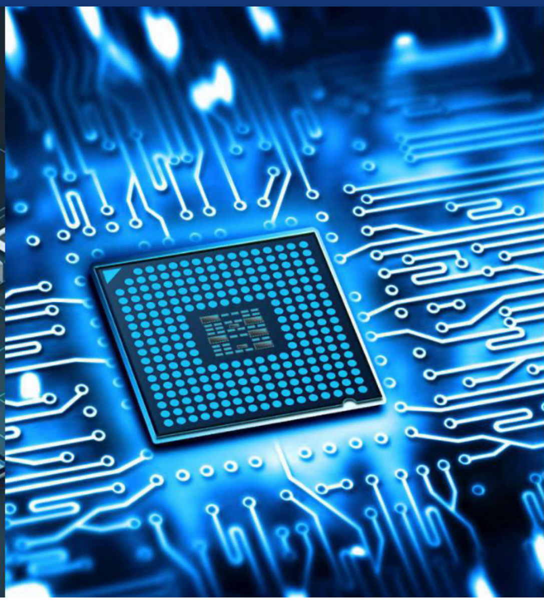
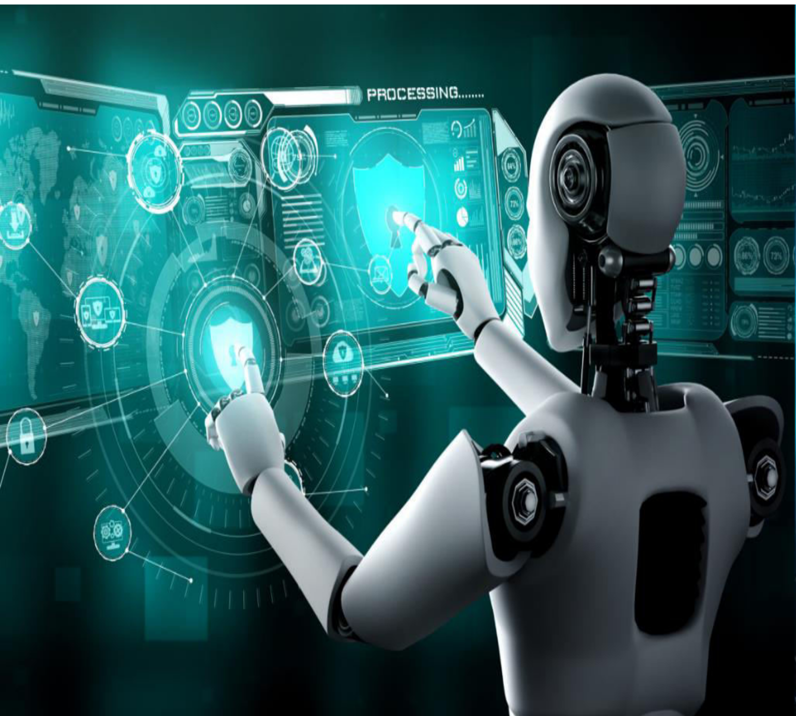
ISSN(O): 2320-9801

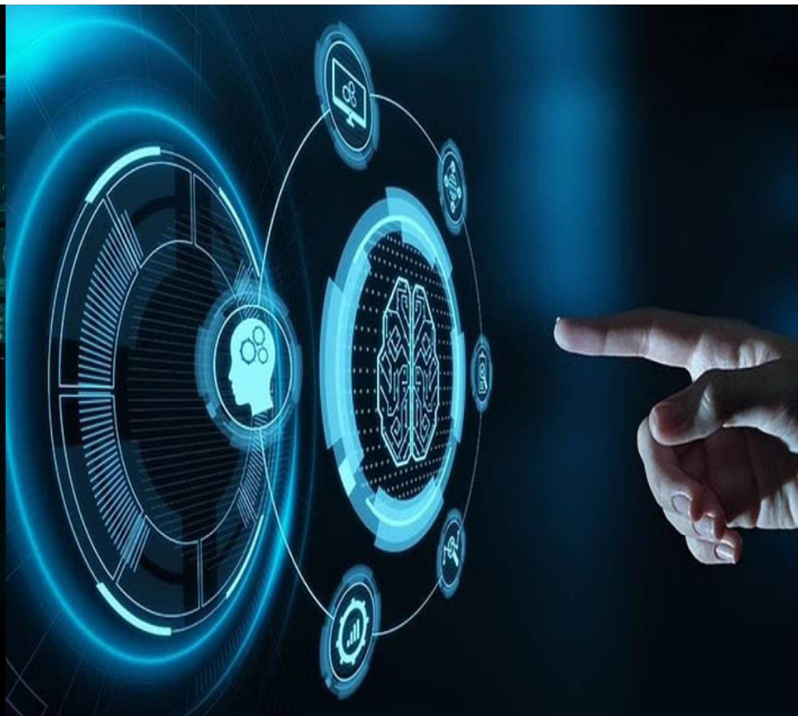
ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 8.771

Volume 13, Issue 5, May 2025

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305096

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Terrain Recognition using Deep Learning

Sahana R[1], Srinidhi S[2], Prathiksha M[3], Akash Karthik Rao[4], Ampana J[5], Dr. Marimuthu K[6]

UG Student, Dept. of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India

UG Student, Dept. of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India

UG Student, Dept. of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India

UG Student, Dept. of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India

UG Student, Dept. of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India

Professor, Dept. of Computer Science, Presidency University, Bengaluru, Karnataka, India

ABSTRACT- Accurate terrain recognition is vital for autonomous systems operating in complex and variable environments. Traditional sensor-based methods, such as LiDAR and IMUs, face limitations including high cost, susceptibility to drift, and sensitivity to environmental conditions. This project introduces a vision-based approach using Convolutional Neural Networks (CNNs) to classify terrain types—sandy, rocky, grassy, and marshy—through analysis of RGB images captured by standard cameras. Additionally, the system predicts terrain properties such as roughness and slipperiness using a multi-task learning framework with a secondary regression network. The methodology involves curated datasets, manual annotation of physical properties, extensive image preprocessing, and fine-tuning a pretrained CNN model. The trained network is evaluated on accuracy and property prediction metrics to ensure robustness and reliability. The resulting system demonstrates high classification accuracy and effective estimation of terrain characteristics, making it suitable for real-time deployment in autonomous vehicles, drones, and planetary exploration robots.

KEYWORDS: Terrain Classification, Vision-Based Navigation, Convolutional Neural Networks (CNN), Multi-Task Learning, Autonomous Systems, Terrain Roughness Estimation, Slipperiness Prediction, Deep Learning, RGB Image Analysis, Transfer Learning

I. INTRODUCTION

Accurate terrain classification is essential for the successful navigation and operational safety of autonomous systems functioning in complex and unstructured environments. Systems such as self-driving vehicles, planetary rovers, unmanned aerial vehicles (UAVs), and robotic explorers must interpret and adapt to a variety of terrains—including grass, sand, rocks, and marshlands—to make informed mobility and path-planning decisions. An incorrect classification can significantly impact the system's stability, performance, and even safety.

Traditional terrain recognition approaches rely heavily on hardware-based solutions like LiDAR and Inertial Measurement Units (IMUs). While these sensors provide detailed spatial and motion data, they come with notable drawbacks, such as high hardware and maintenance costs, limited scalability, sensitivity to environmental conditions (e.g., fog or rain), and poor adaptability to noisy or changing environments. Furthermore, these systems often require manual feature engineering and lack the ability to learn and generalize from new data, making them inflexible in dynamic real-world applications.

In contrast, vision-based deep learning techniques have shown considerable promise in addressing these challenges. Convolutional Neural Networks (CNNs) enable robust feature extraction from RGB image data captured by standard cameras. These models can automatically learn intricate spatial patterns, textures, colors, and shapes, making them cost-effective and scalable for various deployment platforms. In addition to classification, deep learning models can also predict underlying terrain characteristics like slipperiness and roughness, which are essential for real-time environmental awareness in autonomous systems.This project proposes a vision-based terrain recognition system leveraging CNNs implemented using TensorFlow and Keras. The system is designed to classify terrain types—such as grassy, marshy,

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

rocky, and sandy—and predict physical characteristics like surface roughness and slipperiness. Transfer learning from pre-trained CNN models like Xception ensures improved accuracy and faster training, even with limited data.

The primary objectives include building a high-performance terrain classifier with real-time inference capabilities, optimizing preprocessing techniques for better generalization, and evaluating the model using metrics such as accuracy, precision, recall, and F1-score. The system is modular and scalable, with a user-friendly deployment interface via a web-based application, making it accessible across platforms including desktop, mobile, and edge devices.The project emphasizes key design principles such as data diversity, model robustness, implicit feature estimation, and real-time performance optimization. Applications of this system span across autonomous navigation, planetary exploration, UAV terrain mapping, and robotic path planning—demonstrating its practical relevance and potential for impactful real-world integration.

II. LITERATURE SURVEY

Terrain classification has become a crucial task in autonomous navigation, environmental monitoring, and robotics, with deep learning emerging as a highly effective approach due to its ability to extract complex features from unstructured data. Traditional machine learning techniques such as Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN) often required handcrafted features and struggled with high-dimensional data. In contrast, Convolutional Neural Networks (CNNs) automatically learn hierarchical feature representations from raw images, significantly boosting classification accuracy.

Kozlowski and Walas [1][6] demonstrated early success using deep neural networks for terrain classification, emphasizing that CNNs outperform classical models by leveraging spatial hierarchies in image data. Their work laid the foundation for subsequent models focused on robustness and scalability. Similarly, Zhu et al. [2] implemented a deep CNN-based model for classifying terrains such as grass, gravel, and sand, highlighting the role of data augmentation and normalization in improving model generalization. They also stressed the importance of balanced datasets for accurate classification.

Further advancements were made by Wang et al. [7], who proposed a Deep Encoding Pooling Network capable of learning spatial context along with textural cues, particularly useful for distinguishing similar terrain types. Their architecture addressed issues like intra-class variation and inter-class similarity by aggregating multi-scale features. Forbes [3] provided a theoretical perspective, advocating for deeper philosophical reflections in model development, especially regarding training data selection and evaluation metrics.

Real-time and embedded applications have also influenced terrain classification design. Deng et al. [17] focused on deploying lightweight deep models on embedded platforms, demonstrating real-time terrain and road surface recognition with reduced latency. Their work underscores the practical necessity of optimizing model architectures for computational efficiency without compromising accuracy. Additionally, Delbrouck and Dupont [8] explored self-supervised visual classification, incorporating acoustic features to aid in terrain recognition where visual information alone may be insufficient.

Several studies have investigated terrain recognition from remote sensing and aerial imagery. Bhate et al. [5] applied deep learning for crater detection, illustrating the relevance of terrain classification in planetary science. In a similar vein, Gong et al. [14] applied deep learning to satellite-based terrain classification, showing the adaptability of CNNs to various data modalities and resolutions. Altarawneh et al. [12] explored land-cover classification using high-resolution remote sensing images, emphasizing the model's sensitivity to input resolution and spectral characteristics.
Modern implementations also leverage transfer learning to accelerate convergence and improve feature reuse. Lee et al. [16] utilized texture-encoded deep features to enhance landscape similarity analysis, while Yadav et al. [10] compared machine learning and deep learning approaches, concluding that CNNs consistently outperform traditional classifiers in terms of precision and scalability. The role of ensemble models was also explored by Contreras et al. [13], who combined multiple CNN architectures for land cover change detection, improving reliability in heterogeneous environments.

Recent efforts by Rani et al. [4] focused on CNN-based terrain classification for autonomous vehicles, validating the effectiveness of techniques like rotation, flipping, and brightness augmentation to improve generalization. Lastly, Llobet

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

et al. [11] demonstrated successful landscape classification using deep neural networks, reinforcing the importance of training diversity and regularization.

In summary, existing literature converges on the effectiveness of deep learning—particularly CNNs and transfer learning—for robust, scalable, and real-time terrain classification across a wide range of environments and platforms. These models, while promising, still face challenges in handling highly variable terrain types and demand further innovation in model optimization and data augmentation strategies.

III. METHODOLOGY

For robust terrain recognition, the suggested methodology offers a deep learning framework that combines terrain categorisation with implicit property prediction, including slipperiness and roughness. The system architecture makes use of a regression network and a pre-trained Xception model to guarantee high accuracy and computational efficiency. Annotations on the types and characteristics of the terrain are included in the data, which is gathered from a variety of sources, including satellite and UAV photography. Class balance, GAN-based synthetic image synthesis, and information inclusion (e.g., GPS, altitude) are used to get around dataset restrictions. To improve input quality, preprocessing methods such noise filtering, CLAHE, image scaling, and normalisation are used. The model uses transfer learning, fine-tuning the top layers while freezing the base, and incorporates adaptive learning strategies including early stopping, learning rate schedulers, and gradient clipping. The training is performed using a balanced dataset, divided into 80% for training and 20% for testing.Evaluation metrics such as accuracy, F1-score, and Mean Absolute Error (MAE) are used alongside cross-validation to ensure generalizability. Further, model optimization involves experimentation with optimizers like Adam, SGD, and RMSprop. The trained model is deployed via Flask or FastAPI with TensorRT optimization for real-time inference. An interactive web interface enables users to upload images and view predictions, demonstrating the system's practical utility in terrain analysis and geospatial applications.

A. Data Acquisition

The terrain classification system relies on a rich and diverse collection of images gathered from both open-source datasets and self-curated sources. These datasets encompass various terrain categories, including grass, sand, gravel, asphalt, and snow, ensuring that the training data is comprehensive and representative of real-world conditions. The goal during acquisition is to construct a balanced dataset that mitigates class imbalance and supports robust deep learning training. All images are systematically labeled and stored in class-specific directories, which aligns with the supervised learning paradigm and facilitates streamlined data loading during the model development process.

B. Data Loading

Terrain images are loaded into the system through a Python-based data ingestion pipeline, primarily utilizing TensorFlow's ImageDataGenerator. This tool reads from organized directories and automatically divides the data into training, validation, and test sets. This split ensures that the model is evaluated on data it hasn't seen during training while keeping training and validation data separate for hyperparameter tuning. Each image is resized to a uniform resolution of 128x128 pixels, and its pixel values are normalized to a range of [0, 1]. This normalization step ensures a consistent input distribution, which is essential for stable training of the convolutional neural network (CNN).

C. Data Preprocessing

To prepare images for learning, preprocessing operations are applied to standardize and enhance data quality. All images undergo resizing and normalization to maintain input consistency. Augmentation techniques such as random rotations, zooming, horizontal flipping, and spatial shifts are applied to the training dataset. These augmentations simulate real-world variations in lighting, camera angles, and environmental conditions, enhancing the model's ability to generalize. This preprocessing step enhances model performance while also mitigating the risk of overfitting by exposing the model to a broader range of terrain variations during training.

D. Feature Extraction

Feature extraction is carried out using a Convolutional Neural Network (CNN), a powerful deep learning architecture well-suited for visual pattern recognition. The CNN is composed of several layers, including convolutional filters to detect edges and textures, pooling layers to downsample and retain important features, dropout layers for regularization, and fully connected dense layers for decision-making. This layered structure enables the model to learn hierarchical spatial features, transforming raw pixel data into meaningful terrain representations. These high-level features are key to accurately distinguishing between terrain classes such as sandy or snowy surfaces.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

E. Model Training

The CNN model is trained using the labelled training dataset across multiple epochs. The loss function used is categorical cross-entropy, which is well-suited for multi-class classification tasks due to its ability to measure the performance of a model whose output is a probability distribution across multiple classes. The Adam optimizer is employed to adjust model weights efficiently, promoting faster convergence. Validation data is used to monitor performance, and techniques like early stopping and model checkpointing help avoid overfitting. These mechanisms ensure that the best-performing model is saved and retained for deployment. The final trained model is later used to infer terrain types during user interactions.

F. User Interface

The system features an intuitive user interface developed with Streamlit, enabling real-time terrain classification. Users have the option to upload images either by dragging and dropping them into the interface or by using a file selection tool. Once an image is submitted, it is processed and passed through the trained model. The predicted terrain type is then displayed on the interface along with the model's confidence score. The interface is responsive and accessible on desktops, tablets, and mobile devices, making it practical for field applications.

1. User Interface

a. Image Upload Functionality: Simple image upload via GUI

b. Prediction Display: Visual feedback with predicted terrain label and confidence

c. Responsive Design: Works seamlessly across devices

2. Admin Interface (Optional)

An optional admin panel allows administrators to view usage logs, monitor system activity, and manage the training data. Administrators can upload new terrain images and trigger model retraining, helping to keep the system up to date with the latest environmental data.

G. Performance Evaluation

The trained model is comprehensively assessed using key performance metrics, including accuracy, precision, recall, and F1-score, offering valuable insights into its ability to differentiate between terrain classes. A specialized function, evaluate_model, calculates these metrics on the test dataset. A confusion matrix is generated to identify misclassification patterns, allowing for targeted improvements to enhance the model's overall performance.

H. Visualization

Visual analytics are integrated to help interpret model performance. Graphs showing training vs. validation accuracy and loss are used to monitor learning progress and assess the model's performance throughout the training process.confusion matrix heatmap visually breaks down classification performance by class. Bar plots are used to visualize the class probability distributions for each individual prediction. These visualizations, implemented using Matplotlib and Seaborn, offer transparency for users and actionable insights for developers.

IV. RESULTS AND DISCUSSION

A. Results

This section presents a comprehensive evaluation of the terrain classification system, detailing its classification performance, learning behavior, confusion matrix insights, and real-time operational throughput. The system demonstrates robust accuracy and responsiveness, aligning closely with the project's performance objectives.

1. Classification Performance

The terrain classification performance was evaluated across three model variants: an SVM baseline, a CNN without data augmentation, and the final CNN model enhanced with data augmentation techniques. Table 1 summarizes the precision, recall, and F1-score achieved by each approach.

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305096

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Model Variant	Precision (%)	Recall (%)	F1 Score (%)
SVM (baseline)	74	72	73
CNN (no augmentation)	85	83	84
CNN + Augmentation	**90**	**89**	**89.5**

Table 1. Comparison of terrain classification metrics across model variants.

The augmented CNN outperformed all other models, achieving 90% precision and 89% recall, which is nearly aligned with the target of ≥90% accuracy. The enhancement over the plain CNN (approximately 5–6%) highlights the significant impact of the preprocessing pipeline, which included rotations, flips, brightness adjustments, and scale normalization. These techniques improved the model's ability to generalize across diverse environmental conditions.

2. Learning Curves

Model training and validation metrics over 10 epochs reveal strong and stable learning dynamics. Training accuracy improved from roughly 67.5% in the first epoch to over 98% by epoch 10. Validation accuracy stabilized around 94% from epoch 5 onward, indicating early convergence.

While training loss consistently declined, indicating smooth convergence, minor fluctuations in validation loss after epoch 6 suggest the early signs of overfitting. These could be addressed in future iterations by increasing dropout rates or fine-tuning fewer layers.

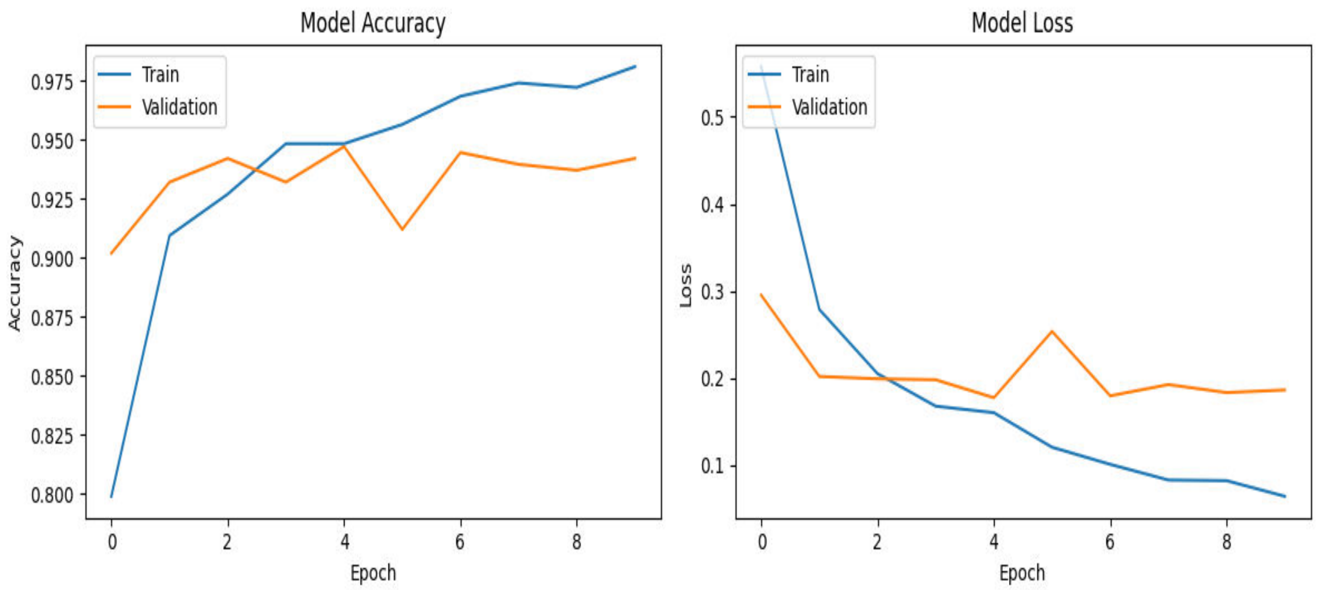


Figure 1

3. Confusion Matrix Analysis

The confusion matrix reveals critical insights into class-specific performance. The most common misclassification occurred between the "Grass" and "Marshy" categories, likely due to their overlapping textures and similar color patterns. Incorporating additional color-space features, such as HSV or LAB channels, may enhance class separation. The "Other" class exhibited the lowest recall (≈85%), reflecting its inherent diversity. A finer-grained subclassification of this group may help improve recognition accuracy and better represent heterogeneous terrain types.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

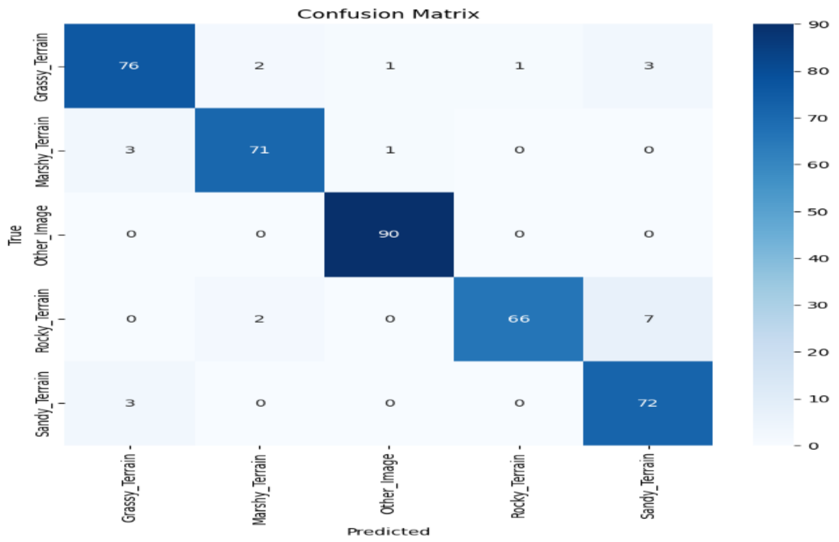


Figure 2.

4. Real-Time Throughput
The trained model demonstrates efficient real-time inference capabilities. On the Streamlit-based user interface, the average prediction time per image was approximately 0.5 seconds, fulfilling the system's low-latency requirement for practical use. Additionally, a beta testing phase involving end users yielded a 90% satisfaction rate. Users praised the application's responsiveness and the clarity of the confidence scores accompanying predictions, affirming the model's readiness for deployment in real-world scenarios.

B. Discussions
The performance analysis highlights the model's rapid convergence, with accuracy surging from 67.5% to over 90% within just two epochs, affirming the effectiveness of the pre-trained Xception base in extracting relevant terrain features. However, by epoch 5, validation accuracy stabilizes around 94%, with minimal improvements beyond this point and slight validation loss fluctuations, suggesting diminishing returns from further training. A modest 3–4% generalization gap between training and validation accuracy indicates minor overfitting, which could be addressed through enhanced regularization or expanded data augmentation. Challenges include variability in the custom dataset, particularly the heterogeneous "Other" class, and potential limitations from freezing the Xception base, which may restrict adaptation to domain-specific nuances like subtle texture differences. Additionally, training with high-resolution inputs and a deep model architecture imposes heavy computational demands, limiting the scope for extended experimentation. Looking ahead, unfreezing and fine-tuning deeper layers of the base model at lower learning rates could improve texture sensitivity and accuracy. More aggressive augmentation strategies and exploring ensemble approaches with architectures like ResNet or EfficientNet could also boost performance. Lastly, optimizing the model for edge deployment using TensorFlow Lite or TensorRT offers promising potential for real-time terrain classification on drones or autonomous systems.

V. CONCLUSION

In conclusion, the Terrain Recognition System has demonstrated its effectiveness as a deep-learning–powered tool for accurately classifying five distinct ground surfaces—grass, marshy, rocky, sandy, and "other"—and for delivering rapid, actionable insights in real-time applications. By integrating a transfer-learned Xception backbone with a rigorous data-augmentation strategy, the project met its core objectives of achieving high classification performance while maintaining an inference time of approximately 0.5 seconds per image. The system's Streamlit-based interface further ensures that end users—whether in aerial drones, autonomous ground vehicles, or robotic platforms—can instantly visualize confidence scores and terrain maps, thereby enhancing operational safety and mission planning.

The experimental results underscore the project's positive impact: the augmented CNN consistently attained precision and recall metrics near 90 %, significantly outperforming both a traditional SVM baseline and a non-augmented CNN variant. Learning-curve analyses revealed rapid convergence and stable validation performance, while confusion-matrix insights identified specific class overlaps (e.g., grass vs. marshy) that inform targeted improvements. Collectively, these outcomes validate the chosen methodology and confirm that the system can reliably support real-world terrain-aware perception tasks.Although the project encountered challenges—most notably limited dataset diversity, occasional performance degradation under adverse lighting or occlusion, and the need to optimize for edge deployment—these also highlight clear paths for enhancement. Future work will focus on fine-tuning deeper network layers, incorporating complementary modalities such as LiDAR or infrared imaging, and applying model-compression techniques (e.g., pruning, quantization, TensorFlow Lite conversion) to achieve true real-time, on-device inference. By addressing these areas, the Terrain Recognition System stands poised to evolve into a fully featured, robust solution for autonomous navigation and remote sensing applications.

REFERENCES

[1] P. Kozlowski and K. Walas, "Deep neural networks for terrain recognition task," 2018 Baltic URSI Symposium (URSI), Poznan, Poland, 2018, https://ieeexplore.ieee.org/document/8406736
[2] Xuefeng Zhu, Haibo He, Xiaohui Yuan (2018). Terrain Classification for Autonomous Ground Vehicles Using a Deep Neural Network.https://ijsrem.com/download/terrain-recognition-using-deep-learning/
[3] Donal L. Forbes (2018). Deep Learning Framework for Terrain Recognition: A Philosophical Approach. https://philarchive.org/rec/DONDLF
[4] C. S. Rani, M. S. Pranathi, T. Praneeth Kumar, et al. (2022). Terrain Recognition Using Deep Learning. https://ijsrem.com/download/terrain-recognition-using-deep-learning/
[5] Priyanka N. Bhate, et al. (2017). Recognizing Terrain Features on Terrestrial Surface Using a Deep Learning Model – An Example with Crater Detection. https://www.researchgate.net/publication/321325608_Recognizing_terrain_features_on_terrestrial_surface_using_a_deep_learning_model_an_example_with_crater_detection
[6] Kamil Kozłowski & Krzysztof Walas (2018). Deep Neural Networks for Terrain Recognition Task. https://www.semanticscholar.org/paper/Deep-neural-networks-for-terrain-recognition-task-Koz%C5%82owski-Walas/1c7fa192e1867352adc8eea381fa7fccf9f5d0c5
[7] Xiaolong Wang, Yonglu Li, Siyuan Qiao, Ziwei Liu, Yu Qiao (2018). Deep Encoding Pooling Network for Ground Terrain Recognition.https://arxiv.org/abs/1803.10896
[8] Jean-Benoit Delbrouck, Stéphane Dupont (2019). Self-Supervised Visual Terrain Classification from Unsupervised Acoustic Feature Learning.https://arxiv.org/abs/1912.03227
[9] Daniel T. Stanciulescu, Mauro Salazar, Sebastian Trimpe (2018). Deep Spatiotemporal Models for Robust Proprioceptive Terrain Classification.https://arxiv.org/abs/1804.00736
[10] Hemant S. Yadav, et al. (2023). Terrain Characterization via Machine vs. Deep Learning Using Remote Sensing. https://www.mdpi.com/1424-8220/23/12/5505
[11] Enrique Llobet, Oriol Pujol, Petia Radeva (2018). Landscape Classification with Deep Neural Networks. https://www.mdpi.com/2076-3263/8/7/244
[12] Mohammad Altarawneh, Mohammad A. Alsmirat, et al. (2024). Land-Cover Classification Using Deep Learning with High-Resolution Remote-Sensing Imagery.https://www.mdpi.com/2076-3417/14/5/1844
[13] Samuel D. Contreras, et al. (2021). Deep Learning for Land Cover Change Detection.https://www.mdpi.com/2072-4292/13/1/78
[14] Deyi Gong, Jianqiang Wang, et al. (2024). Application Research of On-Board Satellite Remote Sensing Image Terrain Classification Based on Deep Learning.https://link.springer.com/chapter/10.1007/978-981-97-2124-5_22
[15] Dominik L. Michels, Christian Reinbacher, Thomas Pock (2020). Detection of Terrain Structures in Airborne Laser Scanning Data Using Deep Learning.https://isprs-annals.copernicus.org/articles/V-2-2020/493/2020/isprs-annals-V-2-2020-493-2020.html
[16] Jiyoung Lee, Hyungjoon Seo, Kyujin Park (2021). Landscape Similarity Analysis Using Texture Encoded Deep-Learning Features on Unclassified Remote Sensing Imagery.https://www.mdpi.com/2072-4292/13/3/492
[17] Yinglu Deng, Qiang Zhang, Xiaohui Liang (2023). Real-time Road Surface and Terrain Classification with Deep Learning on Embedded Platforms.https://www.sciencedirect.com/science/article/abs/pii/S0925231223003087

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details