APPLICATION OF K- NEAREST NEIGHBORS ALGORITHM FOR CLASSIFICATION TASK OF SOILS OBTAINED BY TERRAIN SENSORS ON A FOUR-LEGGED ROBOT USING THE ANYMAL DATASET

Karen Valeria Villanueva Novelo, 2009146@upy.edu.mx; Victor Alejandro Moo Quintal, 2009098@upy.edu.mx; Angel Huerta, 2009071@upy.edu.mx; Esteban Rodríguez, 2009116@upy.edu.mx; Joshua Zamora Ramírez, 1909191@upy.edu.mx.

INDEX

I. INTRODUCTION

THE exploration and navigation of diverse terrains, particularly in remote or extraterrestrial environments, present significant challenges for robotic systems. Legged robots, with their dynamic and adaptable locomotion capabilities, appear as potential solutions in these demanding contexts. Their ability to walk through complex landscapes, from fine-grained, granular terrains to uneven rocky surfaces, is crucial, especially in scenarios where wheeled robots may face limitations as noted by Kolvenbach et al. [4]. The ability of these robots to navigate different surfaces safely and efficiently is essential for tasks like planetary exploration and search and rescue operations.

The key to enhancing the functionality of legged robots lies in the effective utilization of sensor data. Sensors attached to robotic legs can provide invaluable information about the terrain, enabling these robots to adapt their movement strategies in real-time. This adaptability is essential for safe and efficient navigation, particularly in unpredictable environments like those found on other planets. The data gathered from these sensors, encompassing various force, torque, and inertial measurements, can be leveraged to classify different types of terrain. These classifications, in turn, inform the robot's decision-making process, aiding in the selection of the most suitable paths and movement patterns.

However, the processing of sensor data presents its own set of challenges, primarily due to its complexity and high dimensionality. Machine learning algorithms, particularly K-Nearest Neighbors (KNN), have emerged as powerful tools for analyzing this data. KNN is recognized for its ability to handle multi-dimensional datasets and adapt to varying data distributions, as discussed by Tokuç [10]. The selection of the appropriate machine learning algorithm is critical, as it needs to balance computational efficiency with accuracy, especially for real-time applications.

In this project, the focus is on exploring various machine learning models to classify terrain types using sensor data from legged robots, with particular attention given to the K-Nearest Neighbors (KNN) algorithm. This algorithm was selected for its proficiency in handling multi-dimensional datasets and its flexibility with diverse data distributions. The reasons for choosing KNN are discussed, including its capability to process sensor data effectively in real-time, a critical requirement for autonomous robotic navigation.

Furthermore, the preprocessing of sensor data is addressed, highlighting the importance of standardization and cleaning to ensure the accuracy and reliability of the machine learning models. Such preprocessing is crucial in managing the inherent variability and complexity of sensor readings, which may include noise and outliers.

This project contributes to the ongoing development of legged robots, particularly in navigating challenging terrains. It aims to demonstrate how the intelligent processing of sensor data can enhance the adaptability and decision-making capabilities of these robots, potentially leading to more effective and versatile robotic systems in various applications, including planetary exploration and terrestrial search and rescue missions. Additionally, this work adds to the broader field of robotics by showcasing the potential of machine learning in improving the interaction between robots and their environments.

I-A. *Theorethical framework*

I-A1. Terrain recognition: Terrain recognition is a critical aspect of robotic navigation and efficiency. The ability to identify and adapt to different terrain types impacts several key areas:

Fuel Efficiency and Energy Conservation: The identification of terrain types facilitates adjustments in speed and movement, optimizing fuel or energy consumption. This aspect is particularly relevant for the ANYmal robot, as efficient locomotion extends its operational autonomy, as noted by Vangen et al. [2].

Reduced Vehicle Wear and Safety: The understanding of terrain conditions aids in circumventing surfaces that could lead to wear or pose safety hazards. In the case of ANYmal, this translates to an extended operational lifespan and the avoidance of potentially damaging scenarios.

Travel Time and Obstacle Avoidance: The selection of routes with more favorable terrain can considerably diminish travel time. For a quadruped robot like ANYmal, this entails not just pinpointing the most straightforward path but also strategizing routes that bypass obstacles, a concept underlined in the findings of Kozlowski and Walas [1].

Stress Reduction and Specific Tasks: Easier navigation leads to reduced strain on the robot's mechanical systems. In specialized scenarios, such as search and rescue operations, the recognition of terrain is crucial for task planning and execution, as highlighted by Vangen et al. [2].

Autonomous Driving and Interaction with the Environment: In the realm of autonomous navigation, the real-time recognition of terrain is indispensable. It enables ANYmal to make informed decisions, adjust its movements, and effectively interact with its environment, including overcoming steps or circumnavigating obstacles in various settings, as demonstrated by research [1][2].

The integration of advanced sensors and AI for real-time terrain analysis, as emphasized by Kozlowski and Walas [1] and Vangen et al. [2], is crucial for the operational success of robots like ANYmal.

I-A2. Quadruped ANYmal robot: ANYmal is a four-legged robot that has been methodically designed for independent operation in challenging and complex conditions. Driven by specifically designed torque-controllable actuators, the robot platform demonstrates the ability to climb and run dynamically across difficult terrain. Using depth cameras and LIDAR improves ANYmal's ability to sense its surroundings, which enables it to navigate with greater accuracy, autonomously determine its location, and select its footing with caution. This combination of cutting-edge capabilities allows ANYmal to function with flexibility and agility in most challenging circumstances [3].

ANYmal is designed to do certain jobs in commercial and industrial contexts, especially in demanding conditions like search and rescue missions or oil and gas platforms. The robot can travel and function in challenging environments by utilizing its ability to perceive its surroundings. Driven by incredibly accurate actuators, ANYmal exhibits the capacity to perform dynamic running gaits [4].

ANYmal can carry an additional cargo of up to 15 kg and has an ideal size that allows it to navigate and overcome obstacles with ease. The robot, a battery charger, and a single laptop are all that are needed for a simple platform setup. ANYmal has an integrated battery that allows it to operate continuously for two to four hours, depending on the type of activities. Is water and dust proof, with an IP67 rating, specifically built for real-world applications. Its impact-resistant, ruggedized casing adds to its dependability and durability in a variety of operating conditions. [3] The compact version of ANYmal weighs less than 30 kg, yet it possesses the capacity to transport a range of equipment, including batteries, optical and thermal cameras, microphones, dynamic lighting systems, and gas detection sensors. ANYmal's adaptable payload capabilities and lightweight construction make it an excellent choice for a range of applications in tough and dynamic conditions [4].

A combination of carbon fiber and aluminum is used in the construction of the main body and leg parts of ANYmal in order to produce a lightweight design. The onboard batteries, which weigh 3 kg and have a capacity of about 650 Wh, provide power for more than two hours of independent use. Leg padding and a protective frame help to minimize damage in the event of a fall, making deployment and transportation easier. The robot can interact with its environment more effectively because to the tactile feet that optoforce sensors provide. Furthermore, revolving Hokuyo UTM-30LX sensors help create a thorough 3D perception of the surroundings. A modular pan-tilt head with different sensory payload capabilities may be easily installed onto the robot platform to improve ANYmal's flexibility in a variety of settings. ANYmal may be tailored for various purposes and sensory needs because to its modular architecture [5].

According to Kolvenbach [6], ANYmal employs point feet that are composed of an elliptical nitrile rubber (NBR) sole that is reinforced by 15 mm of memory foam (Poron XRD) in order to reduce peak loads upon impact. On hard surfaces, the diameter of the foot usually generates an area of $8\,\mathrm{cm}^2$, which increases to $28\,\mathrm{cm}^2$ when the foot is completely submerged in compressible terrain. These feet, which weigh 325 g total weight (including the shank), have a 6-Axis Force/Torque (F/T) sensor made in-house that can sample data at 400 Hz. Although this foot design works well in compacted soils, it has drawbacks in loose, loose soils because of greater sinkage.

II. Methods and tools

II-A. Dataset Used

The dataset for this project was primarily extracted from the work by Kolvenbach et al. [6], which includes diverse sensor readings using two types of robotic legs with different properties and sensors attached. These sensors were tested on various types of materials, created to simulate the terrain of different types of objects. However, for this project, only their properties were considered. The materials include very fine-grained, porous, and highly compressible dust (ES-1), very fine sand (ES-2), medium-coarse sand (ES-3), and solid rock (Bedrock). Variations and mixtures used were coarse sand (CS), bedrock covered in ES-2, and rocks on ES-3.

The focus of this project was on the 'Pointfoot_dataset', containing various preprocessed readings from the Pointfoot type leg sensors, as used in the ANYmal quadruped robot model. The ANYmal's robot leg consists of a nitrile rubber (NBR) sole of elliptical shape, supported by 15 mm of memory foam (Poron XRD) to reduce peak loads during impact, and integrates six 6-Axis Force/Torque sensors, sampling at 400 Hz [6].

A total of 2600 impacts were recorded with the testbed, and an additional 240 impacts with the ANYmal robot. The minimal post-processing of the acquired foot sensor data involved identifying the impact peak and extracting the raw impact oscillation, resulting in a signal length of 200 sample points or 0.5 s per impact [6].

The classification approach that was used in the mentioned study for analyzing this dataset is based on the analysis of oscillations resulting from a controlled impact on the soil, sensed by Force/Torque and IMU sensors in the feet.

II-B. Preprocessing

The dataset, obtained in matrix format (.mat files), was composed of nine files named 'data_all_...', each corresponding to different types of Martian soil simulants [6]. These simulants included ES-1, ES-2, ES-3, Bedrock, Coarse Sand (CS), and variations like Bedrock embedded in ES-2 and Rocks on ES-3. The material ES-4 was excluded due to its minimal observations compared to other materials, creating an unbalanced dataset. Also, ES-4's contribution to the study was not considered critical as it was a variation of ES-3.

The preprocessing involved merging seven of these files: 'data_all_cs', 'data_all_bedrock', 'data_all_es1', 'data_all_es2', 'data_all_es3', 'data_all_pebbles_in_es3', and 'data_all_bedrock_embedded_es2' [6]. Each file's data were individually labeled post-merging. The terrains were then manually classified and ranked according to their optimally for the movement of the quadruped robot. The classification was as follows:

- Bedrock (New Label 7): Solid, rough limestone, most optimal for the robot.
- Bedrock covered in ES-2 (New Label 6): Stable limestone with a layer of fine sand.
- ES-3 (New Label 5): Gravelly medium-coarse sand, moderately complex terrain.
- Coarse Sand (CS) (New Label 4): Variable and challenging terrain.
- ES-2 (New Label 3): Very fine sand, significant challenges in traversal.
- Rocks on ES-3 (New Label 2): Uneven terrain with volcanic rocks.
- ES-1 (New Label 1): Highly compressible dust, least optimal terrain.

The dataset files contained large matrices with each matrix representing sensor signals for a specific soil type. For instance, the Pointfoot_dataset's 'data_all_cs.mat' file included a matrix (35000 x 6) representing 175 impacts with six columns for Force/Torque data (Fx, Fy, Fz, Tx, Ty, Tz) [6]. These matrices, consisting of rows corresponding to stacked impacts (200 rows per impact), encapsulated the detailed sensor data necessary for the study.

The preprocessing included standardizing the sensor data from all soil types. This step was important for several reasons:

1. **Scaling Features**: The sensor measurements, like force and torque, were in different units and had a wide range of values. Standardizing these measurements brought them to a similar scale. This is important for models that use distance calculations, ensuring that no single type of measurement has too much influence.
2. **Normalizing Data**: The data varied a lot in its distribution, with some values being much more common than others. By standardizing (making the data have an average of 0 and a standard deviation of 1), we made the data more regular. This helps make the process of training models more straightforward and effective.
3. **Reducing Impact of Outliers**: Outliers, or very unusual values, can affect the average and spread of the data. Standardizing the data made these outliers less influential, leading to a more balanced and fair training process for the models.

After standardization, it was observed that duplicate data was removed. Duplicates could have resulted from repeated measurements and could have caused bias in the training. Their removal ensured a more accurate and general understanding by the models.

The final dataset was made more consistent and seemed suitable for machine learning techniques. It was representative of the different terrains, which was important for the development of models that predict how the robot will perform on various types of soil.

II-C. Data Analysis Tools

For preprocessing and analyzing the dataset, we used several tools:

- **MATLAB**: For initial data extraction and manipulation from .mat files.
- **Python and Pandas Library**: For cleaning, transforming, and standardizing the data. Pandas is good for working with large datasets.
- **Scikit-learn**: For standardizing data and developing machine learning models. Its tools are helpful for preprocessing and model training.
- **Jupyter Notebooks**: For an interactive development environment, allowing us to explore, visualize, and document the data analysis process.

These tools together made it easier and more effective to handle, preprocess, and analyze the complex dataset, preparing it for machine learning analysis.

III. Development

III-A. Model selection

The dataset comprised six key features: Fx, Fy, Fz, Tx, Ty, and Tz, representing various sensor readings, along with a categorical label for classification. Preliminary analysis of the dataset indicated a diverse range of values across features, suggesting complex relationships and interactions essential for terrain categorization.

Given the task's classification nature, several algorithms were tested in order to select the best one:

Decision Tree: Known for its simplicity and interpretability, the Decision Tree was first applied. It achieved an accuracy of approximately 81.25 %. However, it was considered that there might be the possibility to obtain better results with other algorithms, especially considering the potential for overfitting in complex datasets. [7]

K-Nearest Neighbors (KNN): The KNN algorithm, initially tested with a default parameter of k=5, exhibited a promising accuracy of about 88.84 %. This result highlighted KNN's capability to handle the dataset's multidimensional nature effectively. [8]

Random Forest: The Random Forest method was also considered for its robustness and ability to manage overfitting, it got an accuracy of 89 %. However, in the context of real-time processing requirements, its computational intensity was a concern. [9]

In the same way, it was taken into account the model's computational efficiency, especially for real-time applications, as the robot will be doing.

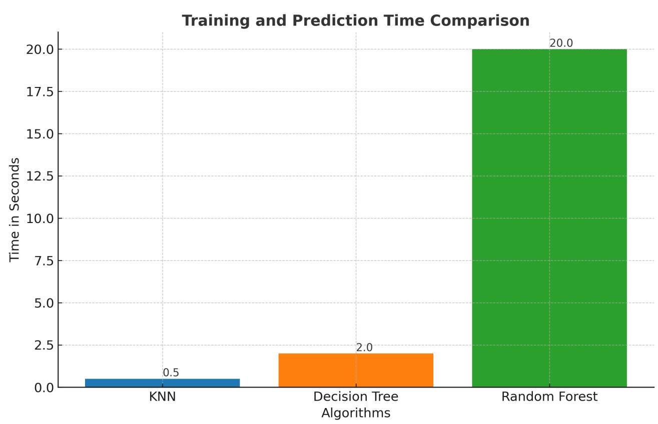

Figura 1. Comparison of algorithms time efficiency

As can be observed, the most efficient was the KNN algorithm, the difference that It had with the Decision tree algorithm is minimal, however, the accuracy, as mentioned before, was higher. Therefore, it was determined that the KNN algorithm was the most accurate for the model.

III-B. KNN Hyperparameter Tuning

One of the most important things to consider when optimizing the KNN algorithm is the fine-tuning of the 'k' value. This parameter determines how many nearest neighbors influence the classification of each data point.

The model was tested with k=3,5,7 and 9. Each value offered insights into the model's sensitivity and generalization capabilities.

As it can be observed, the accuracy obtained from k=5 is similar to k=3; however, it was selected the KNN algorithm because lower values of k can make the model more sensitive to noise in the data, potentially leading to overfitting. On the other hand, slightly higher values like 5 might provide better generalization by considering more neighbors.

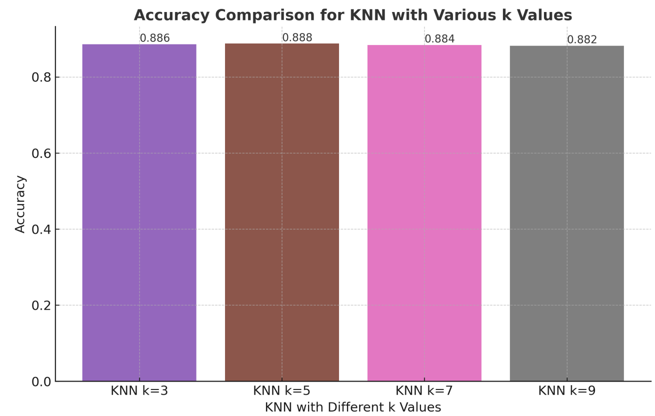

Figura 2. Comparison of algorithms application

III-C. Justification for KNN selection

One of the primary advantages of the KNN algorithm was its adaptability to high-dimensional datasets. The dataset in question comprised sensor readings across multiple axes, resulting in a multi-dimensional feature space. KNN's inherent ability to handle such multi-dimensional data made it a really good choice for this specific application. [10]

KNN is a non-parametric algorithm, which implies that it does not assume any specific data distribution. This characteristic was considered an advantage for the work, given the diverse and complex nature of terrain data. It enabled the algorithm to flexibly adapt to varying terrain conditions. KNN's hyperparameter 'k,' representing the number of nearest neighbors considered for classification, offered the possibility for optimization of the code. Through systematic tuning, the optimal 'k' value (k = 5) was chosen, determining a balance between model accuracy and generalization. [10]

Real-time processing requirements were a fundamental consideration for the terrain classification task, given the robotic application. KNN's computational efficiency, as evidenced by its training and prediction times, made it compatible with real-time development, ensuring timely terrain recognition and response. [11]

The KNN algorithm makes decisions based on the local neighborhood of data points. This feature aligns with the task of terrain classification, where the characteristics of adjacent sensor readings often provide important insights into the terrain type. The way the algorithm uses closeness to make decisions was good in this case, due to the small changes that can be presented in the different terrains the sensors will be reading. [12]

The selection of the KNN algorithm for terrain classification was driven by its inherent characteristics, compatibility with the dataset's multi-dimensional characteristics, adaptability to non-parametric data, and its ability to make local decisions based on proximity. The fine-tuning of the 'k' hyperparameter, along with its computational efficiency, made KNN the optimal choice for this work. Consequently, KNN exhibited compatibility and suitability for achieving accurate and real-time terrain classification in the context of the four-legged robot application which obtains data with different sensors.

IV. Results and discussion

The K-Nearest Neighbors (KNN) algorithm was selected as the primary model for terrain classification in this project, based on its compatibility with the high-dimensional nature of the sensor data from the ANYmal robot. The model demonstrated an impressive accuracy of approximately 88.84 %, which was a significant improvement over other tested algorithms like Decision Trees and Random Forests. This high level of accuracy was achieved through the careful selection of the 'k' value, which was finalized at k=5. This choice struck a balance between sensitivity to noise and generalization capabilities, making the KNN algorithm adept at handling the dataset multidimensional characteristics.

The KNN model's success was primarily due to its ability to make decisions based on the local neighborhood of data points, an approach well-suited for the terrain classification task. The algorithm's adaptability to non-parametric data and its computational efficiency, crucial for real-time processing requirements, being suitable for the project.

The potential of further applications of the dataset is wide. The plan is to utilize the dataset to create different clusters of the soils to detect more specific types of terrains. For example, the category of 'bedrock' could be subdivided into three groups: optimal, medium, and suboptimal bedrock, with corresponding scores of 50, 45, and 40 points, respectively.

This refined classification system would allow for a more deep understanding of terrain types, enabling the ANYmal robot to make more informed decisions about where to step. Implementing reinforcement learning in the future could further enhance the robot's ability to choose the most optimal trajectory on a given path, taking into account factors like energy conservation, time efficiency, and resource management.

By advancing the model to recognize these subtle differences in terrain types, the robot could avoid less favorable surfaces, extending its operational life, ensuring safety, and improving overall efficiency.

V. Conclusions

It was demonstrated that the classification of diverse terrains, essential for efficient robotic navigation, is achievable using advanced machine learning techniques. The study focused on the ANYmal quadruped robot, exploring the use of its sensor data for terrain recognition. The selected K-Nearest Neighbors (KNN) algorithm successfully categorized various terrain types with high accuracy.

The classification process was based on the analysis of sensor data, including Force/Torque and Inertial Measurement Unit (IMU) readings obtained from the robot's feet. The dataset was preprocessed and standardized, to be prepared for analyze. The KNN model, with a hyperparameter of k=5, showed remarkable efficiency in terrain classification, reflecting the algorithm's strength in handling multi-dimensional datasets and its suitability for real-time processing requirements.

The robustness of the KNN model was validated using a comprehensive dataset that collected a wide range of terrain types, including variations in texture and firmness. This approach allowed for an in-depth understanding of the robot's interaction with different terrains, contributing significantly to the development of more advanced navigation strategies.

Achieving an accuracy of approximately 88.84 %, the KNN model proved its effectiveness in terrain classification. This high level of accuracy indicates that only a minimal number of sensors are necessary to achieve reliable terrain classification, a finding that could simplify the sensor configuration on future robotic designs.

The implementation of the KNN algorithm on the ANYmal robot opens up possibilities for more sophisticated terrain recognition and decision-making processes in real-time. Looking ahead, further tuning of the model, along with the incorporation of additional terrain types, could enhance the resolution of the classification. The consideration of a confidence measure in the predictions could also be explored to refine the decision-making process, especially in cases of low confidence.

References

[1] J. M. Idelsohn, .A learning system for terrain recognition,"Pattern Recognition, vol. 2, no. 4, pp. 293-301, 1970. doi: 10.1016/0031-3203(70)90019-1.

[2] P. Kozlowski and K. Walas, "Deep neural networks for terrain recognition task,ïn 2018 Baltic URSI Symposium (URSI), Poznan, Poland, 2018, pp. 283-286. doi: 10.23919/URSI.2018.8406736.

[3] B. Priyaranjan, "Development of quadruped walking robots: A review," Department of Mechanical Engineering, National Institute of Technology, Arunachal Pradesh 791112, India, 2021, pp. 1-8, doi: 10.1016/j.asej.2020.11.005.

[4] "ANYmal research". ANYmal Research. [Online]. Available: https://www.anymal-research.org/

[5] M. Hutter et al., .ANYmal - a highly mobile and dynamic quadrupedal robot,"2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea (South), 2016, pp. 38-44, doi: 10.1109/IROS.2016.7758092.

[6] H. Kolvenbach, C. Bartschi, L. Wellhausen, R. Grandia, and M. Hutter, "Haptic Inspection of Planetary Soils With Legged Robots,ïn IEEE Robotics and Automation Letters, vol. 4, no. 2, April 2019, doi: 10.1109/LRA.2019.2896732.

[7] A. R. Kapil, "Decision Tree Algorithm in Machine Learning: Advantages, Disadvantages, and Limitations,.Analytixlabs, October 1, 2022. [Online]. Available: https://www.analytixlabs.co.in/blog/decision-tree-algorithm/

[8] Genesis, "Pros and Cons of K-Nearest Neighbors,"FromTheGenesis, September 25, 2018. [Online]. Available: https://www.fromthegenesis.com/pros-and-cons-of-k-nearest-neighbors/

[9] A. Fleiss, "What are the advantages and disadvantages of random forest?," Rebellion Research, Feb. 19, 2023. https://www.rebellionresearch.com/what-are-the-advantages-and-disadvantages-of-random-forest

[10] A. A. Tokuç, "k-Nearest Neighbors and High Dimensional Data,"Baeldung, November 9, 2022. [Online]. Available: https://www.baeldung.com/cs/k-nearest-neighbors

[11] "Deep Insights Into K-nearest Neighbors,"Towards AI, July 25, 2020. [Online]. Available: https://towardsai.net/p/machine-learning/deep-insights-into-k-nearest-neighbors

[12] S. Uddin, I. Haque, H. Lu, et al., Çomparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction,"Scientific Reports, vol. 12, p. 6256, 2022. doi: 10.1038/s41598-022-10358-x.