Classification of Terrain Traversability for Autonomous Crawling Vehicles

Harsh Rangwala
Department of Computer Science
George Mason University
hrangwa@gmu.edu

Aniket Datar
Department of Computer Science
George Mason University
adatar@gmu.edu

Divyeshree Potdar
Department of Computer Science
George Mason University
dpotdar2@gmu.edu

***Abstract*—In the context of Autonomous Driving, most vehicles are able to easily traverse a flat well-structured road. While there have been great recent developments in autonomous driving, most of the advancement is focused on on-road driving. Autonomous navigation on an unknown natural terrain can have many critical applications in defence, agriculture, and exploration. Our project formulates the crawling vehicles' traversability in a complex and natural environment as a classification problem. Our goal is to develop a model that can best asses the navigational capacity of a crawler based on the surrounding environment.**

***Index Terms*—crawler, classification, linear models, convolution neural network**

I. Introduction

The advancement of autonomous crawling vehicles has revolutionized their ability to navigate unpredictable and harsh environments. Vehicles like NASA JPL's "Spirit" and "Perseverance" have demonstrated their capability to safely traverse unknown terrains, showcasing the potential of autonomous robots in challenging scenarios [2]. However, despite their mechanical capabilities, wheeled robots are primarily limited to planar workspaces due to existing planning and control techniques, as well as a lack of understanding of terrain traversability and its assessment.

Ensuring safe and successful navigation of a vehicle heavily relies on understanding the traversability of the terrain it encounters. To address this fundamental challenge, this project aims to develop a system capable of predicting whether a given terrain is traversable or not, leveraging RGB-D (color and depth) image data. By harnessing this multimodal data, the proposed system can analyze the characteristics of the terrain and provide an accurate prediction, enabling autonomous crawling vehicles to make informed decisions during navigation.

The motivation behind this research stems from the growing utilization of crawling vehicles in various domains, where these machines encounter complex and unpredictable terrains. By equipping them with the ability to assess the traversability of the terrain ahead, we can enhance their autonomy, efficiency, and overall safety. Current planning and control techniques have largely confined autonomous wheeled robots to planar workspaces, underscoring the need to bridge the gap in understanding and predicting terrain traversability.

Identify applicable funding agency here. If none, delete this.

By leveraging RGB-D image data, which combines visual information with depth perception, we can capture the intricate details of the terrain. This holistic data representation enables the proposed system to analyze the terrain's visual appearance, texture, and structural elements, while also considering the potential obstacles and hazards that may be present. Through this comprehensive analysis, the system can provide accurate predictions regarding the traversability of the terrain, enabling the crawling vehicle to navigate with confidence and adaptability in a wide range of environments.

II. Related Work

Previous research on autonomous crawling vehicles has primarily focused on obstacle detection using passive sensing systems like stereo cameras [3] [4]. Jet Propulsion Laboratory (JPL) has achieved advancements in this area during exploration missions using their robotic vehicles, such as Spirit and Perseverance. However, incorporating additional sensing modalities like radar and lidar can be costly [5].
Cameras, particularly RGB-D cameras like the Microsoft Azure Kinect, offer advantages in terms of cost-effectiveness, high-resolution data, and resilience to environmental interference. [1] This project aims to develop a system for terrain traversability assessment using RGB-D images. Machine learning techniques, including Support Vector Machine (SVM) and neural networks, will be employed to predict terrain traversability based on sensor data from the robot, including IMU, camera, RPM sensor, and optical flow sensor. While prior work has focused on obstacle detection [6], this project's focus is on terrain classification. The proposed system will enable autonomous crawling vehicles to assess the navigational ability on complex and natural terrain. The vehicle used is based on the Traxxas TRX-4 chassis, with a Microsoft Azure Kinect camera providing RGB and Depth images. RPM sensors are employed to detect slippage and speed.

III. Datasets and Features

The project involved the collection of data using a physical robot, with the data pipeline consisting of feed from RGBD camera and IMU. This data was then processed to build an elevation map, represented as a discrete grid map of 2m x 2m with each cell of resolution 6mm, in the robot frame.

The generated map was then exported to a grayscale image where the color is a function of the height of the surface at each grid cell. We then cropped the image to the region of interest, which is the area underneath the expanded robot footprint with a dimension of 400 x 200. Each image was associated with the corresponding roll and pitch value, which were real-time values recorded during the data collection from the IMU.

The collected dataset consisted of approximately 4000 images, each with its respective roll and pitch angles. This dataset was split in an 80:20 ratio as training and test data, respectively. The data was then used to train and test various models to predict the roll and pitch angle of the robot. As these angles directly represent the robot's pose, they serve as indicators of the probability of the robot passing through a particular region.

A sample image generated from the data collection process is shown below.

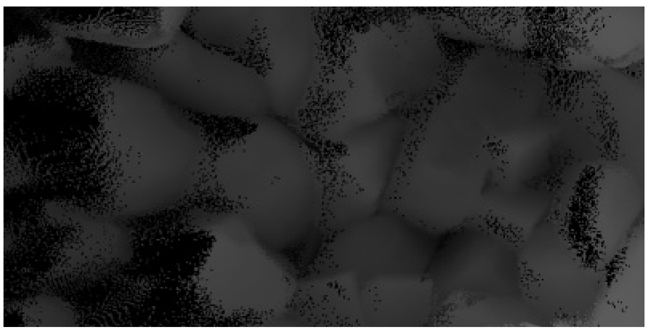

Fig. 1. Elevation Map Size 400 x 200

IV. Methods

Two different methods were attempted to solve the problem. The first method involved regression, where the goal was to predict the continuous values of the roll and pitch angles based on the input image. In contrast, the second method used classification, in which the recorded angle values ranging from -25 to 25 were discretized into 10 bins, and the resulting bin number was used as a class label for training and prediction purposes.

A. Linear Model

To establish a baseline for our models, a simple Linear model was used. The input image was represented as a vector of dimension (n, h, w)
Where,

$$n \text{ (No. of Samples)} = 4000$$
$$h \text{ (Image Height)} = 400$$
$$w \text{ (Image Width)} = 200$$

The array is further flattened make it compatible with the linear model. The output array is of

$$\text{size } (n, h \times w) = (4000, 80000)$$

For the implementation of the linear model we have used sklearn library.

B. Support Vector Machine

For our SVM implementation, we began by preprocessing the flattened image data using the Histogram of Oriented Gradients (HOG) feature extraction technique. This method involves dividing the image into small cells and computing the gradient direction and magnitude of each pixel within each cell. Then, a histogram of the gradient orientations within each cell is constructed and normalized to form a feature vector that captures the texture and shape information of the image.

In our implementation, we used the HOG feature extractor with the following parameters:

$$\text{orientation} = 8$$
$$\text{pixels\_per\_cell} = (4, 4)$$
$$\text{cells\_per\_block} = (4, 4)$$
$$\text{kernel} = 'linear'$$

We used the scikit-learn library to train and test our SVM model. We selected the C-Support Vector Classification (SVC) implementation. The C parameter, which controls the trade-off between maximizing the margin and minimizing the classification error, was set to 1.0.

C. Convolutional Neural Network

Convolutional Neural Networks (CNNs) are widely used in computer vision applications, such as image classification, object detection and recognition, due to their effectiveness in capturing and preserving the spatial relationships between pixels. In this study, we utilized the Keras Sequential API to design our CNN model architecture, which allowed for easy layer stacking.

The input vector for the CNN was of shape $(400, 200, 4)$, where 400 and 200 represent the image height and width, respectively, and we packed the 4 images from the previous time step as a channel. Our CNN model included one convolutional layer with 32 filters of size 3x3, followed by Maxpooling with a pooling window of (2,2) and activation function of ReLU. We then added two more convolutional layers with filter sizes of 64 and 128, respectively, followed by Maxpooling. The output was flattened and two dense layers of 64 and 32 with ReLU activation were added. For the final layer, we added a Dense1 layer for logistic regression and a Dense10 layer with Softmax activation.

During training, we used the 'adam' optimizer, which effectively captured the complex features of our input images and led to a robust and effective CNN model for image classification tasks. By stacking several convolutional and dense layers with appropriate filters and activations, our model was able to provide reasonable accuracy for prediction. Overall, our CNN model architecture was designed to efficiently capture and analyze the features of input images, allowing for effective image classification.

V. Results

Despite multiple attempts at finding suitable parameter combinations, the Linear model failed to provide accurate predictions. The resulting predictions exhibited significant scatter, with no discernible pattern or trend in the data. This suggests that the Linear model alone is unable to capture the underlying structure of the data and that it may not be an appropriate choice for this particular problem.

In an effort to better understand the nature of the model's predictions, a scatter plot (Figure 2) was created to visually represent the relationship between the predicted values and the actual values. The scatter plot provides a better visualization of the distribution of the data and can be useful for identifying any outliers or trends in the data.

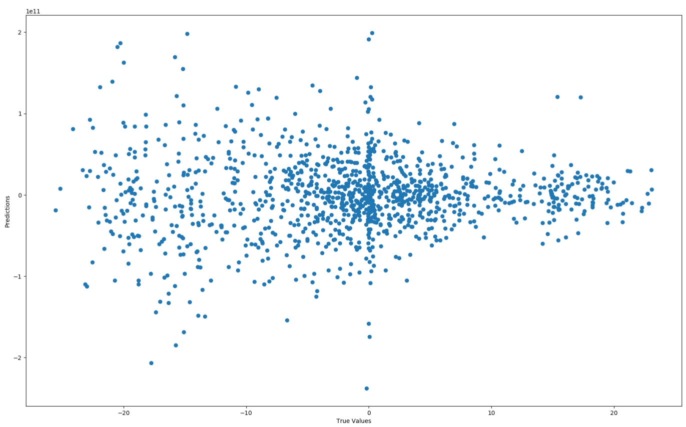

Fig. 2. Linear model true angle vs predicted angle

After attempting to use Principal Component Analysis (PCA) for feature size reduction in conjunction with a Linear model, the convergence of the model was significantly altered. However, the average absolute difference between predicted and true classes remained high at approximately 7.1, indicating that the model's performance is not practical for making conclusive decisions.

While the use of PCA can be an effective technique to reduce the number of features and simplify the dataset, it appears that it was not sufficient to achieve the desired level of accuracy. A scatter plot (referenced as Figure 3) illustrates the relationship between the true class and the predicted class.

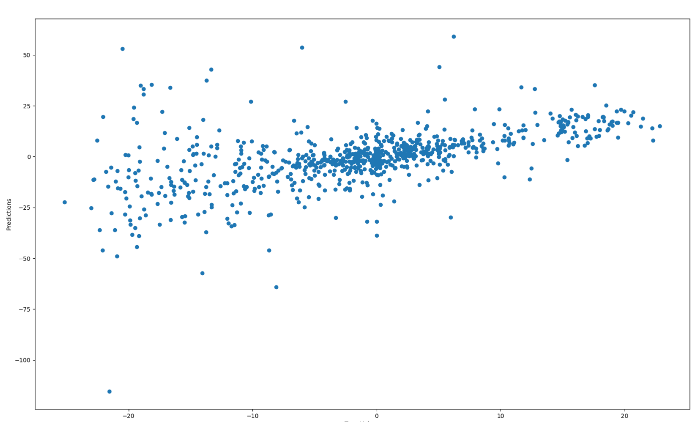

Fig. 3. Linear model with PCA true angle vs predicted angle

The Support Vector Machine (SVM) appears to be a successful for the classification of images, outperforming the previously attempted Linear model. The achieved accuracy of approximately 0.68 indicates a reasonable level of success in correctly predicting the class of a given sample.

To better visualize the performance of the SVM, a scatter plot (referenced as Figure 4) was created to display the relationship between the true class and the predicted class. However, it is important to note that while the SVM achieved a better level of accuracy than the previous models, it is still good enough to be considered as a reliable choice.

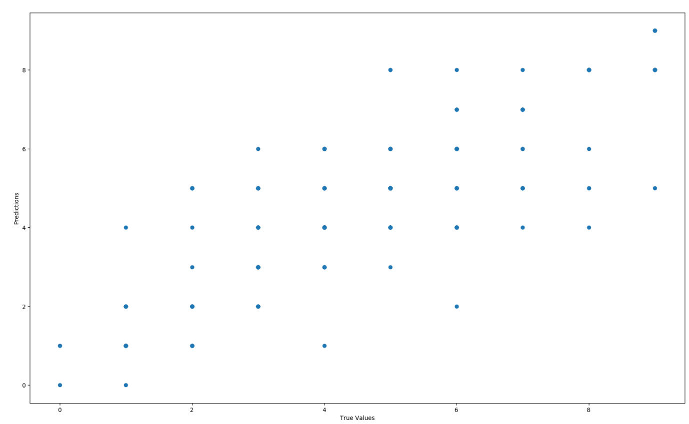

Fig. 4. SVM: True angle class vs predicted angle class

The Convolutional Neural Network (CNN) was trained using a learning rate of 0.0005 and a batch size of 48. To monitor the performance of the model, the training was run for a total of 150 epochs. However, it was observed that the model converged to its minimum error of 1.4 within the first 50 epochs, and further training did not yield significant improvements.

The CNN model performed exceptionally well, with an average absolute difference of approximately 1.36 between the predicted and true angles. To better visualize the accuracy of the model, a scatter plot (referenced as Figure 5) was created to display the relationship between the predicted angle and the true angle.

It was noted during the implementation of the CNN that the model's performance improved with the increase in the size of the training dataset. Initially, the model was trained on only 300 available images, which resulted in overfitting, as the CNN was memorizing the data instead of learning from it. The training error for the available data dropped below 0.0002, but the testing error remained high, ranging between 0.2-0.3.

To address the issue of overfitting, additional images were added to the training dataset, which led to a significant improvement in the CNN's performance. This highlights the importance of having a sufficient amount of diverse training data for this problem.

One thing we observed during the CNN implementation is CNN was getting better as the data size grow. During the initial training, we were using only 300 available images, which was

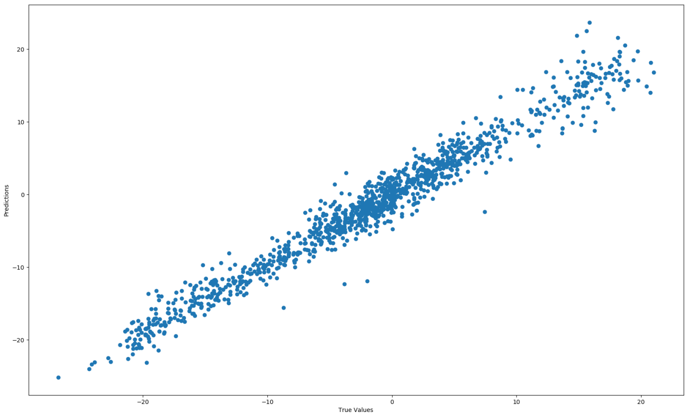

Fig. 5. CNN: True angle class vs predicted angle class

causing over-fitting as the CNN was remembering the data instead of training as the error for the training data dropped below 0.0002 but the testing error was only 0.2-0.3.

VI. Conclusion and Future work

The experiments performed have led us to conclude that the CNN model is a perfect fit for our problem of estimating terrain traversability based on image analysis. The accuracy of the predicted angles, coupled with the potential use of additional features such as RMP values and Optical Flow sensor output, have the potential to further improve the prediction accuracy.

Furthermore, the current dataset used for training and testing only contains a limited set of terrain types. Therefore, it would be interesting to investigate the performance of the proposed methods on different types of terrains, such as sandy, rocky, or muddy, to determine the generalizability of the approach.

In addition to improving the prediction accuracy, the proposed methods could also be integrated into the overall control system of an autonomous crawling vehicle. By doing so, the vehicle would be able to make more informed decisions about its path planning and obstacle avoidance, thereby enhancing the vehicle's overall performance and safety.

Future work could involve the exploration of alternative machine learning models or feature engineering techniques to further enhance the performance of the system. Additionally, the feasibility of incorporating real-time feedback from the vehicle's sensors into the proposed methods could be explored to improve the overall robustness and adaptability of the system in dynamic environments.

References

[1] Shaban, A., Meng, X., Lee, J., Boots, B., Fox, D. (2022). Semantic Terrain Classification for Off-Road Autonomous Driving. In Proceedings of the 5th Conference on Robot Learning (PMLR 164, pp. 619-629).

[2] Mathur, P., Pandian, K. S. (2012). Terrain Classification for Traversability Analysis for Autonomous Robot Navigation in Unknown Natural Terrain. International Journal of Engineering Science and Technology (IJEST).

[3] Rankin, A., Huertas, A., Matthies, L. (2005). Evaluation of stereo vision obstacle detection algorithms for offroad autonomous navigation. In AUVSI Symposium on Unmanned Systems.

[4] Rankin, A., Huertas, A., Matthies, L. (2009). Stereo vision based terrain mapping for off-road autonomous navigation. In Proceedings of SPIE (Vol. 7332, Orlando, FL).

[5] Reina, G., Milella, A., Rouveure, R. (2015). Traversability Analysis for Off-Road Vehicles using Stereo and Radar Data. IEEE International Conference on Industrial Technology, Seville, Spain. DOI: 10.1109/ICIT.2015.7125155.

[6] Li, N., Yu, X., Liu, X., Lu, C., Liang, Z., Su, B. (2020). 3D-Lidar Based Negative Obstacle Detection in Unstructured Environment. In ICVISP 2020: Proceedings of the 2020 4th International Conference on Vision, Image and Signal Processing (pp. 1-6). DOI: 10.1145/3448823.3448865.