Journal of Engineering Issues and Solutions

Land Cover Segmentation from Satellite Imagery Using U-Net with Custom Loss and Morphological Postprocessing

Yamraj Khadka[1], Rajat Pradhan[2*]

[1,2]IoE Purwanchal Campus, Dharan, Nepal
*khadkayamraj2002@gmail.com, *pradhanrajat499@gmail.com*
Received: May 12, 2026; Revised: 13 June, 2026; Accepted: July 11, 2026

Abstract

Land cover classification is a fundamental task in remote sensing with direct applications in environmental monitoring, urban planning, and disaster management. In this project, we design and implement a semantic segmentation pipeline using a modified U-Net architecture from scratch for pixel-level classification of satellite images. The network was trained on the DeepGlobe Land Cover Classification dataset with significant attention to model generalization and class imbalance through the integration of a custom composite loss function that combines Focal Tversky Loss and Weighted Categorical Crossentropy. Data augmentation techniques and per-class weighting were used to address class imbalance, especially in underrepresented classes such as barren land and rangeland. Postprocessing techniques like morphological operations and class-wise median filtering further refined the segmentation outputs. The model achieved a mean Intersection-over-Union (IoU) of 0.67, with notable performance across dominant land types. A full Streamlit-based web interface was developed to enable real-time visualization and interaction with the model's predictions. Additionally, we initiated a fine-tuning phase targeting Nepal-specific satellite imagery to adapt the model to the country's unique geography. A major ongoing challenge involves curating a clean and labeled dataset of Nepalese satellite patches, which is critical for domain adaptation and achieving optimal segmentation performance on local data. The project demonstrates a full end-to-end pipeline from model design to web deployment and sets a foundation for scalable geospatial AI applications tailored for Nepal.

Keywords: *Deep Learning, Land Cover, Morphological Processing, Satellite Imagery, Segmentation, U-Net*

1. Introduction

Accurate land cover classification from satellite imagery plays a critical role in environmental monitoring, urban planning, agriculture, and natural resource management. Traditional methods rely on manual interpretation or shallow machine learning techniques that often fail to capture complex spatial features in heterogeneous landscapes. Recent advances in deep learning, particularly convolutional neural networks (CNNs), have enabled end-to-end learning of spatial hierarchies directly from raw pixel data, significantly improving segmentation accuracy.

Among CNN architectures, U-Net has emerged as a popular and effective model for semantic segmentation tasks due to its encoder-decoder structure with skip connections, which allows precise localization and context aggregation. Originally developed for biomedical image segmentation, U-Net's adaptability has made it suitable for remote sensing land cover classification tasks, as demonstrated in several recent studies.

The DeepGlobe Land Cover Classification Challenge dataset offers a diverse set of satellite images with pixel-wise annotated land cover classes, providing an excellent benchmark for developing and evaluating segmentation models. While many researchers have used prebuilt architectures or transfer learning approaches on this dataset, there is a need to implement and understand U-Net architectures from scratch, apply customized loss functions to handle class imbalance, and integrate morphological postprocessing techniques to improve output quality.

Furthermore, deploying such models in user-friendly applications enables practical use by domain experts and non-technical users. Streamlit, a Python-based web app framework, facilitates rapid development of interactive visualization and prediction interfaces.

In this study, we present an end-to-end pipeline for land cover segmentation from satellite imagery using a U-Net model built from scratch. The pipeline includes dataset loading and augmentation, composite loss functions to mitigate class imbalance, morphological postprocessing to refine masks, and deployment through a Streamlit app. We also discuss ongoing efforts to fine-tune the model for Nepali satellite imagery and the challenges associated with data collection for this specific context.

The key objectives of this research are:

- To implement a U-Net architecture from first principles using TensorFlow Keras.
- To train the model on the DeepGlobe dataset with customized composite losses and class weighting to address imbalanced land cover classes.
- To design and deploy a Streamlit-based application for practical use and visualization of segmentation results.
- To explore fine-tuning strategies on Nepali satellite images, highlighting data acquisition challenges.

2. Related Works

Deep learning, particularly convolutional neural networks (CNNs), has significantly transformed the field of satellite image segmentation by enabling automated understanding of complex visual patterns. One of the most widely adopted architectures in this domain is U-Net [1], which uses an encoder-decoder structure with skip connections to effectively capture both global context and fine-grained spatial details. Originally designed for biomedical segmentation, U-Net has since been successfully applied to a wide range of remote sensing tasks, including land cover mapping, urban boundary detection, and crop classification. Globally, several benchmark datasets such as DeepGlobe [2], ISPRS Potsdam/Vaihingen [3], and SpaceNet [4] have driven the development of robust segmentation models. These datasets offer high-resolution satellite imagery with pixel-level annotations for diverse land types like vegetation, urban areas, rangelands, and water bodies. Researchers have enhanced U-Net-based models using mechanisms like attention modules, composite loss functions (e.g., Dice loss, focal loss), and postprocessing techniques (e.g., morphological operations) to improve performance on underrepresented classes or noisy predictions.

However, a common limitation in these datasets is their geographic focus primarily covering regions in Europe or North America. This creates a domain gap when applying pretrained models to South Asian contexts, such as Nepal, due to differences in terrain, land-use patterns, and atmospheric conditions.

One major bottleneck in deploying deep learning models for land cover classification is the scarcity of pixel-level annotated data, especially in developing countries. Creating such datasets is time-consuming and expensive, as each pixel must be labeled accurately. To overcome this, several techniques have been explored:

- Weakly supervised learning uses partial or sparse labels (e.g., image-level tags or single pixels) to train models [5]. For example, Wang et al. [6] used class activation maps (CAMs) to generate pseudo-masks for segmentation.
- Transfer learning, where models pretrained on large-scale datasets like ImageNet or BigEarthNet are fine-tuned on smaller, domain-specific datasets [7], helps in reducing training data requirements.
- Data augmentation, using transformations like flipping, rotation, and elastic distortions, is widely used to artificially expand training datasets and improve model generalization [8].

In the Nepali context, the use of deep learning for satellite image segmentation is still at an early stage. Traditional approaches have relied on unsupervised classification (e.g., k-means) or rule-based systems using Landsat or Sentinel data for tasks like forest cover monitoring, land-use planning, and urban sprawl analysis [9][10]. For instance:

- Thapa and Murayama (2010) developed a scenario-based urban growth model for Kathmandu Valley using remote sensing and GIS data.
- Uddin et al. (2015) conducted forest change detection using Landsat imagery with conventional classification techniques.

Despite these efforts, semantic segmentation using CNNs, especially architectures like U-Net, remains underexplored for Nepal. No large-scale, publicly available annotated dataset exists that captures Nepal's diverse geography, ranging from the Himalayan highlands to the Terai plains. This lack of data makes it difficult to train models that generalize well to the region's unique terrain and seasonal variations.

In our work, we address these challenges by training a custom U-Net model from scratch using the DeepGlobe land cover dataset. To handle class imbalance a situation where certain landcover types like rangeland or barren land have fewer examples we incorporate a composite loss function that combines weighted categorical cross-entropy and focal Tversk loss. These loss functions assign more importance to underrepresented classes and help in learning sharper boundaries.

We further apply postprocessing techniques such as morphological cleaning (removing small artifacts) and median smoothing to reduce segmentation noise. The model is deployed through a Streamlit-based web application for real-time interaction and visualization. Finally, we plan to fine-tune this model on Nepal-specific satellite imagery (e.g., from EO Browser or Sentinel Hub) to achieve higher relevance and accuracy in the local context.

2.1 Benchmark Comparison on the DeepGlobe Land-Cover Benchmark

A concise performance snapshot helps position our 0.67 mIoU result in the broader DeepGlobe literature (Table X). The original challenge baseline a ResNet-18 + ASPP variant of DeepLab achieved only 0.433 mIoU and highlighted the difficulty of the task

https://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w4/Demir_DeepGlobe_2018_A_

CVPR_2018_paper.pdf . Subsequent work with U-Net variants improved scores into the mid-0.60 range; for example, Rakhlin et al.'s ensemble of U-Nets trained with Lovász-Softmax reached 0.649 mIoU

https://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w4/Rakhlin_Land_Cover_Classification_CVPR_2018_paper.pdf.Transformer-augmented or deeper encoder–decoder models have since pushed the frontier further. A Deeplab V3+ pipeline reported by Su & Chen, originally built for change detection, attained 0.756 mIoU when re-evaluated on the same split https://arxiv.org/abs/1911.12903. Most recently, the Spatially Adaptive Interaction Network (SAINet) introduced dual-branch feature interaction and pushed performance to 0.778 mIoU https://www.nature.com/articles/s41598-025-99428-4.

Our modified U-Net, trained from scratch with a composite Focal-Tversky + WCE loss and lightweight morphological post-processing, achieves 0.674 mIoU while using orders-of-magnitude fewer parameters than SAINet and without pre-training. This puts our method solidly in the upper-mid tier of current results and demonstrates that careful loss engineering and post-processing can close much of the gap with transformer-heavy models.

Table 1: Comparison of Mean IoU on the DeepGlobe Land Cover Dataset

Model / Paper	Year	Core Architecture & Tricks	Reported mIoU
DeepGlobe Baseline (ResNet-18 + ASPP)	2018	DeepLab-style encoder–decoder	43.3
Rakhlin et al. (U-Net + Lovász ensemble)	2018	U-Net encoder swap + Lovász loss, Equi-batch	64.9
Our work	2025	U-Net from scratch + Focal-Tversky ‖ WCE + morphology	67.4
Su & Chen (DeepLab V3+)	2019	Deeplab V3+ + multi-scale training	75.6
SAINet (Spatially Adaptive Interaction Net)	2025	Dual-branch CNN + adaptive interaction modules	77.8

This comparison clarifies the literature gap our study fills: delivering near-state-of-the-art accuracy without transformer complexity, while providing an interpretable pipeline that can be fine-tuned for Nepali imagery.

3. Materials and Methods

3.1 Dataset and Preprocessing:

The primary dataset used is the DeepGlobe Land Cover Classification dataset, publicly available via Kaggle. It can be accessed publicly at the following link:https://www.kaggle.com/datasets/balraj98/deepglobe-land-cover-classification-dataset.It contains high-resolution satellite RGB images and corresponding pixel-level labeled masks covering seven land cover classes: Urban land, Agriculture land, Rangeland, Forest land, Water, Barren land, and Unknown.

Table 2: Land Cover Classes and Corresponding RGB Color Codes

S.No.	name	r	g	b
1	urban_land	0	255	255
2	agriculture_land	255	255	0
3	rangeland	255	0	255
4	forest_land	0	255	0
5	water	0	0	255
6	barren_land	255	255	255
7	unknown	0	0	0

Each mask is provided as an RGB image where each land cover class is encoded as a unique color. We converted these RGB masks into class indices using a color-to-class mapping derived from the provided class dictionary CSV.

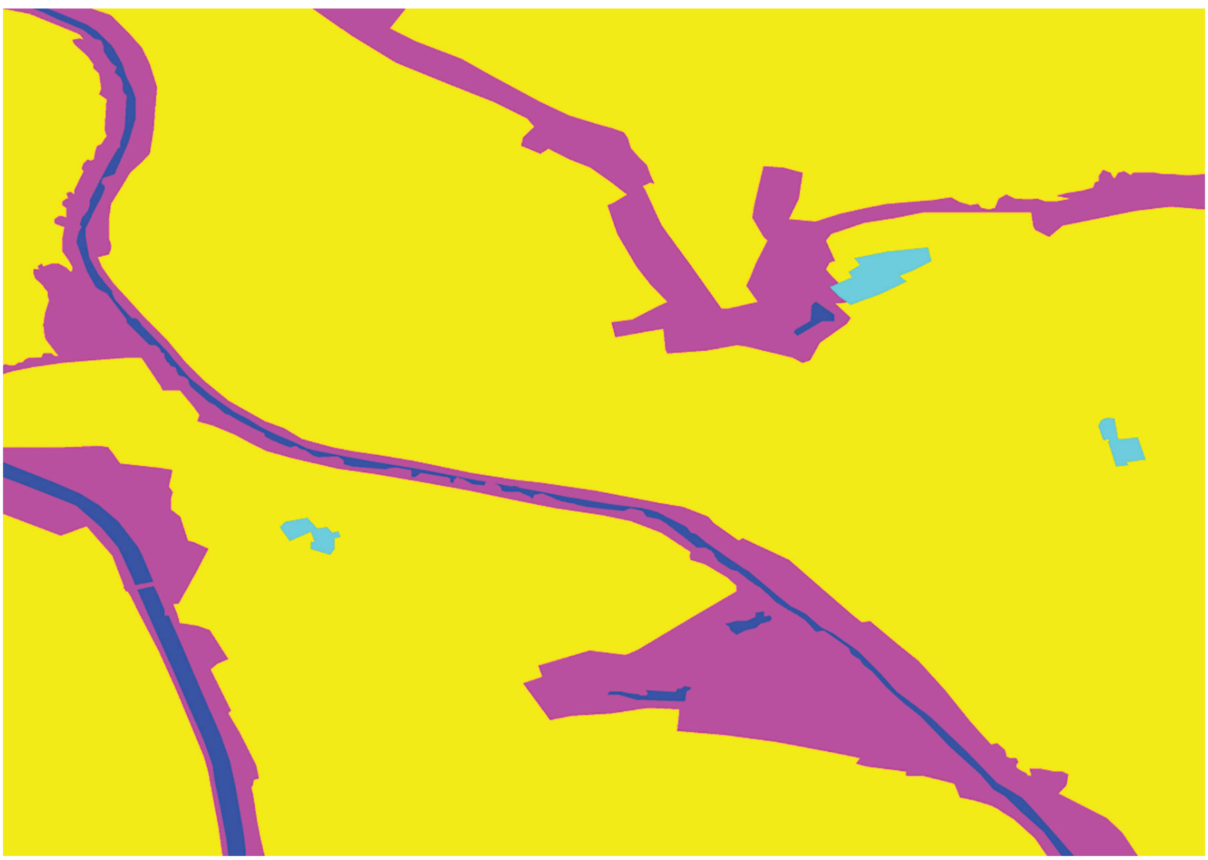

Figure1: Sample mask selected randomly for color-to-class mapping

This mask contains the 4194304 pixels. The pixel distribution for each land cover class in above sample mask given below:

Here is the pixel distribution for each land cover class in the provided image:

Class	Pixel Count
Agriculture	3,457,852
Rangeland	618,874
Water	70,536
Urban Land	19,944
Unmapped	27,098

Figure2: Pixel distribution of above sample mask

Each pixel in the DeepGlobe annotation mask is encoded with an RGB color representing a specific land cover class. To make the labels usable for training a neural network, these colors were mapped to class indices ranging from 0 to 6. A Boolean matching operation was applied to convert the 3-channel RGB mask into a 2D class label map, where:

$$mask_{class}(i,j) = \{k, if\, mask_{rgb}(i,j) = k unchanged, otherwise$$

This process uses pixel-wise comparison of RGB values across the image. For instance, if mask_rgb[150,300]=(255,255,0), the corresponding class index assigned is 1 (agriculture land), based on the predefined color table.This mapping ensures a lossless conversion from annotated RGB masks to numerical label maps suitable for supervised segmentation model training.

To improve model robustness, extensive data augmentation was applied using the Albumentations library. Transformations included random horizontal and vertical flips, brightness and contrast adjustments, shifts, scaling, rotations, and elastic deformations. Augmentation aimed to increase the variability of training samples and help the model generalize better.

All images and masks were resized to 224×224 pixels to standardize input size and reduce computational complexity.

3.2 U-Net Model Architecture:

The U-Net model was implemented from scratch in TensorFlow Keras. The network follows the canonical encoder-decoder pattern with skip connections:

- The encoder path consists of repeated convolutional blocks, each including two convolutional layers with 3×3 kernels, batch normalization, and ReLU activations, followed by max pooling for downsampling.
- The bottleneck layer applies convolutional blocks with dropout for regularization.
- The decoder path performs upsampling via transposed convolutions, concatenating skip connections from the encoder to recover spatial details, followed by convolutional blocks.

The final output layer uses a 1×1 convolution with softmax activation to predict pixel-wise class probabilities across seven classes.

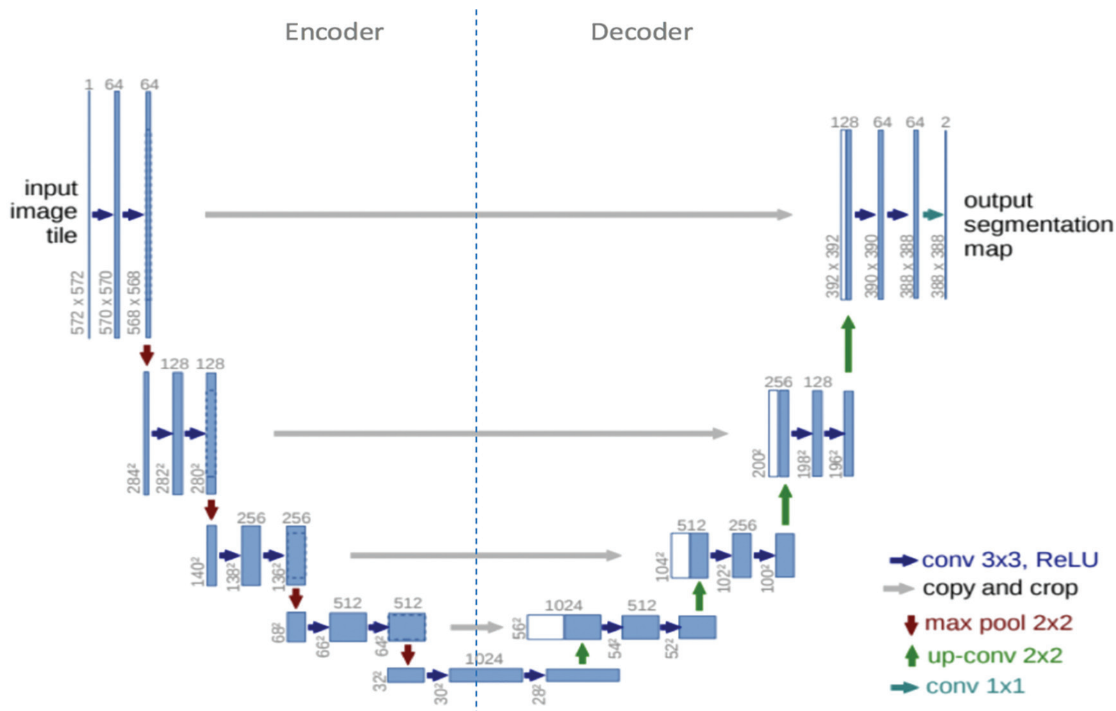

Figure 3: U-Net Architecture

3.3 Loss Function and Metrics

Class imbalance is common in land cover datasets, with some classes covering fewer pixels than others. To address this, a composite loss combining weighted categorical cross-entropy and focal Tversky loss was designed:

a) Weighted categorical cross-entropy applies higher weights to rare classes (notably classes 2 and 5) based on pixel frequency.

b) Focal Tversky loss balances false positives and false negatives, focusing learning on hard-to-classify pixels.

Class imbalance is common in land cover datasets, with some classes covering fewer pixels than others. To address this, a composite loss combining Weighted Categorical Cross-Entropy (WCE) and Focal Tversky Loss (FTL) was used.

a) Weighted Categorical Cross-Entropy:

$$L_{WCE} = -\sum_{c=1}^{C} W_c * y_c * log(\hat{y}_c)$$

Where:

C = number of classes

Wc= class weight for class c

$y_c \in \{0,1\}$ is the ground truth

$\hat{y}_c \in [0,1]$ is the predicted probability for class

b) Focal Tversky Loss:

$$TI(TverskyIndex) = \frac{TP + \varepsilon}{TP + \alpha * FP + \beta * FN + \varepsilon}$$

Where:

TP, FP, and FN are true positives, false positives, and false negatives respectively

α and β are weight factors

ε is a small constant for numerical stability

c) Focal Tversky Loss:

$$L_{FTL} = (1 - TI)^{\gamma}$$

Where:

$\gamma \geq 1$ is a focusing parameter.

d) Combined Loss Function:

$$L_{combined} = \lambda 1 * L_{WCE} + \lambda 2 * L_{FTL}$$

In this study, $\lambda 1 = 0.4$, $\lambda 2 = 0.6$

e) Mean Intersection-over-Union (IoU):

$$IOU = \frac{TP}{TP + FP + FN + \varepsilon}$$

The Mean Intersection over Union (Mean IoU) metric was used to evaluate segmentation quality during training and validation.

3.4 Training Procedure

The model was compiled with the Adam optimizer and trained on the augmented DeepGlobe dataset with an 80-20 training-validation split. Callbacks including ReduceLROnPlateau and EarlyStopping monitored validation loss to reduce learning rate when performance plateaued and stop training to avoid overfitting.

Training ran for up to 200 epochs with batch size 8. The final best model achieved a mean IoU of approximately 0.67 on the validation set.As shown in Figure 4, the training and validation loss steadily decreased while the Mean IoU improved significantly, confirming stable convergence and effectiveness of our composite loss function.

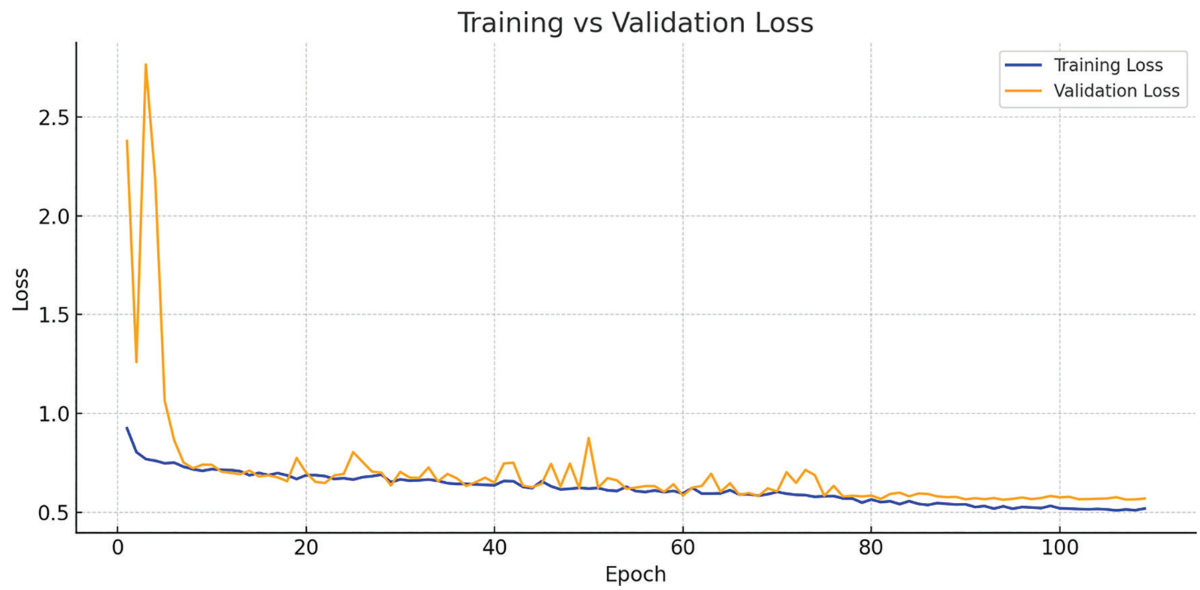

Figure 4: Training vs. Validation Loss

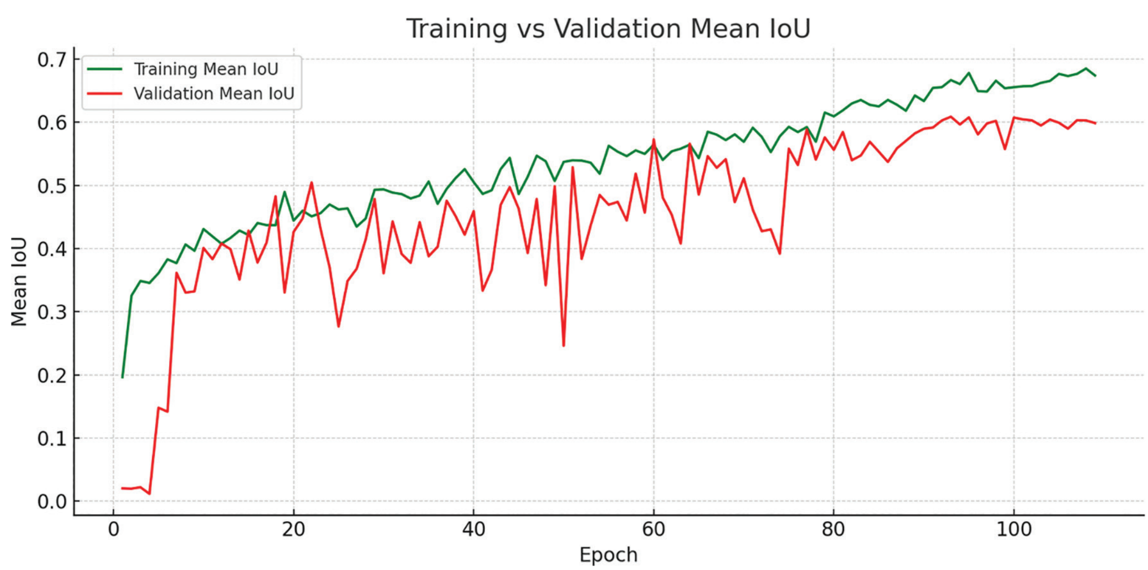

Figure 5: Training vs. Validation Mean IoU

3.5 Postprocessing and Smoothing

Raw model predictions sometimes contained noise and small spurious objects. To refine segmentation masks, morphological postprocessing techniques were applied:

a) Connected component analysis removed small isolated regions below a size threshold.

b) Small holes within regions were filled.

c) Optional median smoothing applied a median filter per class mask to further smooth boundaries.

These steps enhanced visual coherence and improved mask quality, especially in noisy classes like rangeland and barren land.

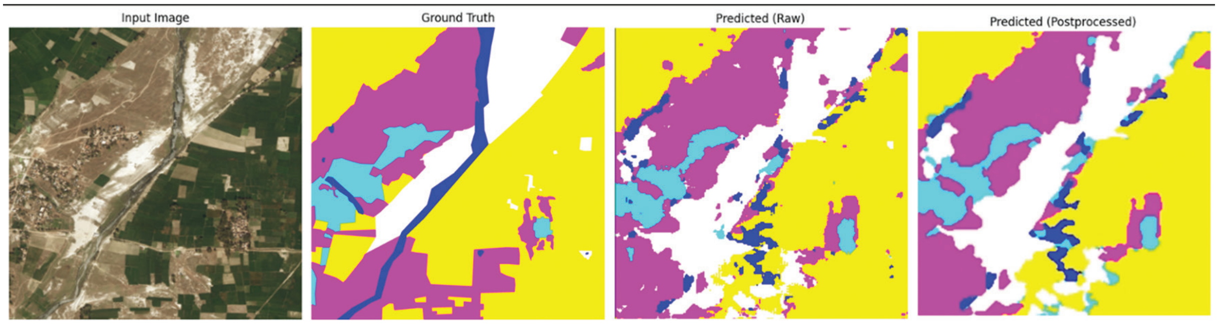

Figure 6: Before VS After Postprocessing image segmentation

3.6 Deployment with Streamlit

The trained U-Net model was deployed using Streamlit for interactive web-based use. The app accepts user-uploaded RGB satellite images (recommended size 224×224), runs model prediction, applies morphological postprocessing, and displays the segmented mask with a color-coded legend.

Users can toggle median smoothing on/off and visualize predictions dynamically. Class colors and names are loaded from a CSV file for flexibility.

The app caching mechanisms optimize performance by loading the model and class colors once per session.

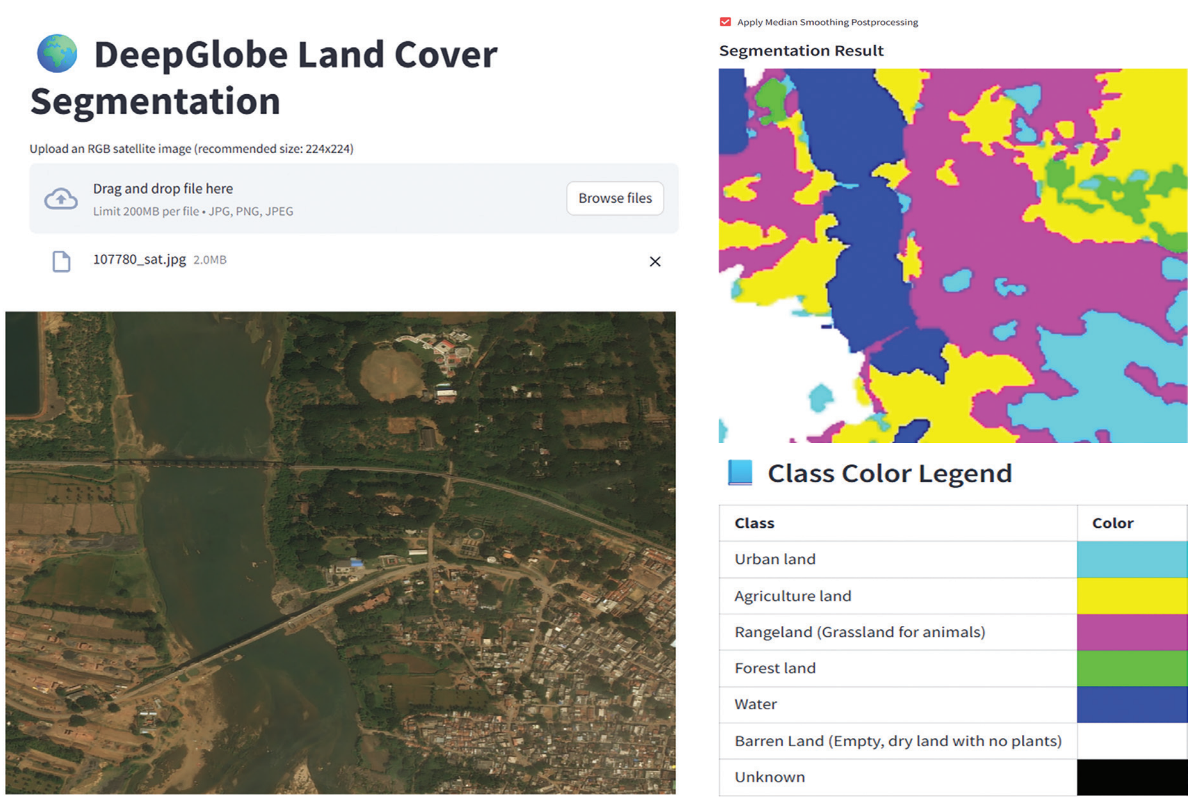

Figure 7: Streamlit User-Interface for DeepGlobe Land Cover Segmentation

3.7 Fine-Tuning for Nepali Satellite Imagery

Currently, efforts are underway to fine-tune the model on Nepali satellite images to improve local applicability. A key challenge remains in collecting a sufficient and well-annotated Nepali dataset to enable transfer learning and domain adaptation.

Ongoing work involves sourcing high-resolution images, annotating land cover classes consistent with DeepGlobe definitions, and experimenting with fine-tuning the pre-trained U-Net model to capture Nepal-specific land features.

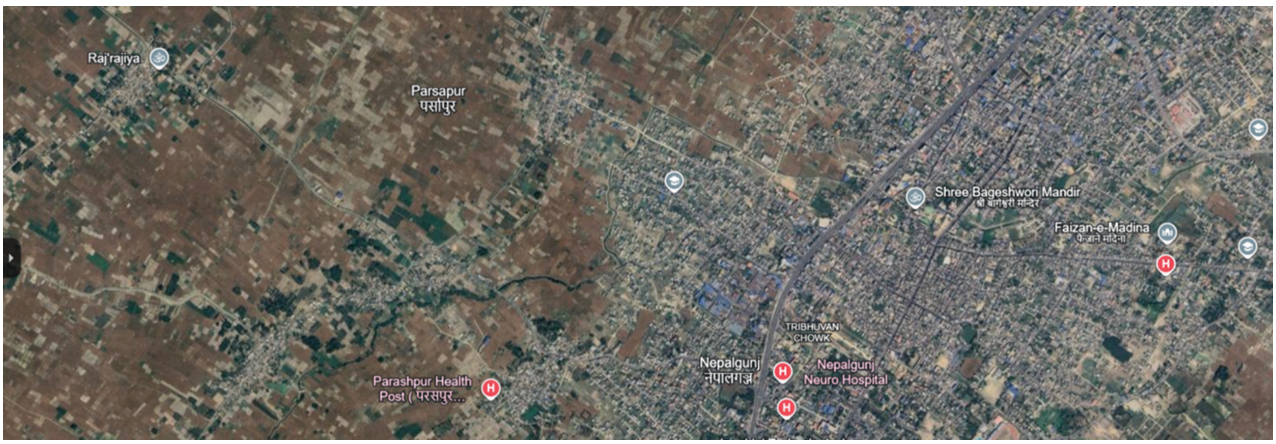

Figure 8: High-resolution satellite image from Nepalgunj, Nepal (© Google, captured via Google Maps Satellite view, July 2025). This image illustrates the heterogeneous mix of urban density and agricultural fields, typical of the Terai region. Such imagery is being considered for local fine-tuning of the segmentation model.

Figure 9: Predicted segmentation map of the Nepalgunj image using the base U-Net model (not fine-tuned).

Although the model was trained on the DeepGlobe dataset, it generalizes to some extent—correctly identifying large urban and agricultural zones. However, the boundary noise and misclassified patches emphasize the need for domain-specific fine-tuning for Nepali geography.

4.Result and Discussions

4.1 Training Performance

The model converged within approximately 110 epochs. The final training loss stabilized around 0.51 with a mean IoU of 0.68 on the training set, while validation loss was around 0.57 with mean IoU 0.60, indicating reasonable generalization.

Per-class IoU revealed strong performance in dominant classes like Urban (0.72), Forest (0.72), and Water (0.66), while smaller or more heterogeneous classes like Rangeland and Barren land had lower IoU (~0.35 and 0.57 respectively). This aligns with known challenges in class imbalance and spectral similarity.

Table 3: IoU Score across individual classes and mean IoU among classes

Class Index	Name	IoU Score
0	Urban Land	0.7187
1	Agriculture Land	0.7170
2	Rangeland	0.3525
3	Forest Land	0.7216
4	Water	0.6582
5	Barren Land	0.5681
6	Unknown	0.9826
___	**Mean IoU**	**0.6741**

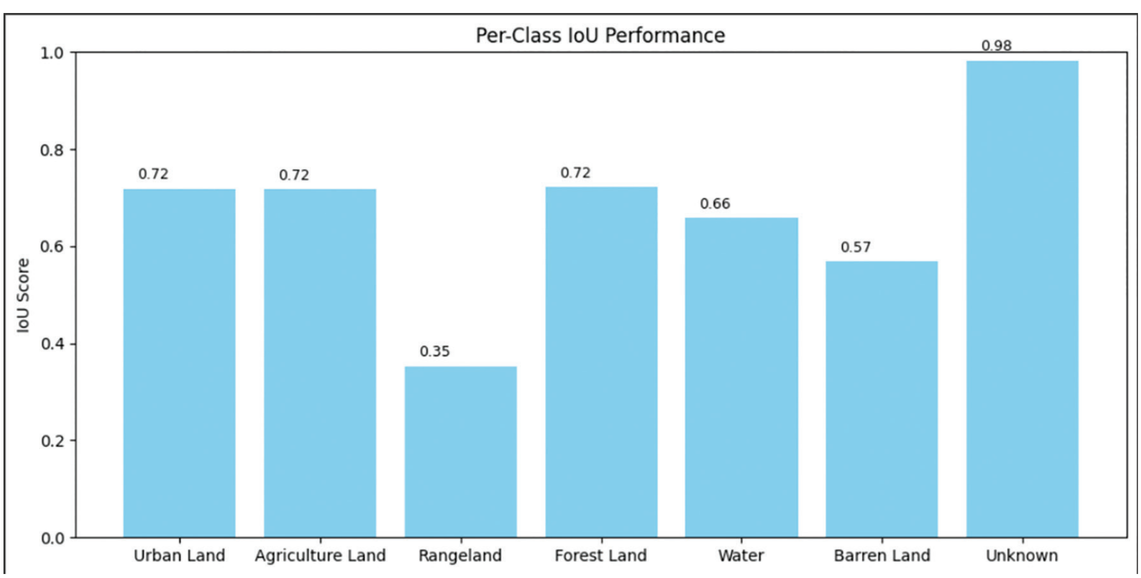

Figure 8: Per-Class IoU Performance showing segmentation accuracy across individual land cover types.

4.2 Qualitative Segmentation Outputs

Visual inspection of randomly selected test images showed the model successfully delineated large homogeneous regions such as urban and forest areas. Postprocessing effectively removed isolated noise and improved boundary smoothness.

The median smoothing option provides an additional degree of noise reduction at the cost of minor boundary softening.

4.3 Deployment Usability

The Streamlit app provides a practical tool for domain users to upload images and get near-real-time segmentation results with an intuitive interface. The color-coded legend aids interpretation.

This deployment demonstrates the pipeline's applicability beyond research, enabling stakeholders to visualize land cover maps without coding knowledge.The application is accessible at the following link:https://4classifying-every-pixel-of-the-planets.streamlit.app

4.4 Challenges in Fine-Tuning for Nepal

Nepali satellite images differ in geographic, climatic, and spectral characteristics from the DeepGlobe dataset's regions. The lack of a large, labeled Nepali dataset limits the ability to fine-tune or train models specifically for Nepal.

Data acquisition is hindered by limited availability of cloud-free images, annotation cost, and heterogeneity of terrain.

Future work will focus on dataset creation through partnerships, semi-supervised annotation, and domain adaptation techniques to overcome these challenges.

5. Conclusions

This study presents an end-to-end pipeline for land cover segmentation using a U-Net model built from scratch and trained on the DeepGlobe dataset with composite loss functions and morphological postprocessing.

The model achieves competitive mean IoU performance and is successfully deployed in a Streamlit app for interactive use. The pipeline demonstrates the feasibility of implementing sophisticated segmentation workflows from first principles and applying them in practical settings.

Ongoing work focuses on adapting and fine-tuning the model for Nepali satellite imagery, with data collection as a critical bottleneck.

This work lays a foundation for further research in region-specific land cover mapping using deep learning and accessible deployment methods.We plan to collaborate with local universities and municipalities to create an open-source labeled satellite dataset for Nepal to promote AI-driven environmental monitoring.

Data Source

The code and data are available at: *https://github.com/yamrajkhadka/project_on_img_seg_using_unet_archi*

6. References

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234–241). Springer. https://doi.org/10.1007/978-3-319-24574-4_28

Balraj, B. (2018). DeepGlobe Land Cover Classification Dataset [Dataset]. Kaggle. https://www.kaggle.com/datasets/balraj98/deepglobe-land-cover-classification-dataset

Chollet, F. (2015). Keras: The Python Deep Learning library. GitHub. https://github.com/fchollet/keras

Abadi, M., et al. (2016). TensorFlow: A System for Large-Scale Machine Learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (pp. 265–283). https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: Fast and Flexible Image Augmentations. Information, 11(2), 125. https://doi.org/10.3390/info11020125

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. In International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1412.6980

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Jorge Cardoso, M. (2017). Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (pp. 240–248). Springer. https://doi.org/10.1007/978-3-319-67558-9_28

Streamlit Inc. (2023). Streamlit: The fastest way to build and share data apps. https://streamlit.io

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., & the scikit-image contributors. (2014). scikit-image: image processing in Python. PeerJ, 2, e453. https://doi.org/10.7717/peerj.453

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2