

Spectral Signature Mapping from RGB Imagery for Terrain-Aware Navigation

Sarvesh Prajapati¹, Ananya Trivedi¹, Nathaniel Hanson², Bruce Maxwell³, and Taşkin Padır^{1,4}

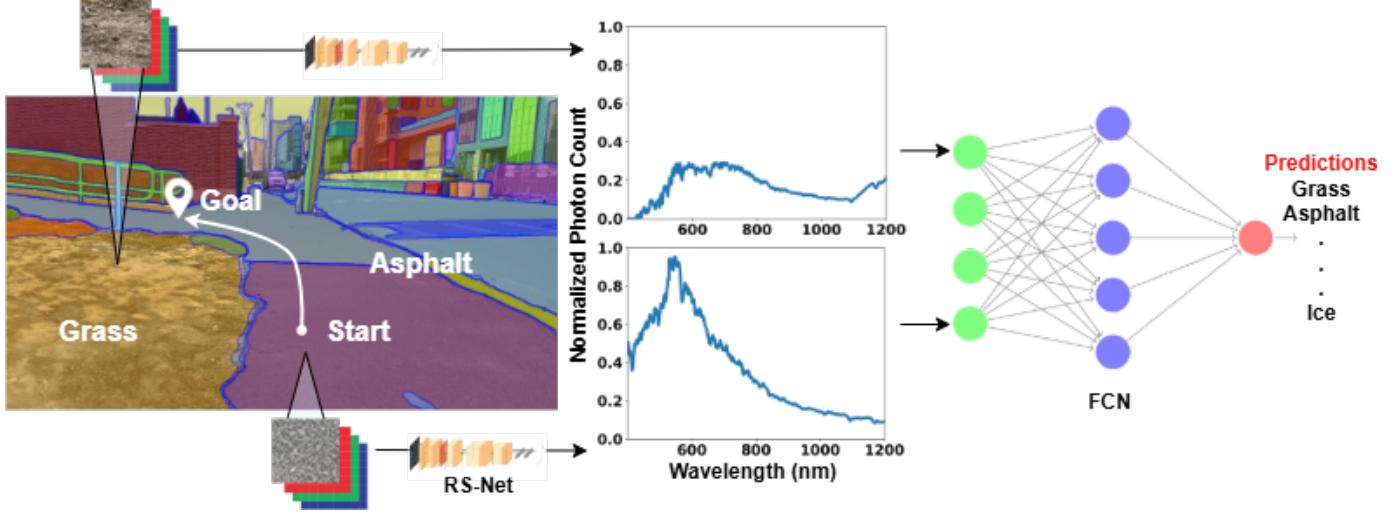


Fig. 1: Our approach takes RGB image patches (Left) as input and predicts their spectral signatures (Middle). These predicted spectra are then mapped to terrain classes by tuning the final fully connected layer (Right). The resulting terrain predictions inform navigation. The robot selects a path along the asphalt road, which is easier to traverse than the surrounding dense grass.

Abstract—Successful navigation in outdoor environments requires accurate prediction of the physical interactions between the robot and the terrain. To this end, several methods rely on geometric or semantic labels to classify traversable surfaces. However, such labels cannot distinguish visually similar surfaces that differ in material properties. Spectral sensors enable inference of material composition from surface reflectance measured across multiple wavelength bands. Although spectral sensing is gaining traction in robotics, widespread deployment remains constrained by the need for custom hardware integration, high sensor costs, and compute-intensive processing pipelines. In this paper, we present RGB Image to Spectral Signature Neural Network (RS-Net), a deep neural network designed to bridge the gap between the accessibility of RGB sensing and the rich material information provided by spectral data. RS-Net predicts spectral signatures from RGB patches, which we map to terrain labels and friction coefficients. The resulting terrain classifications are integrated into a sampling-based motion planner for a wheeled robot operating in outdoor environments. Likewise, the friction estimates are incorporated into a contact

force-based MPC for a quadruped robot navigating slippery surfaces. Thus, we introduce a framework that learns the task-relevant physical property once during training and thereafter relies solely on RGB sensing at test time. The code is available at <https://github.com/prajapatisarvesh/RS-Net>.

I. INTRODUCTION

Autonomous robots are increasingly deployed in everyday settings, ranging from self-driving taxis [1] and search-and-rescue missions [2], [3] to wildfire prevention [4], [5]. In such unstructured environments, reliable autonomy demands more than obstacle avoidance. It requires precise reasoning about how terrain properties influence motion. For example, vehicles must modulate braking on icy roads, and off-road platforms should bypass dense swamps to avoid entrapment. These scenarios show that perception must move beyond geometry and semantics toward reliable estimates of robot–terrain interactions.

Several off-road motion planning pipelines use RGB cameras to identify terrain from images [6]–[8]. In some cases, visually similar surfaces with different physical properties, such as ice on asphalt, may be mislabeled, leading to invalid traversability cost maps. Depth cameras and LiDAR are often used to estimate the ease of motion over a surface [9], [10]. However, the robot must first drive the terrain to create a dataset, which risks hardware damage and necessitates tuning specific to the operating site.

¹The authors are with Northeastern University, Boston, Massachusetts, USA. {prajapati.s, trivedi.ana, t.padır}@northeastern.edu

²Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Massachusetts, USA. nhanson2@mit.edu

³Khoury College of Computer Sciences, Northeastern University, Seattle, Washington, USA. b.maxwell@northeastern.edu

⁴Taşkin Padır holds concurrent appointments as a Professor of Electrical and Computer Engineering at Northeastern University and as an Amazon Scholar. This paper describes work performed at Northeastern University and is not associated with Amazon.

In contrast, spectral sensors offer a non-invasive way to estimate material properties. This is accomplished by leveraging distinct patterns of light absorption and reflection, known as spectral signatures, to characterize underlying material composition. These capabilities are finding use in robotics applications such as wildfire risk monitoring [3], manipulation [11], [12], and exploration [13]. By mapping spectral signatures to physical quantities such as moisture content, rigidity, or surface type, the same sensing stack can be repurposed across robots and environments with minimal changes to the processing pipeline. However, challenges such as custom mounts, calibration requirements, large datasets, and high sensor costs currently limit deployment at scale.

RGB cameras are inexpensive, widely available, and already standard in robotic perception pipelines such as object detection [14] and tracking [15]. Compared to hyperspectral systems, they are cheaper, lighter, and more power efficient, which simplifies integration on mobile platforms. Advances in deep learning [16] now allow RGB imagery to approximate measurements traditionally obtained from more information-dense sensors.

Motivated by this, we seek to retain the deployment advantages of RGB cameras while recovering spectral sensor features. We introduce RGB Image to Spectral Signature Neural Network or RS-Net, a deep neural network architecture trained on spectral data collected from diverse materials. It maps RGB image patches to their corresponding spectral signatures. These estimates are passed to a lightweight feedforward neural network whose weights are fine-tuned for the target physical property. We retrain this network once per task, enabling the same neural network architecture to perform terrain classification and friction estimation. Our entire inference pipeline runs at approximately 5 Hz, making it suitable for real-time robotic applications. Fig. 1 outlines the proposed architecture.

We validate our method in both simulation and hardware experiments. The terrain classification is used in a sampling-based motion planner for outdoor navigation of a skid-steer robot. Similarly, the friction estimates are integrated into a model predictive control (MPC) scheme for a quadrupedal robot operating on slippery surfaces. Finally, we also discuss how the proposed approach generalizes to other robots and additional physical properties relevant to off-road planning.

II. RELATED WORK

Inaccurate estimation of the coefficient of friction (CoF) can lead to slip and skid in mobile robots, resulting in degraded motion planning performance [17]. Many methods that rely on RGB cameras estimate friction via terrain classification. These approaches depend heavily on the quality of segmentation masks and are vulnerable to visual aliasing and mislabeling [6], [8], [18]. Moreover, models trained in one region, season, or camera setup may encounter significantly different illumination, exposure, or texture statistics at deployment, leading to poor generalization. Recent retrieval-augmented vision-language models avoid the need for paired image-friction labels by using language supervision for CoF estimation [19].

However, this comes at the cost of increased inference time. To address these limitations, we bypass semantic segmentation entirely and infer the friction coefficient from predicted spectral signatures. Our approach employs lightweight convolutional and fully connected networks, enabling inference at approximately 5 Hz. This supports real-time deployment while avoiding the computational overhead associated with retrieval-based language models.

Thermal sensing has been explored for friction estimation by identifying material categories under challenging lighting conditions [20]. Its practicality is limited by a short operating range of less than 0.5 m and a low frame rate, which constrain deployment on mobile platforms. Heat transfer tactile sensors estimate material properties through physical contact, which can lead to sensor wear or pose a risk of damage to the robot [21]. Acoustic sensing has been applied to surface classification by recording wheel-ground interaction sounds, but it remains highly susceptible to ambient noise [22]. Wheel encoder and IMU signals are commonly used to estimate robot pose through Kalman filtering frameworks [23]–[25]. These approaches rely on motion-induced signals to capture terrain interaction but require platform and terrain specific calibration, as well as sufficient on-ground excitation. This limits their generalization to new environments and robot configurations. In contrast to these approaches, our method relies solely on RGB input, offering sufficient sensing range for mobile robotic applications. It is fully non-invasive, eliminating the risk of wear or damage associated with contact-based sensors. Estimating material properties directly from spectral signatures removes the need for terrain-specific tuning and supports deployment across varied environments with no additional calibration.

Spectral and hyperspectral sensors capture material-specific reflectance by measuring how surfaces respond across a broad range of wavelengths [26]. This enables fine-grained discrimination between materials that appear visually similar [12], [27], [28]. However, real-world deployment faces several challenges [3]. These include high sensor cost, complex mechanical integration on mobile robots, and large memory requirements due to the high-dimensional nature of the underlying data. We address the limitations of spectral sensing by predicting spectral profiles directly from RGB terrain patch. This removes the need for expensive, high-bandwidth sensors while leveraging compact, low-cost RGB cameras already integrated into most robotic platforms. The reduced data dimensionality enables efficient storage and real-time inference, making the approach practical for large-scale, field-deployable systems.

Recent traversability estimation methods are beginning to move beyond pure geometry by incorporating richer terrain cues. TERP [29] constructs stability-based costs from elevation maps, while TRIP [10] predicts traversability from steppability analysis on dense 3D reconstructions. However, both these methods lack explicit modeling of material properties and can still misclassify low-friction surfaces. EVORA [9] learns traction distributions from exteroceptive sensing and incorporates these estimates into a CVaR-aware motion planner. However,

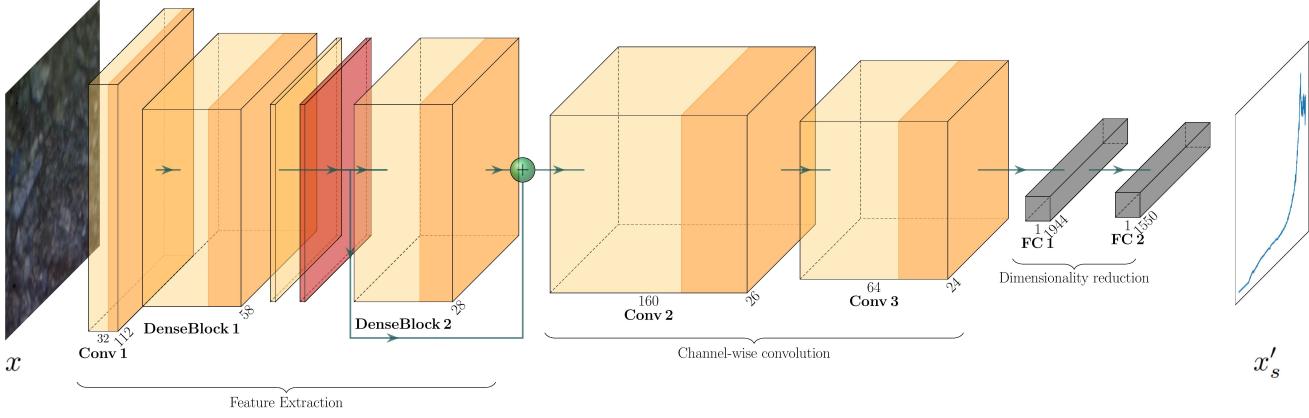


Fig. 2: Architecture to predict spectral signature x'_s from RGB image x .

it relies on supervised learning from physical interaction data collected over rough terrain, which limits its scalability. In contrast, we propose a unified approach that estimates planner-agnostic material properties by fine-tuning a lightweight fully connected head on top of spectral inference network. This allows the same backbone to be reused across motion planning tasks with minimal retraining.

III. METHODOLOGY

In this section, we first outline the neural network architecture used for mapping RGB inputs to spectral signatures. We then describe the output head that maps the predicted spectra to motion planning relevant quantities such as terrain classification and coefficient of friction. Subsequently, we explain the learning objectives and the optimization strategy used in the training process. Finally, we describe how the resulting predictions are integrated into existing motion-planning pipelines for both wheeled and legged robots.

A. RS-Net Architecture and Task-Specific Prediction Head

As shown in Fig. 2, RS-Net consists of three main stages: (i) feature extraction with DenseNet, (ii) feature fusion and channel reduction via shallow CNN layers, and (iii) spectral projection through fully connected layers. We explain each of these components below.

DenseNet for Feature Extraction: To extract global features from the input RGB image x , we use a pretrained deep neural network, DenseNet-169 [30]. Other DenseNet variants can be appropriately substituted, as the first two dense blocks share the same structure across different versions of DenseNet. We provide a high-level overview of the data flow through DenseNet below.

We use $G(\cdot)$ to denote generic convolution–pooling operations, which appear both in the initial stem and in transition layers within the network. Each dense block is denoted by $H(\cdot)$, where the ℓ th layer receives the feature maps of all preceding layers, $\mathbf{x}_0, \dots, \mathbf{x}_{\ell-1}$, as input:

$$\mathbf{x}_\ell = H_\ell([\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{\ell-1}]) \quad (1)$$

Here, $[\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{\ell-1}]$ denotes the concatenation of the feature maps produced by layers 0 through $\ell - 1$.

The image x first passes through the operation $G(\cdot)$, which produces the initial feature map.

$$x_1 = G(x) \quad (2)$$

x_1 is now passed sequentially through the first two dense blocks of the DenseNet backbone. The first block, which contains 6 convolutional layers, captures fine-grained local textures and low-level visual features. The second block, consisting of 12 convolutional layers, builds upon these features to extract higher-level contextual information.

In our implementation, the first dense block outputs x_7 , which is passed through a transition layer denoted $G(\cdot)$ to produce x_8 . This intermediate representation is then processed by the second dense block, whose output is x_{21} . The data flow through these operations is summarized as follows:

$$x_7 = H_7([x_1, x_2, \dots, x_6]) \quad (3)$$

$$x_8 = G(x_7) \quad (4)$$

$$x_{21} = H_{21}([x_8, x_9, \dots, x_{20}]) \quad (5)$$

We construct a fused global feature representation by concatenating two intermediate outputs: the max-pooled output from the first transition layer and the features produced by the second dense block. Let \oplus denote channel-wise concatenation. The resulting fused tensor is given by:

$$x_f = x_8 \oplus x_{21} \quad (6)$$

CNN Layers: The fused feature map x_f is processed by two lightweight convolutional layers for dimensionality reduction. The first layer reduces the channel count from 160 to 64 and downsamples the spatial resolution. The second layer further compresses the representation from 64 to 9 channels, producing a compact descriptor that both regularizes the model and reduces the computational cost of the subsequent spectral projection.

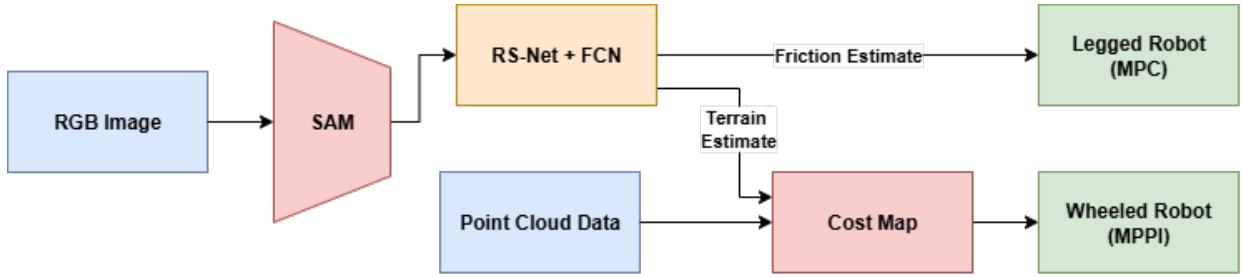


Fig. 3: Overview of our motion planning pipeline. SAM extracts image patches from the RGB input, which are passed through RS-Net followed by a Fully Connected Network to predict per-patch friction estimates or terrain labels. These predictions inform friction constraints or terrain-aware cost maps for downstream motion planning algorithms.

Fully Connected Layers: The 9-channel feature map is flattened and passed through two fully connected layers. The first projects the flattened input from 1944 to 1550 dimensions, followed by a second fully connected layer that maintains the dimensionality at 1550. Denoting this fully connected transformation by the composite mapping $S(\cdot)$, the final spectral output is given by:

$$x'_s = S(x_f) \quad (7)$$

Task-Specific Prediction Head: We append a lightweight fully connected network (FCN) after the spectral projection to map the predicted spectral profile x'_s to the desired planner-facing quantity, e.g., material class or friction coefficient. The spectral vector $x'_s \in \mathbb{R}^{B_s}$ is high-dimensional, where we use $B_s = 1550$ bands. To reduce this to a compact task-specific representation, we employ a multi-layer perceptron (MLP) with the structure $1550 \rightarrow 512 \rightarrow 128 \rightarrow K$, where K is the output dimension. For classification tasks, K denotes the number of classes and a softmax activation is applied to produce per-class probabilities. For regression tasks, $K = 1$ to output a scalar value such as surface friction. All intermediate layers use Gaussian Error Linear Unit (GELU) activation and dropout for regularization.

B. Neural Network Training Workflow

Training is conducted end-to-end, allowing gradients from the final task to flow back and influence the intermediate spectral representation learned by RS-Net. This coupling ensures that the DenseNet features are shaped not just to minimize reconstruction error but also to benefit the downstream task.

At inference time, only RGB input is provided. RS-Net predicts the spectral vector x'_s , which is then processed by the task-specific Fully Connected Network (FCN) to yield the required physical property. These predictions are made patch-wise across the image. The resulting patch-level outputs are then separately post-processed into a full-resolution output map, aligned with the input image.

We train on the VAST dataset released by Hanson et al. [28], which contains high-resolution RGB images of terrain surfaces, paired spectral reflectance profiles, and 9-DoF IMU data across 11 terrain types under varied lighting conditions. RS-Net is trained on six terrain classes: asphalt, brick, grass, ice, sand, and tile. It is also evaluated on five held-out

classes: carpet, concrete, gravel, mulch, and turf, to assess generalization.

We first pretrain RS-Net to learn an RGB-to-spectrum mapping using a mean-squared error loss. Given the ground-truth spectrum $x_s \in \mathbb{R}^{B_s}$ and the RS-Net prediction $x'_s \in \mathbb{R}^{B_s}$, the training loss is:

$$\mathcal{L}_{\text{spec}} = \frac{1}{B_s} \|x'_s - x_s\|_2^2 \quad (8)$$

After pretraining, the task-specific FCN maps the predicted spectrum x'_s to a planner-facing property. The full architecture is then trained jointly using a weighted combination of spectral and task losses:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{task}} + (1 - \alpha) \mathcal{L}_{\text{spec}}, \quad \text{with } \alpha = 0.7 \quad (9)$$

For classification tasks such as terrain labeling, $\mathcal{L}_{\text{task}}^{\text{terrain}}$ is defined as the standard cross-entropy loss between the ground-truth label y_i and the predicted softmax probability y'_i over K terrain classes. The ground-truth labels y_i are obtained from [28]:

$$\mathcal{L}_{\text{task}}^{\text{terrain}} = - \sum_{i=1}^K y_i \log(y'_i) \quad (10)$$

For scalar regression tasks such as estimating a friction coefficient, $\mathcal{L}_{\text{task}}^{\text{friction}}$ is defined as the mean absolute error (L1 loss) between the predicted and ground-truth values z'_n and z_n , over a batch of N samples. The ground-truth friction values z_n are obtained from [31]:

$$\mathcal{L}_{\text{task}}^{\text{friction}} = \frac{1}{N} \sum_{n=1}^N |z'_n - z_n| \quad (11)$$

We train for 50 epochs using the Adam optimizer [32] with a learning rate of 1×10^{-3} on a Lenovo Legion 5 Pro laptop with an Intel i7 12700H processor, 32 GB RAM, and an NVIDIA RTX 3070 GPU. Despite using a consumer-grade, lightweight GPU, the full training process completed in under 10 hours.

C. Integrating RS-Net into the Navigation Pipeline

To integrate RS-Net into the navigation pipeline, we preprocess RGB frames into material-specific patch inputs. Segmentation masks are first obtained using Segment Anything (SAM) [33] to isolate homogeneous regions. From each mask,

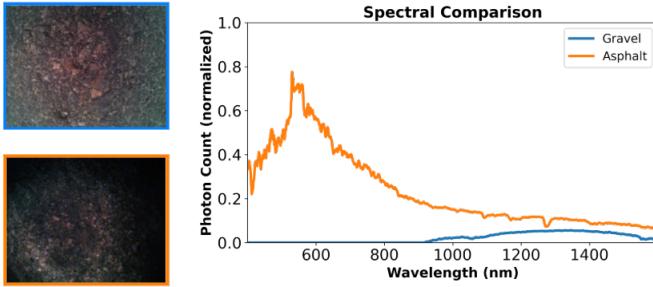


Fig. 4: Top: RGB patch of gravel. Bottom: RGB patch of asphalt. RS-Net predicts distinct spectral signatures for these visually similar surfaces.

we extract the largest square crop and apply minor recropping to standardize dimensions, ensuring compatibility with RS-Net’s fixed input size prior to inference. An overview of the perception-to-planning pipeline is shown in Fig. 3, illustrating how RS-Net predictions are converted into actionable inputs for downstream navigation in both wheeled and legged robots.

IV. EXPERIMENTS AND RESULTS

We perform three sets of experiments to evaluate RS-Net. First, we assess its ability to reconstruct spectral signatures from RGB images through standalone spectral estimation. Next, we integrate the friction prediction module into a quadruped robot’s control stack and evaluate its impact on navigation performance. Finally, we deploy RS-Net’s terrain classification outputs on a wheeled robot to guide outdoor navigation.

A. Evaluation of RGB-to-Spectral Mapping

A key motivation for our approach is to address the failure of appearance based terrain classification methods in distinguishing between materials that look similar but differ in physical properties. Fig. 4 illustrates this using two terrain patches, asphalt and gravel, that appear nearly identical in RGB imagery. Despite this visual similarity, RS-Net reconstructs distinct spectral signatures for each. Asphalt exhibits consistently higher photon counts than gravel, allowing clear separation in the spectral domain. This demonstrates that RS-Net learns to extract fine-grained visual cues beyond texture to infer material composition. This capability is essential for downstream planning and control tasks that require reliable reasoning about terrain properties in visually ambiguous environments.

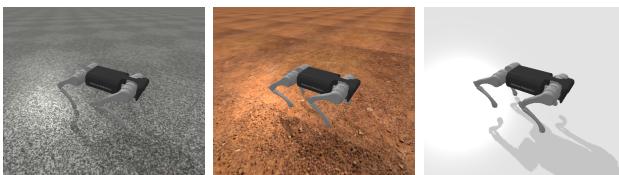


Fig. 5: Representative surface textures used in simulation. From left to right: asphalt, mud, and ice.

B. Quadrupedal Robot Navigation

As shown in Fig. 3, our method directly predicts the coefficient of friction μ from RGB input. We put this to test by deploying the resulting estimates in a quadruped robot navigating slippery surfaces. To prevent foot slippage, the ground reaction forces must satisfy the friction cone constraint:

$$\sqrt{f_x^2 + f_y^2} \leq \mu f_z \quad (12)$$

where f_x , f_y , and f_z are the contact forces along the respective axes. The shape of this cone, and therefore the feasible set of forces, is directly defined by μ [34], making accurate friction estimation critical for stable locomotion. The Linear Model Predictive Controller (LMPC) we use is based on [17]. The MPC parameters used for simulation and hardware experiments are shown in Table I. The controller tracks a desired center of mass height of 0.32 meters and a forward velocity of 0.5 m/s, while enforcing the friction cone constraint during optimization. We evaluate our method in both simulation and hardware. In each case, RGB images are segmented into patches using SAM. A friction estimate is computed for each patch, and the minimum predicted value is used to inform the LMPC.

TABLE I: LMPC parameters used for quadruped navigation.

Parameter	Value
Stepping Frequency	2.75 Hz
Foot Height	0.10 m
Planning Horizon	10 steps
Planning Timestep	0.05 s
Position Weight (Z)	50
Velocity Weights (X, Y)	10, 5
Angular Velocity Weights (X, Y, Z)	0.2, 0.2, 1.0
Roll and Pitch Weights	0.2, 0.2
Control Penalty Weights	1e-6

1) *Monte Carlo Evaluation in Simulation*: We use PyBullet [35] to simulate 1000 episodes for rigorous evaluation of our framework across varied terrains. In each trial, one terrain class is sampled from a predefined library consisting of asphalt, brick, grass, ice, sand, and tile. The corresponding surface texture is applied as the ground plane in the simulator, as shown in Fig. 5. A corresponding true friction coefficient $\mu \in [0.05, 1.0]$ is set based on the material. The rendered RGB image of the terrain is passed to the trained network to estimate $\hat{\mu}$, which is then used by the controller to perform locomotion. We compare controller performance under two conditions: one using a fixed friction value of $\mu = 0.5$, and the other using the estimated friction $\hat{\mu}$ predicted from RGB input. The goal is to evaluate whether learned friction estimates yield better locomotion performance across diverse surfaces. Success Rate is defined as the percentage of trials where the center of mass height remains above 0.25 meters throughout the episode. Average Slippage Ratio is computed as the ratio of tangential to vertical contact force magnitudes, with higher values indicating greater slippage. Normalized Tracking Cost measures how accurately the controller follows the desired

center of mass trajectory. Normalized Effort Cost reflects the total magnitude of contact forces applied, with higher values indicating less efficient control.

TABLE II: Results from 1000 Monte-Carlo Simulations.

Metric	Gait	LMPC	LMPC w/ RS-Net
Success Rate (%) \uparrow	Trot	79	100
Average Slippage Ratio \downarrow	Trot	0.20	0.12
Normalized Tracking Cost \downarrow	Trot	1.4 ± 0.3	1.1 ± 0.05
Normalized Effort Cost \downarrow	Trot	1.6 ± 0.2	1.1 ± 0.05

As shown in Table II, for the trotting gait, our framework performs better than simply using a fixed but small friction value. This improvement comes from the controller’s ability to adapt the friction cone based on the predicted terrain properties, rather than applying the same force limits across all surfaces. To further evaluate stability under low-friction conditions, we test the controller on an icy surface while tracking the center of mass height at 0.32 meters, as shown in Fig. 6. With active friction estimation via RS-Net, the robot maintains stability throughout the trial, whereas the fixed-friction baseline loses balance and falls within 3 seconds.

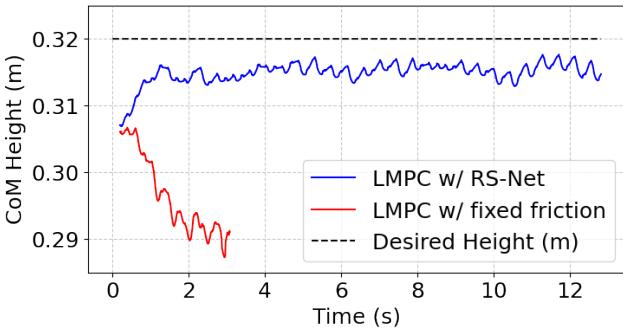


Fig. 6: Height tracking experiment on slippery ice surface.

2) *Real-World Evaluation on Slippery Terrain:* We conduct hardware experiments using a Unitree Go1 quadruped traversing an indoor track that includes a deliberately low friction section, created by applying cooking oil to a whiteboard surface. All trials are performed from a fixed initial pose under consistent lighting conditions.

As shown in Fig. 7 we evaluate two controller variants: one using a constant friction value, and the other using RS-Net for real time friction estimation. In the fixed friction case, the robot slips and fails on the low friction patch. RS-Net instead predicts higher friction on regular surfaces and lower friction on the slippery region, allowing the controller to adjust its force constraints accordingly and maintain stability.

C. Wheeled Robot Navigation

In off road navigation, certain surfaces such as asphalt are more favorable for traversal than others, such as dense grass. To account for this, we use the same RS-Net architecture as in the quadruped experiments, but fine tune only the final fully connected layer to perform terrain classification. The network

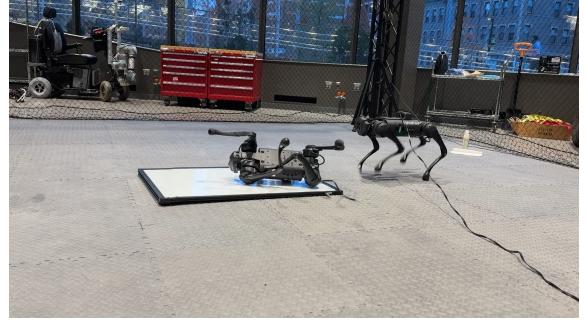


Fig. 7: Quadruped navigating from a high-friction rubber patch onto a low-friction oiled surface and back to rubber. Top: Controller with small but constant friction coefficient fails. Bottom: RS-Net-enabled online friction estimation stabilizes locomotion across changing surfaces.

outputs a discrete terrain label for each input patch, which is then used to construct a terrain aware cost map.

Once terrain labels are predicted, they are projected into 3D space using camera–LiDAR calibration [36], and ground level patches are extracted using Patchwork [37]. RS-Net runs at 5 Hz, updating terrain estimates in real time. The resulting terrain aware cost map is integrated into a sampling based motion planner based on Model Predictive Path Integral (MPPI) control [25]. MPPI evaluates 1024 candidate trajectories per planning cycle and selects the optimal one using a weighted sum of all trajectories.

As shown in Fig. 8, we use a 2D gridworld environment to tune the MPPI cost weights. A square material patch is randomly placed between a fixed start and goal location and treated as a high-cost region in the motion planner. The planner operates with a 5 second planning horizon, a timestep of $\Delta t = 0.05$ seconds, and control limits of 2 m/s linear velocity and 3 rad/s angular velocity.

We deploy our framework on a Clearpath Jackal robot in an outdoor setting with the start and goal placed across an asphalt route interspersed with grass patches. An onboard RGB camera streams terrain images to RS-Net, which assigns higher traversal costs to grass compared to asphalt.

We compare two planning conditions: (i) Baseline MPPI, which uses geometry-only costs and ignores material cues, and (ii) RS-Net + MPPI, which integrates a terrain-aware cost layer

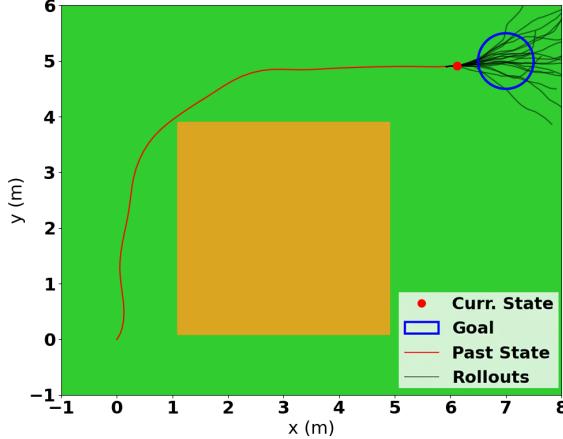


Fig. 8: Grid world simulation used to tune MPPI costs. Green indicates traversable space, brown denotes high-cost material patch, and black lines show sampled rollouts.



Fig. 9: The RS-Net-enabled wheeled robot navigates along the asphalt path, actively avoiding the grass region to reduce traversal risk.

based on RS-Net prediction. In repeated trials, both planners use identical MPPI hyperparameters. The baseline selects the shortest diagonal path through grass, minimizing time-to-goal but traversing less reliable terrain. In contrast, as shown in Fig. 9, the RS-Net-enabled planner avoids grass entirely, favoring longer asphalt-only routes. This yields comparable time-to-goal while significantly reducing the risk of getting stuck.

V. LIMITATIONS

Fig. 10 reports the spectral signature predicted by RS-Net for turf terrain, which was not included in the training set. The overall spectral shape is preserved, with peaks and dips occurring at similar wavelengths, but the predicted magnitude is consistently biased. This indicates a limitation of the current model. Although the trend is captured, the prediction is not precisely calibrated for unseen classes. To address this, we aim to explore alternate feature extractors beyond DenseNet [38] and perform additional tuning on larger and more diverse spectral datasets [39]. Additionally, we aim to explore pixel-wise spectral inference [40] in place of the current patch based

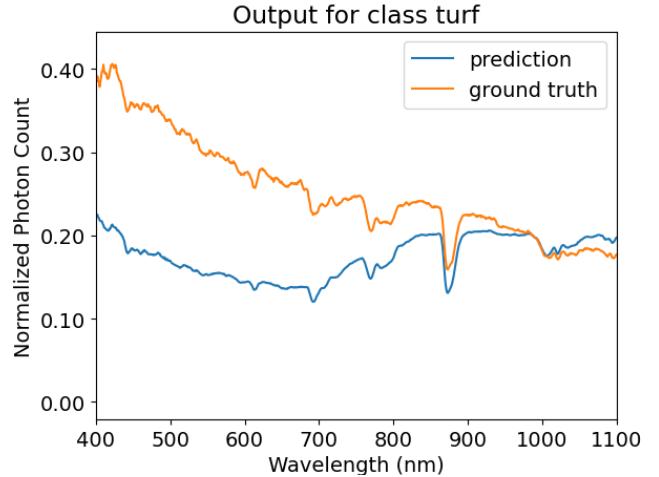


Fig. 10: Predicted spectral profile v/s ground truth data for a class not in the training set.

approach. This may lead to more fine grained predictions and improved spatial resolution of terrain estimates.

VI. CONCLUSION

We present a modular architecture for estimating terrain properties to support motion planning in mobile robots. This is achieved by training a neural network to predict spectral signatures directly from RGB images, enabling low cost sensing without the need for dedicated spectral hardware. A simple training mechanism allows the output to be adapted for different physical properties, such as terrain class and friction coefficient. In future work, we plan to extend this framework beyond navigation to tasks such as manipulation, and to infer additional physical properties relevant to robot-environment interaction, including moisture content [41] and terrain compliance [42].

REFERENCES

- [1] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. F. R. Jesus, R. F. Berriel, T. M. Paixão, F. W. Mutz, T. Oliveira-Santos, and A. F. de Souza, “Self-driving cars: A survey,” *CoRR*, vol. abs/1901.04407, 2019. [Online]. Available: <http://arxiv.org/abs/1901.04407>
- [2] F. A. . E. E. Staff, “Fire Departments Adopt Robotic Technologies for Their Operations — fireapparatusmagazine.com,” <https://www.fireapparatusmagazine.com/fire-technology-innovation/fire-departments-adopt-robotic-technologies-for-their-operations/>, [Accessed 17-09-2025].
- [3] N. Hanson, S. Prajapati, J. Tukpah, Y. Mewada, and T. Padir, “Forest biomass mapping with terrestrial hyperspectral imaging for wildfire risk monitoring,” 2024. [Online]. Available: <https://arxiv.org/abs/2411.16107>
- [4] V. Nikitin, S. Golubin, R. Belov, V. Gusev, and N. Andrianov, “Development of a robotic vehicle complex for wildfire-fighting by means of fire-protection roll screens,” in *IOP Conference Series: Earth and Environmental Science*, vol. 226, no. 1. IOP Publishing, 2019, p. 012003.
- [5] L. M. Ferreira, A. P. Coimbra, and A. T. de Almeida, “Autonomous system for wildfire and forest fire early detection and control,” *Inventions*, vol. 5, no. 3, p. 41, 2020.

- [6] M. Brandão, Y. M. Shiguematsu, K. Hashimoto, and A. Takanishi, “Material recognition cnns and hierarchical planning for biped robot locomotion on slippery terrain,” in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 81–88.
- [7] T. Guan, D. Kothandaraman, R. Chandra, A. J. Sathyamoorthy, K. Weerakoon, and D. Manocha, “Ga-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments,” *IEEE Robotics and Automation Letters*, vol. 7, pp. 8138–8145, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:247025506>
- [8] M. Otoofi, W. J. Midgley, L. Laine, H. Leon, L. Justham, and J. Fleming, “Estimating friction coefficient using generative modelling,” in *2023 IEEE International Conference on Mechatronics (ICM)*, 2023, pp. 1–6.
- [9] X. Cai, S. Ancha, L. Sharma, P. R. Osteen, B. Bucher, S. Phillips, J. Wang, M. Everett, N. Roy, and J. P. How, “EVORA: Deep Evidential Traversability Learning for Risk-Aware Off-Road Autonomy,” *IEEE Transactions on Robotics*, vol. 40, pp. 3756–3777, 2024.
- [10] M. Oh, B. Yu, I. M. A. Nahrendra, S. Jang, H. Lee, D. Lee, S. Lee, Y. Kim, M. K. Christiansen, H. Lim, and H. Myung, “Trip: Terrain traversability mapping with risk-aware prediction for enhanced online quadrupedal robot navigation,” 2024. [Online]. Available: <https://arxiv.org/abs/2411.17134>
- [11] Z. Erickson, N. Luskey, S. Chernova, and C. C. Kemp, “Classification of household materials via spectroscopy,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 700–707, 2019.
- [12] N. Hanson, W. Lewis, K. Puthuveetil, D. Furline, A. Padmanabha, T. Padir, and Z. Erickson, “Slurp! spectroscopy of liquids using robot pre-touch sensing,” *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3786–3792, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:252816085>
- [13] N. Hanson, G. Lvov, V. Rautela, S. Hibbard, E. Holand, C. DiMarzio, and T. Padir, “Prospect: Precision robot spectroscopy exploration and characterization tool,” 2024. [Online]. Available: <https://arxiv.org/abs/2403.17232>
- [14] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: <http://arxiv.org/abs/1506.02640>
- [15] Q. Wang, Y.-Y. Chang, R. Cai, Z. Li, B. Hariharan, A. Holynski, and N. Snavely, “Tracking everything everywhere all at once,” in *International Conference on Computer Vision*, 2023.
- [16] S. Vijayarangan and D. Wettergreen, “Assisting spectral mapping using cameras,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 10723–10728.
- [17] A. Trivedi, S. Prajapati, M. Zolotas, M. Everett, and T. Padir, “Chance-constrained convex mpc for robust quadruped locomotion under parametric and additive uncertainties,” *IEEE Robotics and Automation Letters*, 2025.
- [18] R. Ojala and E. Alamikkotervo, “Road surface friction estimation for winter conditions utilising general visual features,” *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1375–1382, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:269362070>
- [19] M. Elnoor, K. Weerakoon, G. Seneviratne, R. Xian, T. Guan, M. K. M. Jaffar, V. Rajagopal, and D. Manocha, “Robot navigation using physically grounded vision-language models in outdoor environments,” 2024. [Online]. Available: <https://arxiv.org/abs/2409.20445>
- [20] Y. Cho, N. Bianchi-Berthouze, N. Marquardt, and S. J. Julier, “Deep thermal imaging: Proximate material type recognition in the wild through deep learning of spatial surface temperature patterns,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–13. [Online]. Available: <https://doi.org/10.1145/3173574.3173576>
- [21] T. Bhattacharjee, H. M. Clever, J. Wade, and C. C. Kemp, “Material recognition via heat transfer given ambiguous initial conditions,” *IEEE Trans. Haptics*, vol. 14, no. 4, p. 885–896, Oct. 2021. [Online]. Available: <https://doi.org/10.1109/TOH.2021.3089990>
- [22] N. Roy, G. Dudek, and P. Freedman, “Surface sensing and classification for efficient mobile robot navigation,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, 1996, pp. 1224–1228 vol.2.
- [23] A. Trivedi, M. Zolotas, A. Abbas, S. Prajapati, S. Bazzi, and T. Padir, “A probabilistic motion model for skid-steer wheeled mobile robot navigation on off-road terrains,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 12599–12605.
- [24] M. Elnoor, K. Weerakoon, A. J. Sathyamoorthy, T. Guan, V. Rajagopal, and D. Manocha, “Amco: Adaptive multimodal coupling of vision and proprioception for quadruped robot navigation in outdoor environments,” 2024. [Online]. Available: <https://arxiv.org/abs/2403.13235>
- [25] A. Trivedi, S. Prajapati, A. Shirgaonkar, M. Zolotas, and T. Padir, “Data-driven sampling based stochastic mpc for skid-steer mobile robot navigation,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 16619–16625.
- [26] M. Akewar and M. Chandak, “Hyperspectral imaging algorithms and applications: A review,” 12 2023.
- [27] U. Singh, N. Trehan, R. Vyas, and O. Vyas, “Terrain classification using transfer learning on hyperspectral images: A comparative study,” in *IEEE*, Jun. 2022.
- [28] N. Hanson, M. Shaham, D. Erdoğmuş, and T. Padir, “Vast: Visual and spectral terrain classification in unstructured multi-class environments,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 3956–3963.
- [29] K. Weerakoon, A. J. Sathyamoorthy, U. Patel, and D. Manocha, “TERP: reliable planning in uneven outdoor environments using deep reinforcement learning,” *CoRR*, vol. abs/2109.05120, 2021. [Online]. Available: <https://arxiv.org/abs/2109.05120>
- [30] G. Huang, Z. Liu, and K. Q. Weinberger, “Densely connected convolutional networks,” *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: <http://arxiv.org/abs/1608.06993>
- [31] M. Brandão, K. Hashimoto, and A. Takanishi, “Friction from vision: A study of algorithmic and human performance with consequences for robot perception and teleoperation,” Nov. 2016.
- [32] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [33] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, “Segment anything,” 2023. [Online]. Available: <https://arxiv.org/abs/2304.02643>
- [34] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, “Mit cheetah 3: Design and control of a robust, dynamic quadruped robot,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2245–2252.
- [35] E. Coumans and Y. Bai, “Pybullet, a python module for physics simulation for games, robotics and machine learning,” <http://pybullet.org>, 2016–2021.
- [36] L. Zhou, Z. Li, and M. Kaess, “Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5562–5569.
- [37] H. Lim, O. Minho, and H. Myung, “Patchwork: Concentric zone-based region-wise ground segmentation with ground likelihood estimation using a 3d lidar sensor,” *IEEE Robotics and Automation Letters*, 2021.
- [38] H. Bao, L. Dong, and F. Wei, “Beit: BERT pre-training of image transformers,” *CoRR*, vol. abs/2106.08254, 2021. [Online]. Available: <https://arxiv.org/abs/2106.08254>
- [39] J. Kim and S.-H. Baek, “Pixel-aligned rgb-nir stereo imaging and dataset for robot vision,” 2024. [Online]. Available: <https://arxiv.org/abs/2411.18025>
- [40] N. Hanson, B. Pyatski, S. Hibbard, C. DiMarzio, and T. Padir, “Hyperdrive: Visible-short wave infrared hyperspectral imaging datasets for robots in unstructured environments,” 2023. [Online]. Available: <https://arxiv.org/abs/2308.08058>
- [41] C. Graham, J. Girkin, and C. Bourgenot, “Spectral index selection method for remote moisture sensing under challenging illumination conditions,” *Scientific Reports*, vol. 12, no. 1, p. 14555, Aug. 2022.
- [42] A. Solomon and A. Sundaresan, “Land-cover classification with hyperspectral remote sensing image using cnn and spectral band selection,” *Remote Sensing Applications: Society and Environment*, vol. 31, p. 100986, 05 2023.