Terravision

Members:

Hrishikesh Hiremath,
Darryl David,
Aditi Wekhande,
Sai Sanjana Reddy,
Misha Jain,
Prachi Patil

Introduction

Effective terrain recognition is a critical component of autonomous systems and robotics, facilitating safe and efficient navigation. However, conventional methods for terrain classification often fall short of providing the depth of information required for robust decision-making in dynamic environments. To address these limitations, this proposal introduces a novel deep learning algorithm that leverages Convolutional Neural Networks (CNNs) to significantly enhance terrain recognition.

Our proposed algorithm goes beyond traditional approaches by not only identifying surface types such as sandy, rocky, grassy, and marshy but also extracting implicit quantities of information, including roughness and slipperiness. These additional attributes are fundamental for high-level environment perception and are crucial in ensuring vehicle stability and traction control.

This proposal outlines the technical aspects of our algorithm, including the methodology, data requirements, and potential applications across autonomous vehicles, robotics, and outdoor exploration. Furthermore, it discusses the challenges inherent to terrain classification and how our approach seeks to overcome them. By combining deep learning with implicit quantities of information, our algorithm promises to revolutionize terrain recognition, advancing the state-of-the-art in autonomous navigation and environmental perception.

Motivation

The need for terrain classification :

Terrain classification, an intricate process involving the categorization of land surfaces by a gamut of attributes, occupies a pivotal position within numerous sectors. It serves as a linchpin in the orchestration of astute decision-making processes, the judicious allocation of resources, and the conscientious implementation of safety protocols.

Terrain classification is a fundamental process that holds immense significance across multiple domains, impacting decision-making, resource allocation, and safety measures. This comprehensive examination delineates the multifaceted needs driving terrain classification, demonstrating its indispensable role in diverse sectors, including civil engineering, agriculture, transportation, environmental management, defence, and disaster preparedness.

1. Safety and Risk Mitigation:

Natural Disasters: Terrain classification aids in the identification of high-risk areas susceptible to natural disasters such as landslides, floods, and wildfires. This critical information informs disaster preparedness and response efforts, safeguarding human lives and infrastructure.

2. Infrastructure Development:

Terrain data is imperative for civil engineering projects, including the construction of roads, bridges, and buildings. It ensures structural integrity, cost-effectiveness, and compliance with safety standards.

3. Precision Agriculture and Land Use:

Agriculture: Precision agriculture relies on terrain classification for optimizing irrigation, planting, and harvesting processes. It maximizes crop yields while minimizing resource usage.

Land Use Planning: Terrain data informs land use decisions, guiding zoning, conservation, and sustainable development practices.

4. Transportation and Navigation:

Route Planning: Terrain classification assists in route planning across various transportation modes. It ensures safe and efficient navigation by identifying optimal paths and avoiding treacherous terrain.

5. Environmental Management:

Environmental Monitoring: Terrain classification is vital for monitoring changes in land cover, vegetation, and ecosystems. It supports biodiversity studies, habitat preservation, and climate change assessments.

6. Defence and Military Operations:

Mission Planning: In the defence sector, terrain classification is instrumental in mission planning. It informs troop movements, weapon deployment, and communication strategies, contributing to operational success and strategic advantage.

7. Resource Allocation:

Forestry: Terrain classification identifies suitable logging areas while preserving sensitive ecosystems.

Mining: It aids in the allocation of mining resources, optimizing extraction processes, and minimizing environmental impact.

Energy Exploration: In energy sectors, terrain data informs the location of resource reserves and aids in efficient extraction.

8. Disaster Management:

Evacuation Planning: During natural disasters, terrain classification assists in predicting impact and planning safe evacuation routes.

Relief Operations: It plays a crucial role in disaster relief efforts, enabling effective resource allocation and damage assessment.

9. Environmental Impact Assessment:
Development Projects: Terrain classification is integral to assessing the environmental impact of major projects. It supports regulatory compliance and ensures responsible development.

10. Scientific Research:
Geological Studies: Researchers employ terrain classification to investigate geological, geomorphological, and hydrological phenomena, advancing our understanding of natural processes.

Challenges

Problems that arise during terrain classification :

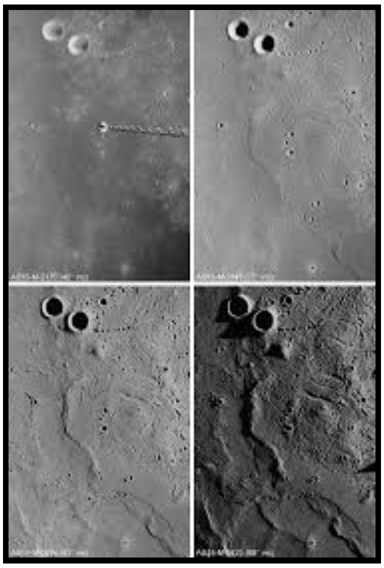
Illumination Changes:

1. **Source Variability:** Illumination changes are primarily caused by variations in the position and intensity of light sources, such as the sun or artificial lighting. As a rover or robot moves across a terrain, the position of the sun changes, leading to different lighting conditions throughout the mission.

2. **Impact on Visual Data:** These changes in illumination have a profound effect on the appearance of the terrain as captured by visual cameras. Shadows, highlights, and overall brightness levels can vary significantly, making it challenging to extract consistent features for terrain classification.

3. **Challenges for Perception:** For a perception system relying solely on visual data, changes in illumination can result in misclassifications or difficulties in

distinguishing between terrain types. For instance, a rocky surface may appear less rocky in certain lighting conditions, leading to potential navigation errors.



4. **Dynamic Adaptation:** To address illumination changes, autonomous systems must be capable of dynamically adapting their perception models. This might involve adjusting camera settings, utilizing different exposure levels, or compensating for shadows to maintain accurate terrain classification.

Complexity of RGB Data:

1. **Multi-Factor Influence:** RGB data captured by cameras is influenced by several factors simultaneously. These include the incident light (intensity and angle), reflections from the environment, and the material properties of the terrain. Each of these factors contributes to the observed RGB values, making the data complex and multifaceted.

2. **Dynamic Nature:** The material properties of the terrain can change over time. For example, a surface may become wet, dry, dusty, or reflective, altering its appearance in RGB images. Traditional analytical models need help to account for such dynamic changes.

3. **Machine Learning Challenges:** When using machine learning approaches for terrain classification, the high dimensionality and complex interactions between these factors pose challenges. Traditional feature engineering may not adequately capture the relevant information, and the models may require large amounts of diverse training data to generalize effectively.

4. **Transferability:** Models trained on RGB data in one environment may not easily transfer to another due to differences in lighting, materials, and

reflections. This lack of transferability necessitates adaptable models or domain adaptation techniques to handle changing terrains.

5. **Robustness Needed:** Achieving robust terrain classification with RGB data often involves advanced techniques such as domain adaptation, feature normalization, and data augmentation to account for the variability in material properties and lighting conditions.

To improve terrain classification in such challenging conditions, several strategies which we will consider are:

1. **Multimodal Sensor Fusion:** we will combine data from multiple sensors, such as visual cameras, infrared cameras, and LIDAR, to create a more robust representation of the terrain. Each sensor can provide complementary information that helps mitigate the effects of changing illumination.

2. **Machine Learning:** we will employ machine learning techniques that are robust to variations in input data. Convolutional Neural Networks (CNNs) and deep learning models can adapt to changes in illumination and complex data patterns.

3. **Data Augmentation:** Enhance the training dataset by artificially generating variations in lighting conditions and terrain types. This can help our machine-learning models become more resilient to changes in illumination.

Approach

Data pre-processing and collection:
We will gather a dataset of RGB and thermal images with the corresponding terrain labels. Each RGB image will have a corresponding thermal image, and both will represent the same geographical location or scene. Our dataset will cover a diverse range of terrains that we want to classify.

We will annotate our dataset with terrain labels corresponding to each synchronized pair of RGB and thermal images. Each pixel in the RGB and thermal images will be labelled with the corresponding terrain class.

Pipeline:

We will create a pipeline that concatenates the RGB and the thermal images before feeding them into the DeepLab network. This will be done using a deep learning framework like TensorFlow.

We will first load both the images and then will make sure that both have the same dimensions. Then both the images are normalized to have values between 0 and 1. We will do this by dividing the pixel value for the higher bit-depth images.

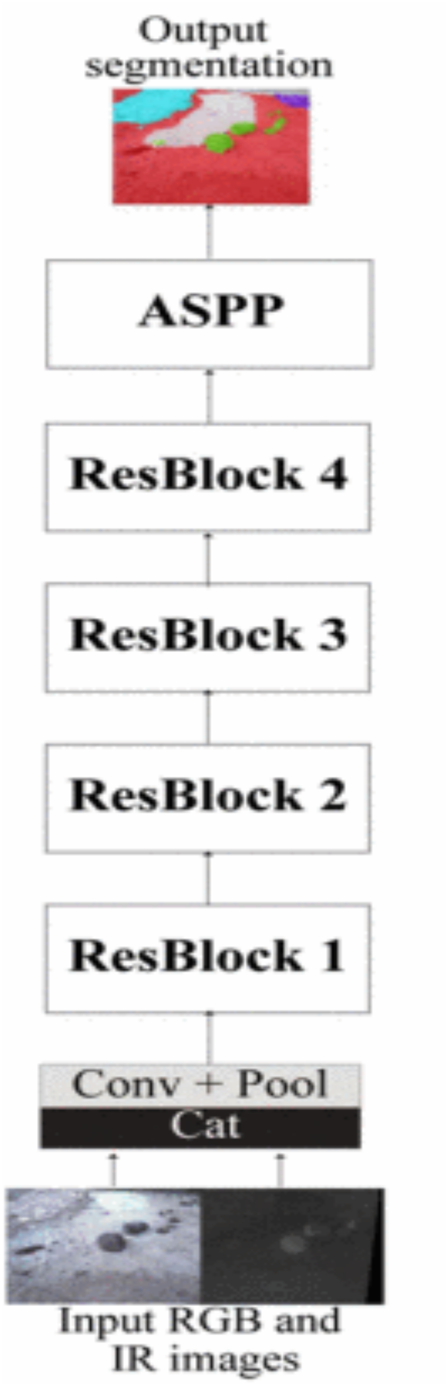
After normalization and format conversion, concatenate the two images along the channel axis to create a multi-channel input. The resulting concatenated input will be a single image with both RGB and thermal information stacked along the channel axis.

Model Architecture:

We propose a novel deep learning-based terrain classification here based on FCN(DeepLab). Fully Convolutional Networks (FCN) is an effective and faster algorithm.
We are going to use a DeepLab-based architecture for semantic segmentation. We are going to implement this either by building our own model from scratch or by using pre-trained models like DeepLabv3+ for TensorFlow.

By modifying the input layer of the DeepLab model to accept both RGB and thermal images as input. To create multi-channel input we are going to concatenate these images along the channel axis.

And then by fine-tuning our model according to our dataset to adapt it to our specific terrain classification task.



(a) TDeepLab BL

Data Splitting:
After the model architecture, we are going to split the dataset into training, validation, and test sets, ensuring that the data distribution across different terrain classes is balanced in each set.

Training:
We are going to train our model on the training dataset by using an appropriate loss function for semantic segmentation, such as cross-entropy loss.

By experimenting with different hyperparameters, learning rates, and regularization techniques to optimize the model's performance.

Monitor training with metrics like accuracy, intersection over union(IoU), and loss to ensure convergence.

Validation:
By evaluating the model's performance on the validation set regularly during training to detect overfitting and make necessary adjustments.

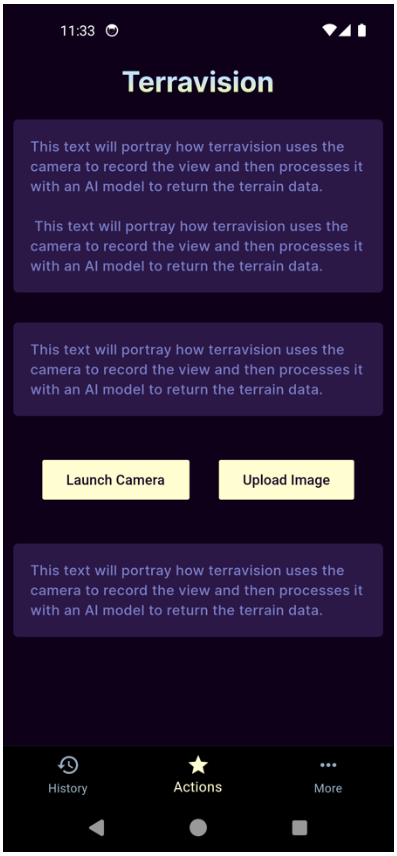
Testing:
Once training is complete, we are going to access the model's performance on the test data to obtain unbiased performance metrics.

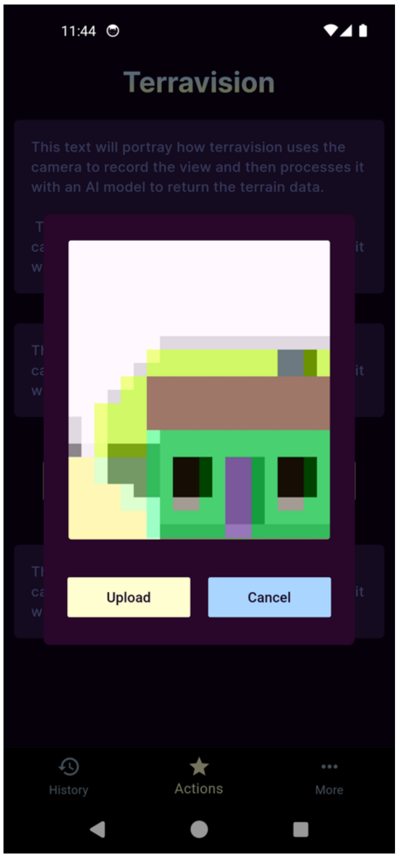
Post-Processing:
We will apply post-processing techniques to refine the segmentation results if needed. This can include techniques like morphological operations or filtering to remove noise or small artifacts.

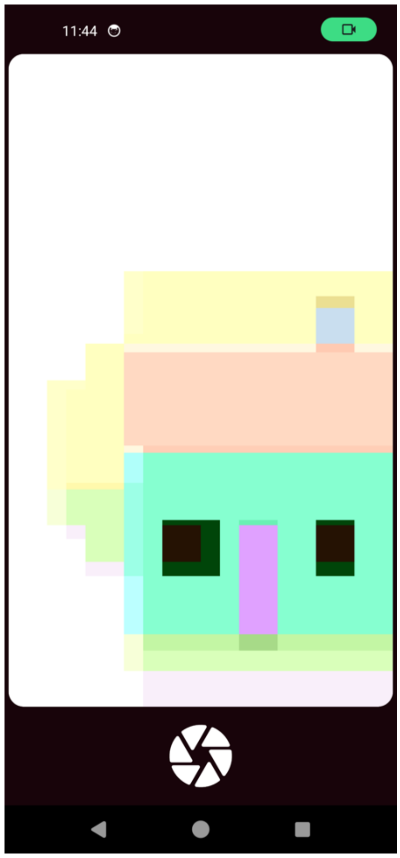
App:
An app will be used to upload and receive the data from the machine learning model.

Images can be captured via the camera and uploaded from the storage. It would also later feature a live camera detection feature.

It has been made using Flutter, hence can run on most of the popular platforms, such as Android, iOS, macOS, chrome, Linux, Windows.







Thank You,

Contact Details:

Hrishikesh Hiremath
9449796245
me220003037@iiti.ac.in