Landslide Detection Using Deep Learning and Multispectral Satellite Images

Geethanjali Javvadi

Department of ECE
GVPCEW Madhurawada

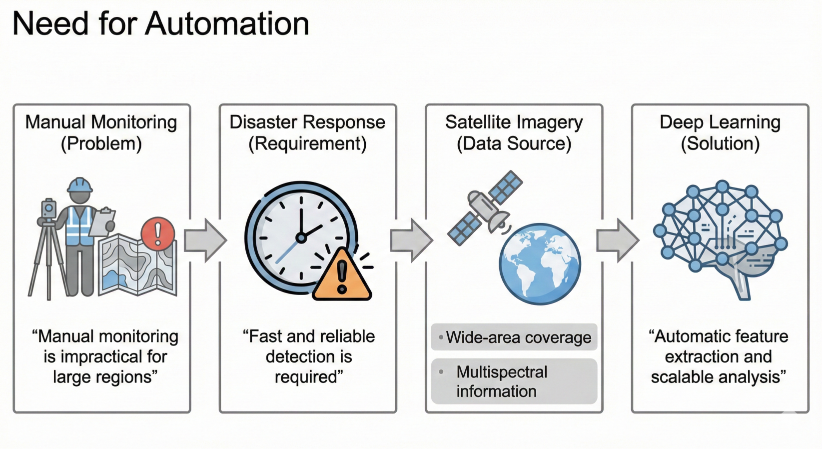
January 14, 2026

Table of Contents

Problem Statement

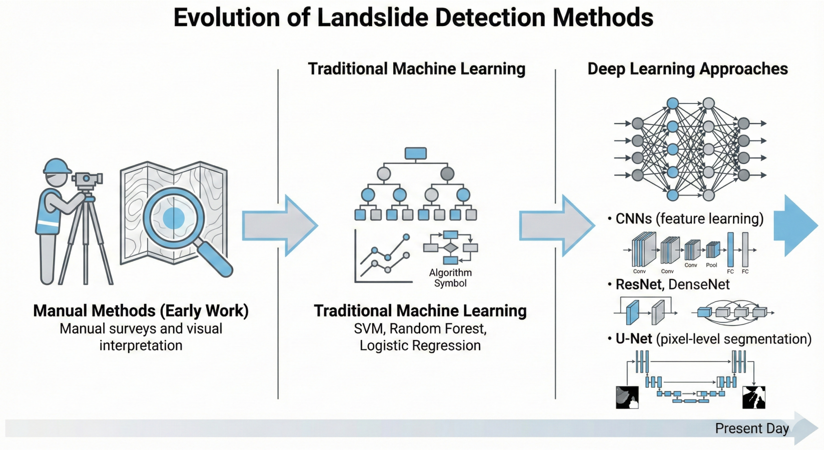
- Landslides are among the most destructive natural disasters.
- They cause severe loss of:
  - Human life
  - Infrastructure
  - Natural resources

Need for Automation



- Manual monitoring is impractical for large geographical regions.
- Disaster response requires fast and reliable detection.

Previous Work

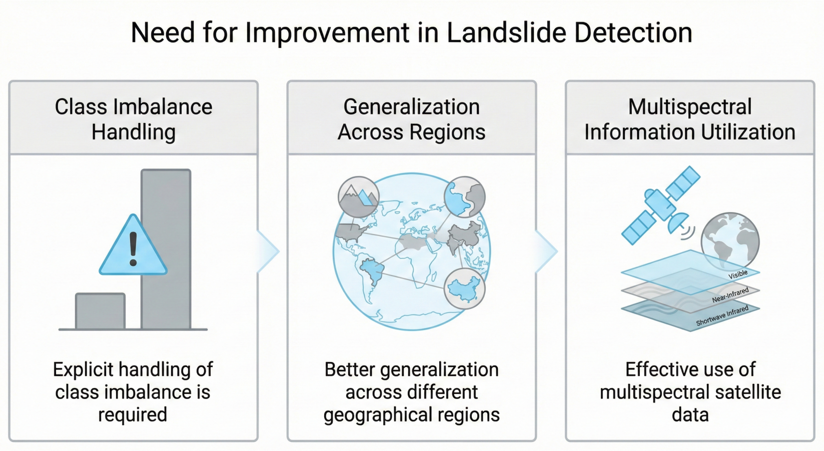


- Landslide detection methods evolved from manual surveys to machine learning and deep learning approaches.
- Deep learning models enable automatic and robust feature learning compared to traditional methods.

Limitations of Previous Work

- Severe class imbalance in landslide datasets.
- Deep models overfit due to limited landslide samples.
- Underutilization of multispectral satellite data.
- Bias in fully connected classifiers.

Need for Improvement

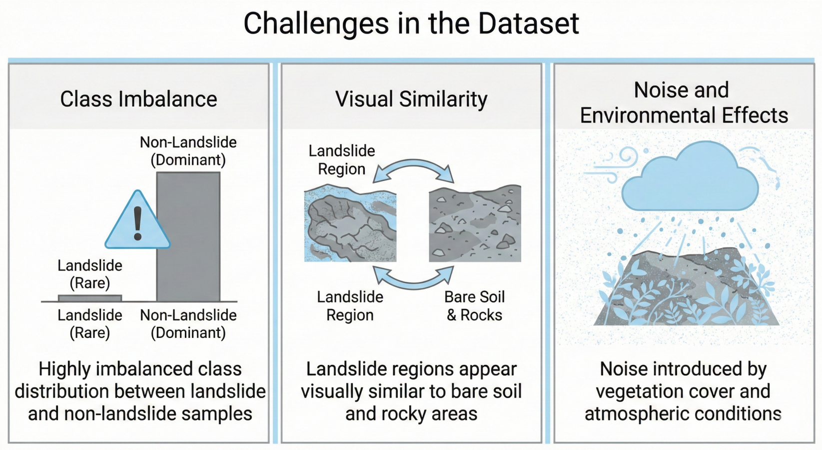


- Existing methods lack effective class imbalance handling and generalization.
- Improved utilization of multispectral data is essential for robust landslide detection.

Dataset Description

- Multispectral satellite dataset from **Zindi Africa**.
- Acquired via Landslide Detection competition.
- Each sample:
  - 64×64 pixels
  - 12 spectral bands
  - Stored in `.npy` format
- Two classes: Landslide and Non-landslide.

Challenges in the Dataset



- The dataset suffers from class imbalance, visual ambiguity, and environmental noise.
- These challenges motivate robust data balancing and deep feature learning approaches.

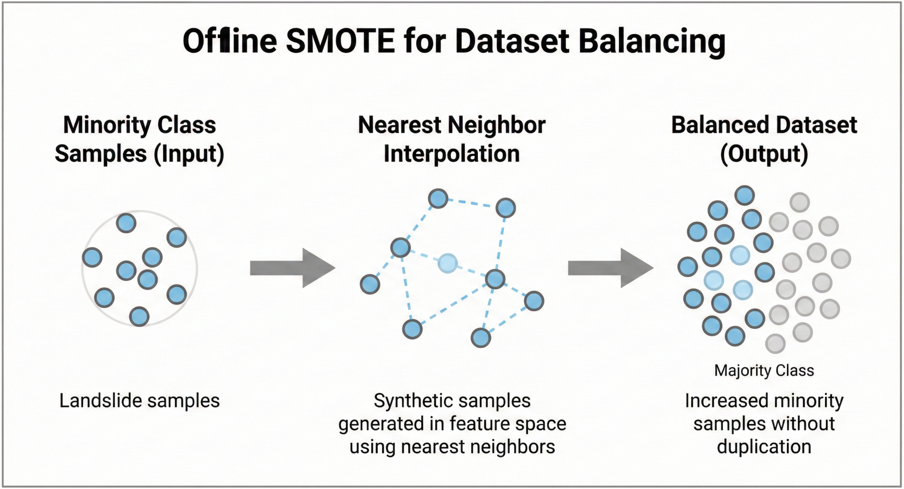
Methodology Overview

- Input multispectral satellite images.
- Preprocessing and normalization of raw data.
- Handling class imbalance using offline SMOTE.
- Creation of a balanced training dataset.
- Deep feature extraction using EfficientNetV2-Large.

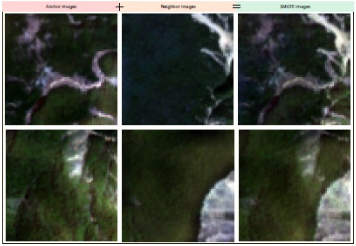
Class Imbalance Problem

- Landslide datasets are highly imbalanced in nature.
- Non-landslide samples significantly outnumber landslide samples.
- This imbalance leads to:
  - Biased learning toward majority class
  - Poor representation of landslide regions
- Addressing class imbalance is essential before training deep models.

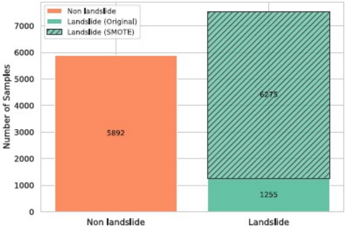
Offline SMOTE for Dataset Balancing



- SMOTE generates synthetic landslide samples by interpolating between nearest neighbors.



(a)



(b)

- After applying SMOTE, landslide and non-landslide classes are better balanced.

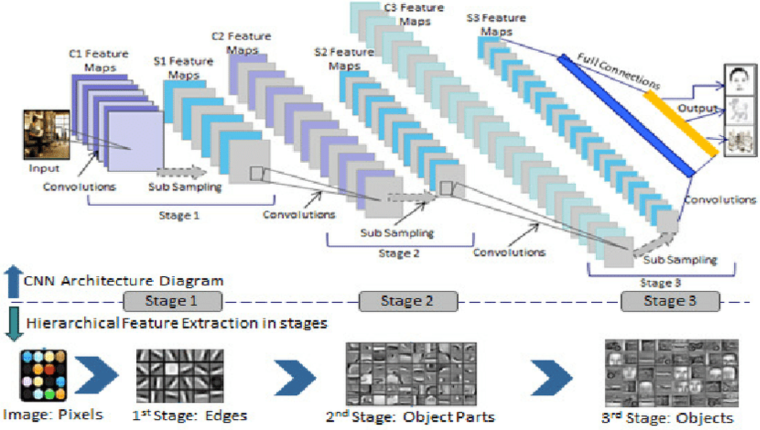
Feature Extraction

- Feature extraction transforms raw multispectral images into meaningful representations.
- Deep features are automatically learned from data without manual design.
- Extracted features capture:
  - Spatial information such as texture and shape
  - Spectral information from multiple satellite bands
- These features are more robust than traditional handcrafted features.

Why CNNs for Feature Extraction

- Convolutional Neural Networks (CNNs) are well-suited for image-based tasks.
- CNNs learn hierarchical features:
  - Low-level features: edges and textures
  - Mid-level features: regions and patterns
  - High-level features: semantic representations
- Local connectivity and weight sharing improve generalization.

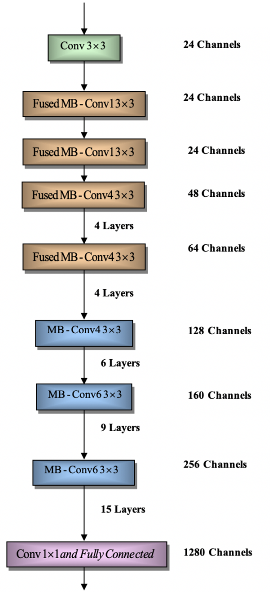
CNN Feature Hierarchy



EfficientNetV2 Architecture Overview

- EfficientNetV2 is a modern and efficient deep convolutional architecture.
- It follows a compound scaling strategy to balance:
  - Network depth
  - Network width
  - Input resolution
- Designed for faster training and improved accuracy.
- Uses Fused-MBConv and MBConv blocks for efficient feature learning.

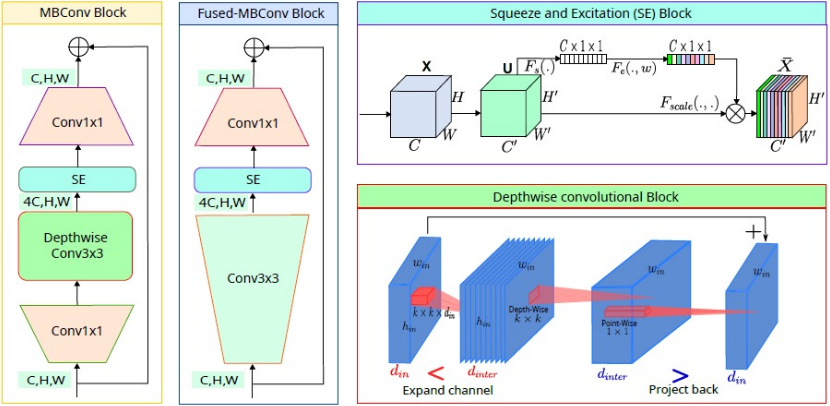
EfficientNetV2 Architecture



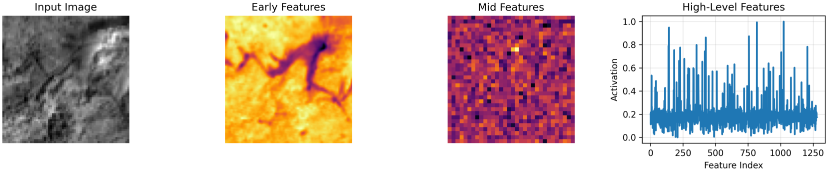
EfficientNetV2 for Multispectral Feature Extraction

- Original EfficientNetV2 is designed for 3-channel RGB images.
- In this work, the input layer is modified to support:
    - 12-channel multispectral satellite images
- Enables effective utilization of terrain, vegetation, and soil information.
- EfficientNetV2 acts as a robust deep feature extractor for landslide detection.

EfficientNetV2-Large Architecture



Feature Extraction Process Visualization



Step-wise visualization of feature extraction in EfficientNetV2: Input multispectral image $\rightarrow$ Early convolutional features $\rightarrow$ Intermediate abstract features $\rightarrow$ High-level feature embedding

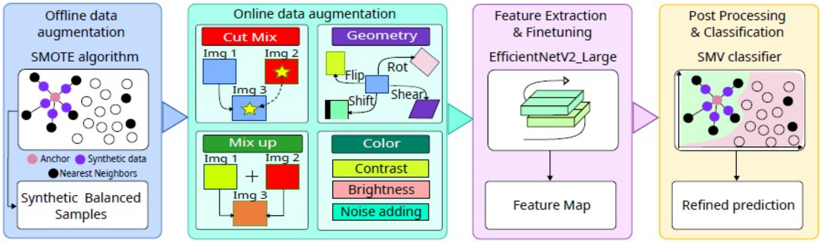
Explanation of Feature Extraction Stages

- **Input Image:**
  - Represents a multispectral satellite patch used for landslide detection.
  - Contains terrain and surface information across multiple spectral bands.
- **Early Feature Maps:**
  - Capture low-level patterns such as edges, textures, and local intensity variations.
  - Useful for identifying abrupt terrain changes.
- **Mid-Level Feature Maps:**
  - Learn higher-order spatial structures and region-level patterns.
  - Encode combinations of textures and terrain characteristics.
- **High-Level Feature Embedding:**
  - Obtained after global pooling in EfficientNetV2.
  - Produces a compact feature vector used for final classification.

Feature Extraction Summary

- Multispectral images are processed using EfficientNetV2-Large.
- The network learns discriminative spatial and spectral features.
- Extracted deep features form the foundation for subsequent classification.
- This stage completes the core representation learning of the framework.

High-Level Framework



Work Completed So Far

- Studied the problem of landslide detection using satellite imagery.
- Reviewed existing literature and identified limitations of prior methods.
- Collected and analyzed multispectral satellite dataset from Zindi Africa.
- Performed dataset understanding and identified class imbalance issues.
- Designed the overall deep learning framework.
- Implemented feature extraction using EfficientNetV2-Large:
    - Adapted the network for 12-channel multispectral input
    - Extracted high-level spatial and spectral features
- Completed the core representation learning stage of the project.

Future Work

- Train the deep learning model with optimized hyperparameters.
- Integrate Support Vector Machine (SVM) for final classification.
- Perform extensive performance evaluation using:
  - Accuracy, Precision, Recall, F1-score
- Analyze results and compare with existing approaches.
- Prepare final documentation and presentation.