost of transportation. On the other hand, the issue becomes slippage when moving on surfaces with low friction such as waxed tile. Thus, choosing the appropriate gait strategy during locomotion is critical in achieving efficient and robust locomotion for robots and animals.

In nature, small animals sense the type of terrain they are moving on using the myriad of mechanoreceptors and tactile sensing elements on their feet. For instance, the shovel snouted lizard utilizes the load information in its tendons when running on sand. Gait adaptation of ground robots using tactile sensing will be very useful for robots to reduce the cost of movement and increase maneuverability. However, for small legged robots, implementing such system has been difficult due to challenges such as small leg size, curved leg surface, and wiring.

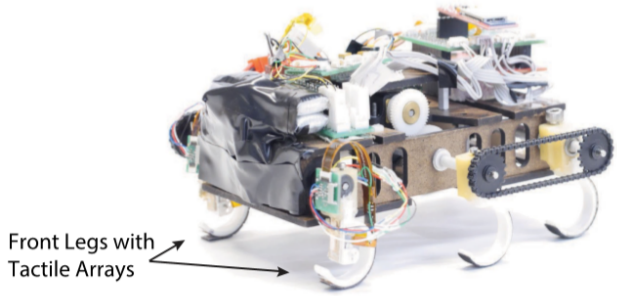


Figure 1. The SAIL-R robot

Recent advances in the capacitive sensing technology address the aforementioned difficulties. The development of small and lightweight capacitive to digital converters (CDC) enable the design of a compact tactile sensing system. The readily available commercial manufacturing of flexible circuits make them suitable for curved surfaces and minimizes the volume required for wiring.

In this study, we investigate the utility of employing deep learning as a means to classify terrain and terrain type based on information regarding ground-robot interaction and robot state. In particular, we use data collected from the SAIL-R robot (Fig. 1) built at the Biomimetics and Dexterous Manipulation Laboratory (BDML) at Stanford University that uses the capacitive sensing technology for collecting ground reaction force (GRF).

## 2. Related Work

The integration of force torque sensors and tactile sensors for robotic locomotion has been explored extensively over the years. The design and implementation of these sensors being a rather custom process. Various attempts have been made using different transduction methods and varying size depending on the application.

One of the earliest integration of tactile sensing for locomotion is shown in [5] where the ground reaction force data was used to plan gaits and maintain control in irregular terrains.

For larger robots where size and weight is less of a constraint, directly mounting bulky force torque sensors or tactile sensors have been explored. In [4], force torque sensors have been installed in the Honda ASIMO robot to plan and control gait with sensory feedback. They discovered that high acceleration and impact during contact resulted in higher noise of sensor data. Chuah and Kim [3] propose a lightweight and robust tactile sensor array for 3 axial force sensing for the MIT Cheetah robot.

Approaches not utilizing direct contact information also have been explored. Indirect sensing through reading acceleration and orientation with an IMU sensor has been investigated as shown in [2].

Classification of terrains using deep learning has also been explored in [1] where data collected from large 6-axis force torque sensors attached on shanks or legged robots were passed through a neural network. They used recursive neural network (RNN) and long short-term memory (LSTM) to capture temporal data and achieved an accuracy of $93\%$, but their sensors lacked spatial information on contact area which could also be useful.

Previous work performing classification on terrain class (high-friction, low-friction, deformable, granular) and gait adjustment with SAIL-R [9] used an SVM with 39 hand-designed features extracted from the ground reaction force data such as taxel peak force, and control input such as motor RPM. This has shown promising results on terrain class with an accuracy of $82.6\%$ on average, but suffered from low accuracy ($< 70\%$) on high-friction and low-friction terrains. By utilizing the spatio-temporal information encoded on the tactile data to its fullest extent, it can be possible to bolster the accuracy in these classes of terrains as well.

Convolutional neural networks (CNNs) that use one-dimensional convolutions have grown in popularity for tasks involving time-series data, in part, due to their computational efficiency over RNN structures [6]. We propose using a 1D CNN architecture instead of an SVM for terrain classification. We expect that the self learned features from a CNN will include temporal information from the tactile data which is not captured by the hand designed feature set used by the SVM.

# 3. Data

## 3.1. Data Collection

### 3.1.1 Tactile Sensor

The SAIL-R robot has a tactile sensor installed in each of its front legs. (Fig. 2) Each sensor is an array of 5 capacitive taxel ($5 \times 5mm$) that measures normal force, and

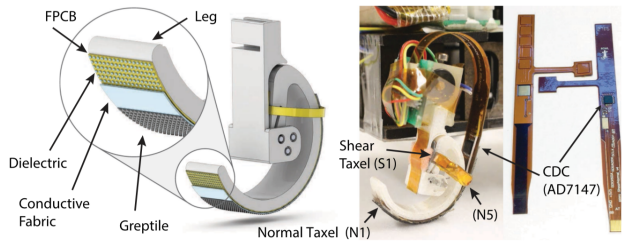


Figure 2. The tactile sensor array on the front leg.

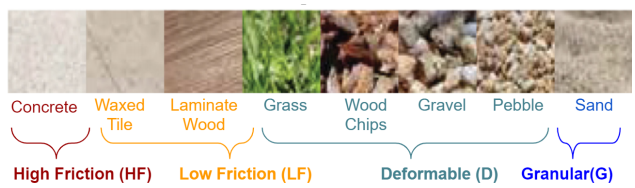


Figure 3. Outdoor terrains with varying physical properties for gait experiments and higher level classes.

another normal taxel sandwiched inside the hub of the C-shaped leg with pre-tension using a rubber band in order to measure shear force. Due to the circular shape of the leg, normal force and shear force measurement is decoupled as any force acting in the normal direction points towards the center of the of the circular shape where the shear taxel is positioned and does not produce moment. With shear force on the leg, the circular leg deforms such that it either applies compression or tension to the shear taxel. The sensor array which is fabricated on a flex circuit for conformity is glued on the 3D-printed C-shaped leg. A 16-b CDC (Analog Devices AD7147) reads the sensor array output at a frequency of 217Hz.

### 3.1.2 Terrain

The terrains for locomotion experiments were selected such that they include a variety of physical properties such as surface friction, stiffness, and dissipation as shown in (Fig. 3). They were grouped into 4 classes based on similarity of the properties: high-friction and high stiffness (HF), low-friction and low stiffness (LF), deformable (D), and granular (G).

## 3.2. Data Preprocessing

The continuous time-series data of the 6 capacitive taxels were segmented to reflect information for each step a leg made, as shown in (Fig. 4). From the data collected during locomotion, 33 hand designed features were derived as shown in Table 1. Some of the features are information inferable from the raw data such as 'Peak amplitude of sum' which is the peak amplitude of the sum of all normal taxels. The others are control inputs and robot states such as gait parameters or input current at peak force, which are not inferable from the raw sensor data.

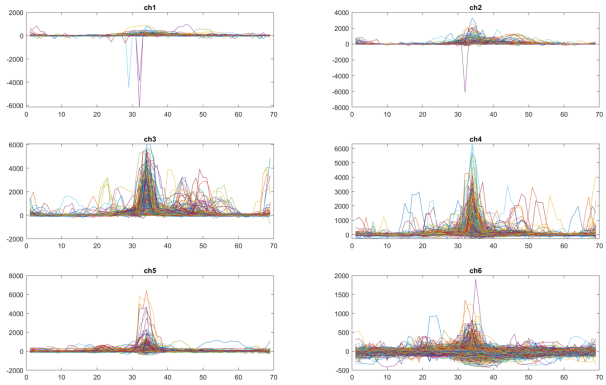


Figure 4. Superimposed plot of each sensor taxel readings. Ch6 is the shear taxel.

# 4. Methods

## 4.1. 1D Convolutional Neural Network Feature Extractor

In order to extract temporal information from the raw sensor reading and use it for classification, we have implemented a 1D convolutional neural network (CNN) (Fig.5). The 1D CNN was chosen among other structures known to consider time-dependent information such as the recursive neural network (RNN) or long short-term memory (LSTM) due to higher computational efficiency. The 6 individual taxel channel readings were transformed as a $6 \times 1 \times 69$ $[C \times H \times W]$ tensor that was used as input. The CNN is composed of 4 convolutional layers where the initial layer expands the channels from 6 to 24, and the other layers retain the channel size as 24. A kernel size of 1 by 5 was used to capture enough temporal information from the 69 sensor readings per channel. A stride of 1 and zero-padding was used to keep the width constant. Each convolutional layer was followed by a ReLU activation fuction and a max pooling reducing the size by half. Regularization was carried out by using dropout at a rate of 0.5 during training. This resulted in 1368 learned features which was then passed through a fully-connected layer. During the learning phase, Adam with a learning rate of 1e-4 was chosen as the optimizer with cross entropy loss.

$$Cross\ Entropy\ Loss = -log(\frac{exp(x[class])}{\Sigma_j exp(x[j])})$$

## 4.2. Terrain Classifier

Two fully-connected layers using ReLU activation convert the learned features into classification results. We investigate performance for both terrain classification and higher-level class prediction in order to examine the performance of using the CNN on the dataset, in contrast to [9] which uses SVM only for class prediction. When examining the effect of added hand-designed features, they were concatenated with the learned features before being input to the first fully-connected layer. Models including all 33 hand-designed features were investigated, as well as models including only a subset (i.e. control input), as many of the features can be inferred from the raw data.

| Index | Feature Name |
|---|---|
| 1 | Peak amplitude of sum |
| 2 | Area under curve of sum |
| 3 | Gait parameter – Motor RPM |
| 4 | Gait parameter – Ratio $T_s$ to $T_c$ |
| 5 | Average amplitude of sum |
| 6 | Gait parameter - $\omega_{slow}$ |
| 7-11 | Individual normal taxel force peak amplitude |
| 12-16 | Ratio of individual taxel force peak to peak sum |
| 17-19 | Min, max, average shear taxel force |
| 20 | Motor RPM at peak sum |
| 21 | Input current at peak sum |
| 22-23 | Input current average, range |
| 24-28 | Average normal force on individual taxels |
| 29-33 | Motor RPM at peak force for individual taxels |

Table 1. Tactile Machine Leaning Feature Set

# 5. Results

## 5.1. SVM Baseline

Before discussing the performance of our neural network based classifier, we present the results of a support vector machine (SVM) approach as a baseline for comparison. To mimic approaches for terrain classification from tactile data used in previous studies, namely [9] and [8], the SVM used a PUK kernel. The 33 hand-designed features reported in Table 1 were used as input to the SVM.

Figure 6 shows the output when labels are the four higher-level terrain classes and Fig. 7 shows the output when labels are the eight individual terrains. Similar to the results of [9], which used the same SVM approach on a superset of the hand-designed features used here, the lowest accuracy case is distinguishing high-friction from low-friction surfaces. The difficulty in separating high-friction and low-friction persisted in the individual terrain predictions, as concrete (HF) was often confused with waxed tile or laminate wood (LF). Additionally, accuracy in distinguishing the four individual terrains that comprise the deformable terrain class was relatively low.

## 5.2. Terrain Classification

Figure 8 shows how accuracy on the train and validation sets evolve as training progresses both for the terrain class classifier and individual terrain classifier. The trends show a sharp increase in accuracy in the first few epochs, then more marginal gains as training continues. While there is a gap

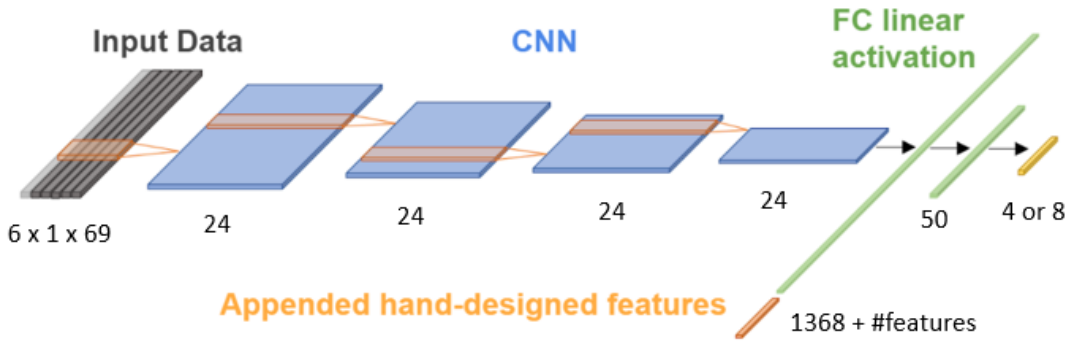


Figure 5. The neural net architecture.

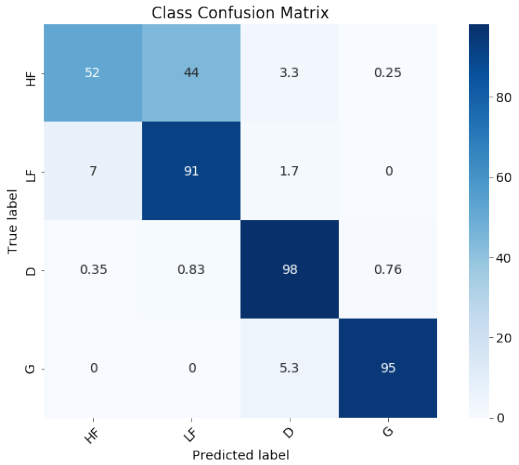


Figure 6. SVM Confusion Matrix (Class)

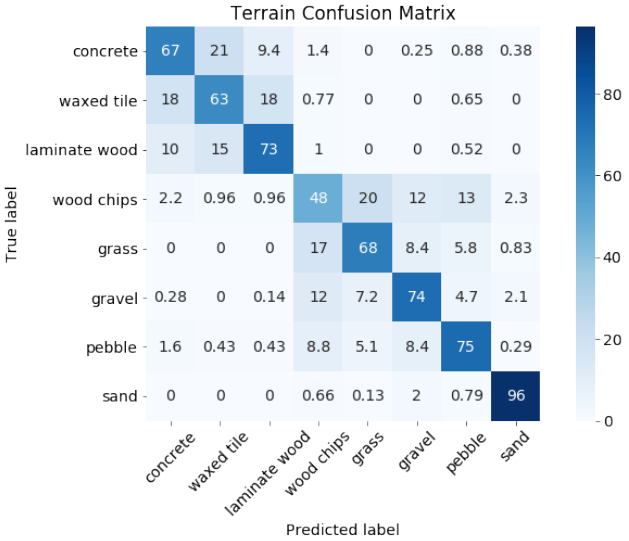


Figure 7. SVM Confusion Matrix (Terrain)

between the training and validation accuracies, we do not see overfitting to the extent that validation accuracy begins to drop.

The total accuracy over all classes for the SVM baseline

| Model | Class | Terrain |
|---|---|---|
| SVM Baseline | 84% | 70% |
| Raw Data | 87% | 69% |
| Raw Data + All Features | 97% | 80% |
| Raw Data + Control Input Features | 93% | 76% |

Table 2. Summary of Results

and each of the three final neural net models are reported in Table 2. For clarity, each of the listed model has two variants: one trained to predict high-level class and one trained to predict individual terrain. The only difference between the model architectures for these variants is the output final dimension of the last fully-connected layer. Figures 9 and 10 show the confusion matrices for the Raw Data + All Features models. Confusion matrices for the Raw Data and Raw Data + Control Input Features models can be found in the Appendix.

# 6. Discussion

## 6.1. General Performance

### 6.1.1 Comparison to SVM Baseline

It can be seen that using a deep learning method can improve on the results of the SVM baseline for the terrain classification task. The Raw Data + All Features model showed an increase of over 13% for terrain class prediction and 10% for individual terrain prediction. The most significant accuracy boost from the baseline was in distinguishing high-friction from low-friction surfaces, which was one of the main motivations in exploring deep learning for this application. While the SVM classifier only correctly classified high-friction surfaces about half of the time, our highest performing deep learning model achieved over a 90% accuracy on high-friction surfaces, though this remains the lowest performing class.

### 6.1.2 Implications of Dataset

A key reason that the most common confusion is between high-friction and low-friction for all models investigated is the data itself. As the class labels imply, the main distinguishing factor between high-friction and low-friction surfaces is the coefficient of friction. As noted in [9], for the terrains used here, the difference between coefficient of friction for high-friction ($\mu = 1.0$) terrains and low-friction ($\mu = 0.5$) terrains is relatively small. Additionally, the time-series tactile data is noisy and has low-resolution in the time-dimension. This fact in conjunction with the short contact time per step for hard surfaces leads to poor quality shear data, which intuitively is the most important channel for distinguishing coefficient of friction.

It can be seen that it is more common for high-friction to be mistaken for low-friction than vice versa. This may be due to a class imbalance in the data. While the data is balanced in terms of terrain, it is imbalanced in terms of higher-level terrain class as the high-friction and granular classes include only one terrain each while low-friction contains two and deformable encompasses four. Therefore, there are nearly twice as many low-friction samples as there are high-friction samples, which could explain the aforementioned observation.

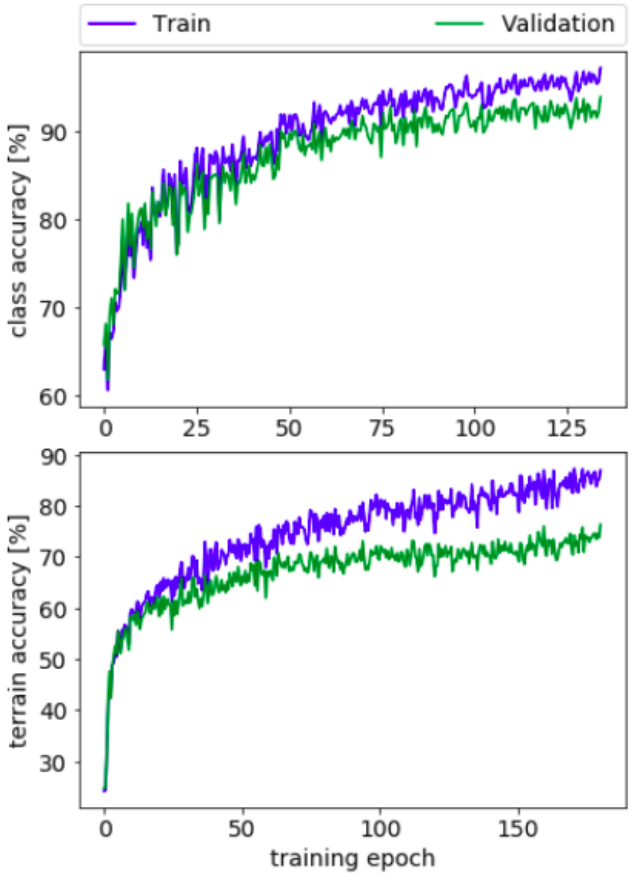


Figure 8. Training Curves

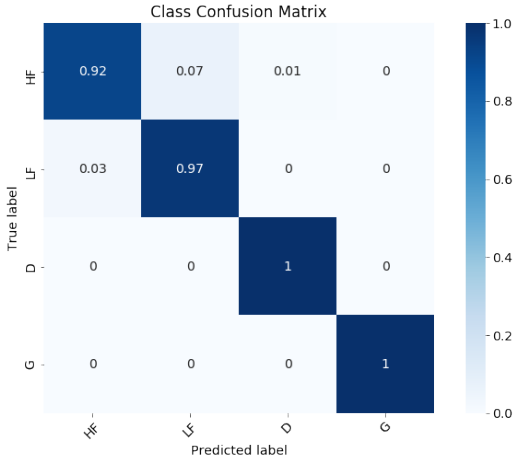


Figure 9. Raw Data + All Features Model Confusion Matrix (Class)

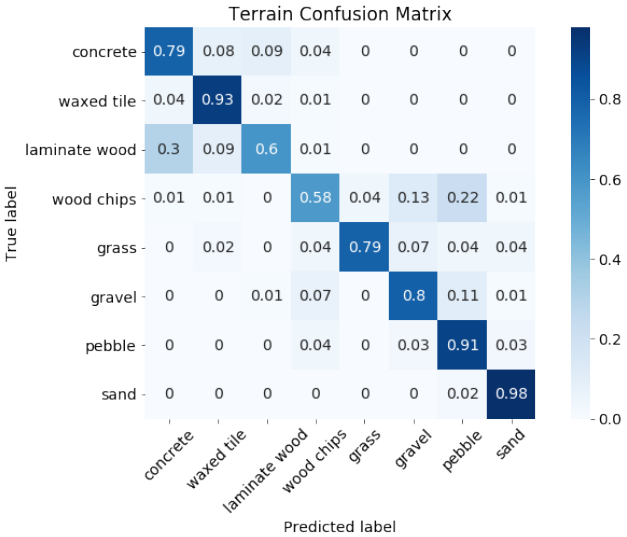


Figure 10. Raw Data + All Features Model Confusion Matrix (Terrain)

## 6.2. Effect of Hand-Designed Features

The features listed in Table 1 were hand-picked because they have an intuitive relevance to the terrain classification task. Therefore, it was expected that integrating all 33 of these features into our network would improve performance. Indeed, we see that the Raw Data + All Features model achieves the highest accuracy of the models explored, both for class and terrain. However, when employing this method for real-time classification it may be undesirable computationally to compute these features for each step. The authors of [9] reduced their feature set for real-time classification for this same reason. Moreover, a majority of the features (e.g. peak amplitude of force, average normal force on the taxels, etc.) can be directly inferred from the time-series sensor data and a well-performing 1D

CNN feature extractor should produce features of similar relevance from the raw data alone.

Using the same CNN feature extractor architecture but performing classification only using learned features, we see from Table 2 that accuracy drops down to roughly the performance of the SVM baseline. This is unsurprising as a select number of the features encode crucial information that cannot be inferred from the tactile data. The most notable of these are the gait parameter control inputs used when collecting the data. For several of the individual terrains, data was collected from trial runs of the robot using different gait parameters. Two samples on the same terrain but with different gait parameters could result in inconsistent raw data as the robot is moving differently as it traverses the terrain. These inconsistencies between samples of the same class, without knowledge of the gait parameters used, could cause confusion for the classifier. Referring again to Table 2, it can be seen that including just the gait parameter control inputs (3 features of the 33 total) improves accuracy significantly.

### 6.3. Terrain Class vs. Individual Terrain

The task of predicting terrain is more challenging than that of predicting terrain class, as terrains within the same higher-level class share a great deal of similarities. All models achieved lower total accuracy when predicting terrains than classes. Also, it was observed that models trained using individual terrain labels were more prone to overfitting than models trained on classes, which can be seen in Fig. 8.

As concrete is the only terrain in the high-friction class, one would suspect that as high an accuracy should be achievable for concrete as for high-friction. Yet unintuitively, the best accuracy observed for concrete using the terrain classifier was well below the best accuracy observed for high-friction using the class classifier. The lack of transparency of neural networks makes it difficult to definitely state the reason for this unusual result. In fact, as these classifiers were trained separately (rather than asking the same classifier to predict both class and terrain), it could simply be that the training of the class model settled on a better local minimum than that of the terrain model. A longer training time may also benefit accuracy for concrete as, upon examining the training curves in Fig. 8, a full plateau may not have been reached by the time that training was halted. Another possibility is that the neural network architecture used was more suited for class prediction than terrain prediction and that hyperparameter tuning or increasing complexity may lead to better accuracy for concrete.

## 7. Conclusion and Future Work

The deep learning classifier presented, which uses a 1D CNN to extract features from time-series tactile data, is a promising candidate for real-time terrain classification on a running robot. It outperforms SVM on hand-extracted features, which has previously been used for the same application, by over 10% both for predicting terrain class and individual terrain.

To further improve upon these results in the future, new data on the SAIL-R platform could be collected at a higher sampling frequency, leading to increased resolution in the time-dimension. With higher resolution data, the 1D CNN may be able to produce richer features that encode phenomena such as slip, which would improve the classifier's ability to differentiate high-friction and low-friction surfaces.

While previous work with SAIL-R adapted gait parameters depending on terrain class, the performance gains for predicting individual terrain achieved here opens the door for further optimizing gait for specific terrains. Going one step further, it would be interesting to expand this classifier to predict not the terrains themselves, but properties of the terrain such as coefficient of friction and stiffness. This could lead to the development of a more generalizable framework for adapting the gait of small-legged robots traversing diverse landscapes.

In the future, it would be interesting to explore deep learning architectures that are able to take in time-series data of arbitrary length. While the 1D CNN structure is a computationally efficient way to make use of temporal information, architectures such as LSTMs [6] or increasingly popular Transformers [7] may be able to produce predictions mid-step whereas the current 1D CNN implementation requires a fixed length time-interval of data corresponding to one complete step.

## 8. Appendix

### 8.1. Additional Confusion Matrices

In the Results section we included confusion matrices for the Raw Data + All Features models. Confusion matrices for the Raw Data and Raw Data + Control Input Features are included here.

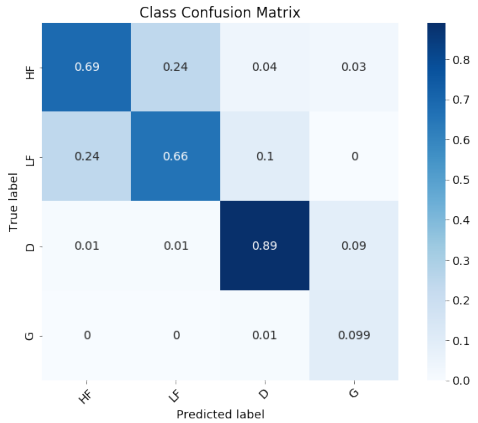


Figure 11. Raw Data Model Confusion Matrix (Class)

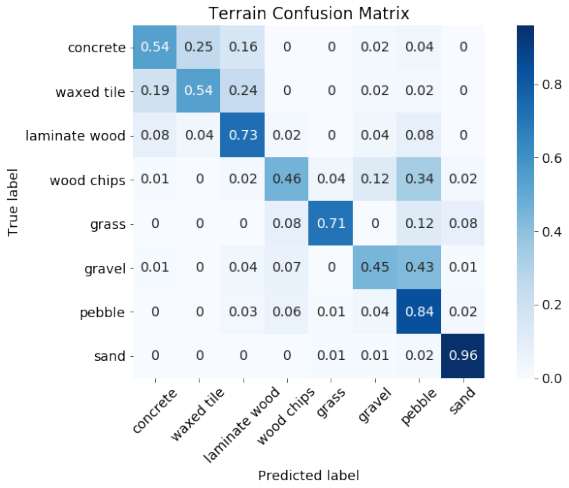


Figure 12. Raw Data Model Confusion Matrix (Terrain)

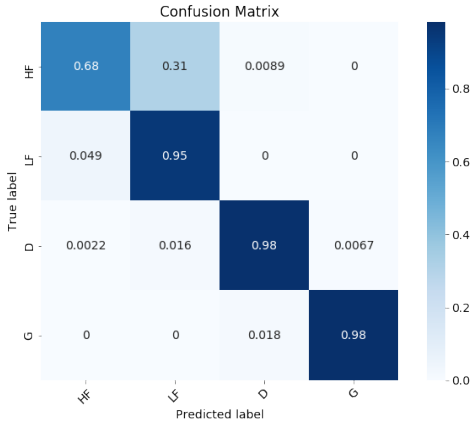


Figure 13. Raw Data + Control Input Features Model Confusion Matrix (Class)

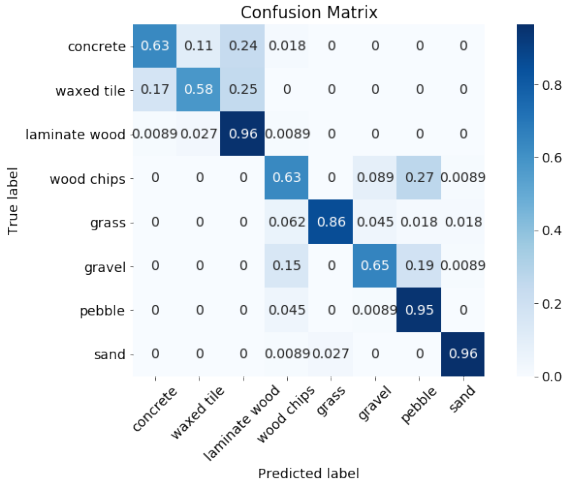


Figure 14. Raw Data + Control Input Features Model Confusion Matrix (Terrain)

## 9. Acknowledgements

Much thanks to Alice Wu and Tae Myung Huh for providing the tactile dataset for SAIL-R as well as their SVM implementation, which was used here as a baseline.

## References

[1] J. Bednarek, M. Bednarek, L. Wellhausen, M. Hutter, and K. Walas. What am i touching? learning to classify terrain via haptic sensing.

[2] W. Bosworth, J. Whitney, S. Kim, and N. Hogan. Robot locomotion on hard and soft ground: Measuring stability and ground properties in-situ. *Int. Conf. Robot. Autom.*, pages 3582–3589, 2016.

[3] M. Chuah and S. Kim. Enabling force sensing during ground locomotion: A bio-inspired, multi-axis, composite force sensor using discrete pressure mapping. *Sensors J*, 14:1693–1703, 2014.

[4] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. *Int. Conf. Robot. Autom.*, 2:1321–1326, 1998.

[5] S. Hirose. A study of design and control of a quadruped walking vehicle. *Int. J. Robot. Res.*, 3:113–133, 1984.

[6] F. Karim, S. Majumdar, H. Darabi, and S. Chen. Lstm fully convolutional networks for time series classification. *IEEE access*, 6:1662–1669, 2017.

[7] J. Oh, J. Wang, and J. Wiens. Learning to exploit invariances in clinical time-series data using sequence transformer networks. *arXiv preprint arXiv:1808.06725*, 2018.

[8] X. A. Wu, T. M. Huh, R. Mukherjee, and M. Cutkosky. Integrated ground reaction force sensing and terrain classification for small legged robots. *IEEE Robotics and Automation Letters*, 1(2):1125–1132, 2016.

[9] X. A. Wu, T. M. Huh, A. Sabin, S. A. Suresh, and M. R. Cutkosky. Tactile sensing and terrain-based gait control for small legged robots. *IEEE Transactions on Robotics*, 2019.