Terrain-Aware Tile Classification and Optimal Path Planning Using CNNs

Name: Shashank Ranjan
Enrollment Number: 25113130
Branch: Civil Engineering
Academic Year: 2025–2029
GitHub Repository: github.com/shashank0r/mosaic-image

1 Introduction

Vision-based navigation requires accurate understanding of both global scene context and local traversability. End-to-end models often struggle to capture such structure while respecting spatial constraints. This project adopts a hierarchical approach that explicitly separates perception and planning into multiple stages.

The proposed pipeline prioritizes engineering robustness and interpretability over geometric perfection. This makes it suitable for structured datasets where annotations and constraints are predefined.

2 Problem Formulation

Each input image represents a navigable environment with a defined start and goal location. The environment is represented as a 20×20 grid, where each grid cell belongs to one of the following classes:

- 0 — Walkable
- 1 — Wall (blocked)
- 2 — Hazard
- 3 — Start
- 4 — Goal

Each image is also associated with a terrain label from the set {desert, lab, forest} and a velocity boost grid that influences traversal cost.

The objective is to compute an optimal path from the start tile to the goal tile using four-directional movement.

3 System Overview

The complete pipeline consists of the following stages:

1. Terrain identification using a global CNN
2. Approximate decomposition of the image into a 20×20 tile grid
3. Terrain-specific tile classification using CNNs
4. Enforcement of unique start and goal locations
5. Cost map construction using terrain and velocity information
6. Shortest path computation using Dijkstra's algorithm

This hierarchical decomposition reduces learning complexity and allows deterministic corrections at intermediate stages.

4 Terrain Identification CNN

A high-capacity convolutional neural network is used to classify the global terrain of each image. Images are resized to 128×128 and passed through four convolutional blocks consisting of convolution, batch normalization, ReLU activation, and max pooling.

The network outputs one of three terrain classes. This prediction determines the selection of the terrain-specific tile classifier and the cost parameters used during path planning.

5 Tile-Based Representation

5.1 Approximate 20×20 Tiling

The image is divided into a uniform 20×20 grid. Due to image resolution constraints, this division does not always produce perfectly uniform tiles, particularly near image boundaries. However, the majority of tiles remain approximately uniform.

This approximation was selected because it aligns with ground-truth annotations, avoids complex grid detection heuristics, and provides a simple and robust representation for tile-level classification. The minor non-uniformity does not significantly affect semantic learning or navigation performance.

5.2 Terrain-Specific Tile CNNs

For each terrain type, a separate tile classification CNN is trained. Each tile is resized to 32×32 pixels and classified into one of five semantic classes.

The tile CNN architecture consists of three convolutional blocks followed by fully connected layers with dropout regularization. Training separate models for each terrain reduces class confusion arising from large visual differences between terrains.

6 Start and Goal Enforcement

CNN predictions may violate structural constraints by producing multiple start or goal tiles, or by missing them entirely. To ensure correctness, a deterministic heuristic is applied:

1. All predicted start and goal labels are suppressed.
2. The tile with highest confidence for the start class is assigned as the unique start.
3. The tile with highest confidence for the goal class, excluding the start tile, is assigned as the goal.

This guarantees exactly one start and one goal per image.

7 Cost Map Construction

Traversal cost for each tile is computed as:

$$\text{Cost} = \text{Base Terrain Cost} - \text{Velocity Boost}$$

Walls are assigned prohibitively large costs, hazards incur higher penalties, and velocity boosts reduce traversal cost. This formulation integrates learned perception with physical constraints.

8 Path Planning

An optimal path is computed using Dijkstra's algorithm on the 20×20 grid under four-directional movement constraints. The resulting path is converted into a sequence of moves (U, D, L, R) and exported in CSV format for submission.

9 Conclusion

This project demonstrates that a hierarchical CNN-based pipeline, combined with classical graph search and deterministic heuristics, can effectively solve terrain-aware navigation problems from visual input.

A key limitation of the proposed approach lies in the uniform 20×20 image tiling strategy. Since the image resolution is not always perfectly divisible by 20, some tiles—particularly near image boundaries—are not strictly uniform in size. This introduces minor geometric inconsistencies and violates the assumption of perfectly aligned tiles. However, this design choice was made deliberately to prioritize simplicity, alignment with training images. Empirically, the majority of tiles remain approximately uniform as a result, the tile classification CNNs are not significantly impacted by these inconsistencies.

Future work could explore adaptive tiling strategies, learned grid inference, or boundary-aware tile normalization to address this limitation more rigorously.