PROJECT REPORT

AI-Driven Archaeological Site Mapping and Monitoring using Satellite Imagery and Machine Learning

Cherukuri V V M Subhash.

Table of Contents

1.Problem Statement

The core problem addressed in this project is to develop an AI pipeline that helps in:

- Detecting potential ancient structures and ruin regions
- Identifying vegetation and terrain patterns
- Predicting risk of damage due to erosion
- Providing a visual interface for model outputs

This helps improve archaeological monitoring and decision-making using AI automation.

2.Introduction

Archaeological sites are valuable cultural assets but they are often affected by natural degradation, vegetation growth, and terrain erosion. Manual identification and monitoring of such regions using satellite imagery is time-consuming.

This project aims to build an AI-based system that can analyze satellite images to support archaeologists by:

1. Identifying and segmenting ruins, vegetation, and terrain from satellite images using semantic segmentation.
2. Detecting archaeological structures in images using object detection.
3. Predicting erosion risk and erosion severity based on terrain features using machine learning models.
4. Visualizing the outputs on an interactive dashboard for easy interpretation and demonstration.

3.Objectives

The major objectives of this project are:

- Collect satellite images of archaeological regions using Google Earth Pro.
- Create labeled datasets for segmentation and detection tasks.
- Train a segmentation model (U-Net) for classifying pixels into:
    - Background
    - Ruins
    - Vegetation
    - Terrain
- Train an object detection model (YOLOv8) to locate archaeological structures with bounding boxes.
- Build an erosion prediction model using terrain features.
- Evaluate models using standard metrics:

    - IoU, Dice Score (Segmentation)
    - Precision, Recall, mAP (Detection)
    - Accuracy, RMSE, R^2 (Erosion prediction)
- Create an interactive dashboard using Streamlit.

4.Tools and Technologies Used

Category	Tools / Technologies
Data Collection	Google Earth Pro
Annotation Tool	Labelbox
Segmentation Model	U-Net (PyTorch)
Object Detection Model	YOLOv8 (Ultralytics)
Erosion Prediction	Random Forest Classifier + Regressor
Programming Language	Python
Dashboard	Streamlit
Image Format	PNG

5.Dataset Collection and Preparation

5.1 Data Source

The dataset was collected using Google Earth Pro, using satellite screenshots captured from multiple geographical regions containing ruins, vegetation, and terrain.

Project learning

5.2 Dataset Classes (Segmentation)

The segmentation dataset was prepared with the following classes:

Project learning

- Background
- Ruins
- Vegetation
- Terrain

5.3 Dataset Organization

The dataset images were organized into folders based on class-based collection:

Folder	Description
ruins	Images with visible archaeological structures
vegetation	Images with dense forest/tree regions
terrain	Images with open land/background

5.4 Annotation Process (Labelbox)

- A segmentation project was created in Labelbox.
- Polygon-based labeling was performed for selected images.
- Approximately 12–13 images per class were labeled initially.
- Some unclear images were skipped to maintain dataset quality.

5.5 Annotation Export and Mask Generation

- Annotations were exported in NDJSON format from Labelbox.
- Polygon annotations were converted into pixel-wise masks.
- Dataset was split into train/validation sets for model training.

Result:

- Satellite dataset collected .
- Images annotated.
- Masks generated .
- Dataset ready for training .

---

6. Model Implementation, Training and Evaluation

6.1 Model Selection

For semantic segmentation, U-Net was selected because it is effective for pixel-wise classification tasks in medical and satellite imagery.

6.2 Training Setup

Parameter	Value
Model	U-Net
Framework	PyTorch
Optimizer	Adam
Loss Function	Cross Entropy
Hardware	CPU

Parameter	Value
Epochs	10–20

Dataset split used in training:

- 80 images for training
- 20 images for testing

6.3 Evaluation Metrics

Two evaluation metrics were used:

- IoU (Intersection over Union)
- Dice Score

6.4 Segmentation Results

The final segmentation model performance achieved:

- Mean IoU Score: 8.49%
- Mean Dice Score: 7.8%

Class-wise evaluation:

Class	IoU (%)	Dice (%)
Ruins	2.43	3.63
Vegetation	7.85	4.33
Terrain	15.20	15.44
Overall Mean	**8.49**	**7.8**

6.5 Observations

- Ruins IoU was low due to limited labeled data for ruins.
- Vegetation was sometimes confused with terrain due to visual similarity.
- Terrain achieved better segmentation due to majority presence and class dominance.

Result:

- U-Net implemented .
- Model trained .
- IoU & Dice evaluated .
- Results analyzed .

7. Object Detection using YOLO

7.1 Purpose of YOLO in this Project

While U-Net performs pixel-level segmentation, YOLO is used for object detection, where the task is to locate archaeological structures and provide bounding boxes with class labels.

7.2 YOLO Dataset Preparation

- Data source: Google Earth Pro satellite imagery
- A separate dataset was prepared for object detection.
- Only one class was used:
  archaeological_structure
- Roads, buildings, vegetation, and plain land were avoided in labeling.

7.3 YOLO Annotation using Labelbox

Annotation Type	Bounding Boxes
Total labeled images	27
Train images	20
Validation images	7

Labeling rules followed:

- Tight bounding box on ruins/structures
- Avoid overlapping boxes
- Skip images without clear structures

7.4 YOLO Training Configuration

Parameter	Value
Model	YOLOv8 Nano
Epochs	50
Image size	640
Framework	Ultralytics YOLO
Hardware	CPU

7.5 YOLO Model Results

Model evaluation results obtained:

Metric	Value
Precision	97.4%
Recall	28.6%
mAP@0.5	34.3%
mAP@0.5:0.95	16.1%

7.6 YOLO Result Interpretation

- The model achieved high precision, indicating fewer false detections.
- Recall and mAP were moderate due to limited dataset size.
- The results confirm YOLO can successfully detect archaeological structures from satellite imagery.

Final Result of YOLO Model:
The YOLOv8 object detection model successfully detected archaeological structures from satellite imagery with high precision. However, due to the limited dataset size, recall and mAP were moderate.

8.Terrain Erosion Prediction

8.1 Objective

The objective of Milestone 3 is to predict:

- Erosion-prone vs stable areas (classification)
- Erosion severity score (regression)

using terrain features like slope, elevation, vegetation, and terrain type.

8.2 Dataset Preparation

The erosion dataset was prepared using:

- Google Earth Pro observations
- Manual and logical labeling
- Feature engineering

Features included:

- Slope
- Elevation
- Vegetation (Low/Medium/High)

- Terrain_Type
- Erosion_Label (0/1)
- Erosion_Score (0–1)

8.3 Models Used

Model Type	Algorithm	Output
Classification	Random Forest Classifier	Stable vs Erosion-prone
Regression	Random Forest Regressor	Erosion severity score

8.4 Results

Regression model achieved:

- **RMSE:** 0.064
- **R^2 Score:** 0.945

This indicates strong correlation between terrain features and erosion severity.

Result:
Terrain erosion prediction using classification and regression was successfully implemented and evaluated.

9.Evaluation, Visualization and Final Reporting

9.1 Objective

The objective of Milestone 4 is to:

- Evaluate all models
- Integrate outputs into an interactive dashboard
- Prepare final report and demo submission

9.2 Dashboard Implementation

A Streamlit dashboard was created to visualize:

YOLO Object Detection

- Original image selection
- Predicted image display
- Detection count
- Confidence values
- YOLO training metrics (mAP, precision, recall)

Terrain Erosion Prediction

- Dataset visualization
- Filters by terrain and vegetation
- Summary cards and charts for erosion risk analysis

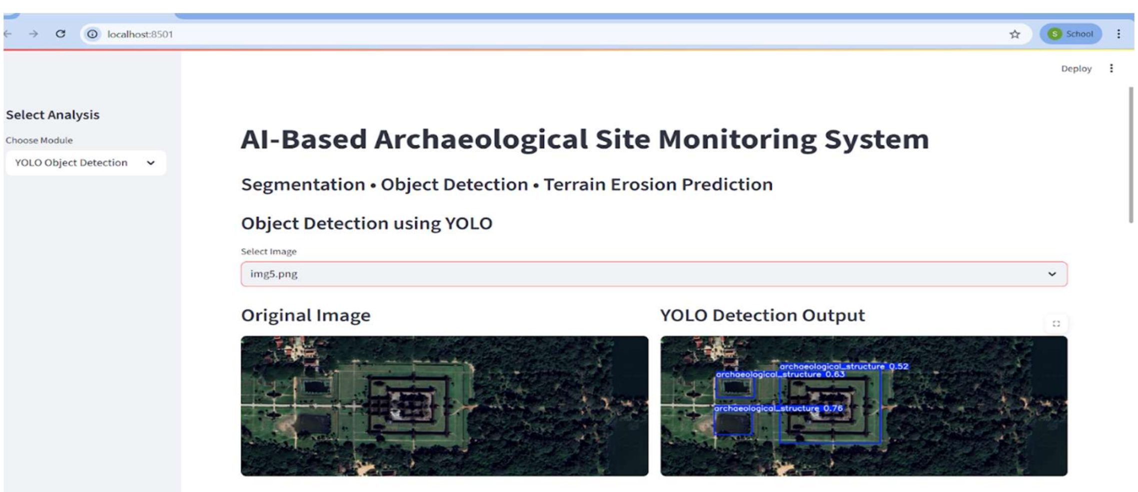
Result:
A working dashboard was successfully built for project visualization and demonstration.

---

10. Final Model Performance Summary

Module	Task	Metric	Result
U-Net	Segmentation	Mean IoU	8.49%
U-Net	Segmentation	Mean Dice	7.8%
YOLOv8	Object Detection	Precision	97.4%
YOLOv8	Object Detection	Recall	28.6%
YOLOv8	Object Detection	mAP@0.5	34.3%
YOLOv8	Object Detection	mAP@0.5:0.95	16.1%
RF Classifier	Erosion Prediction	Accuracy	High
RF Regressor	Erosion Severity	RMSE	0.064
RF Regressor	Erosion Severity	R^2 Score	0.945

OUTPUT:



localhost:8501

School

Deploy

Select Analysis

Choose Module

YOLO Object Detection

Detection Summary (Count + Confidence)

Total Objects Detected: 3

	Class_ID	Class_Name	Confidence
0	0	Archaeological_Structure	0.7616
1	0	Archaeological_Structure	0.6323
2	0	Archaeological_Structure	0.5225

- **Average Confidence:** 0.639
- **Max Confidence:** 0.762
- **Min Confidence:** 0.522

YOLO Training Metrics (results.csv)

	epoch	time	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)	metrics/mAP50-95(B)	val/box_loss	val/cl
45	46	1013.04	0.9723	2.7033	1.1025	0.9793	0.2857	0.3384	0.1602	2.2448	2
46	47	1031.57	1.1176	2.408	1.1297	0.9777	0.2857	0.3296	0.1537	2.2703	2
47	48	1051.45	0.9642	3.2103	1.1517	0.9753	0.2857	0.3301	0.1538	2.2302	2
48	49	1071.45	1.0871	2.5148	1.1147	0.9741	0.2857	0.3443	0.1587	2.2485	2
49	50	1091.41	1.1099	2.2034	1.0691	0.9743	0.2857	0.3434	0.1613	2.225	2

Final Epoch Summary

- Final Epoch: 50
- metrics/mAP50(B): 0.343
- metrics/mAP50-95(B): 0.161
- metrics/precision(B): 0.974
- metrics/recall(B): 0.286

Deploy

Select Analysis

Choose Module

Terrain Erosion Predi...

AI-Based Archaeological Site Monitoring System

Segmentation • Object Detection • Terrain Erosion Prediction

Terrain Erosion Prediction Dashboard

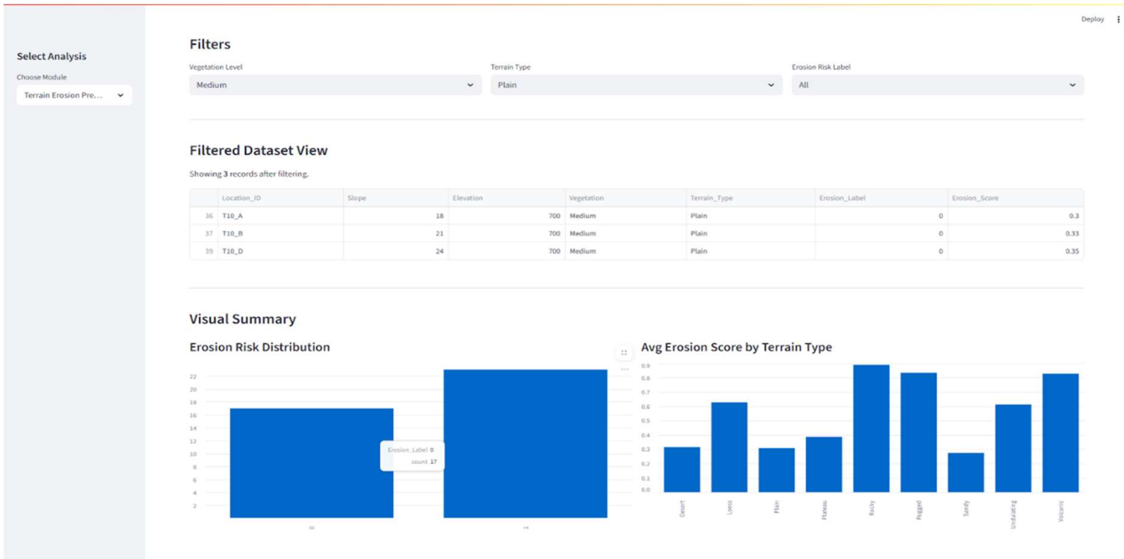
Dataset Overview

Total Records	Total Columns	Stable Areas (0)	Erosion-Prone Areas (1)
40	7	17	23

Average Erosion Severity Score: **0.597**

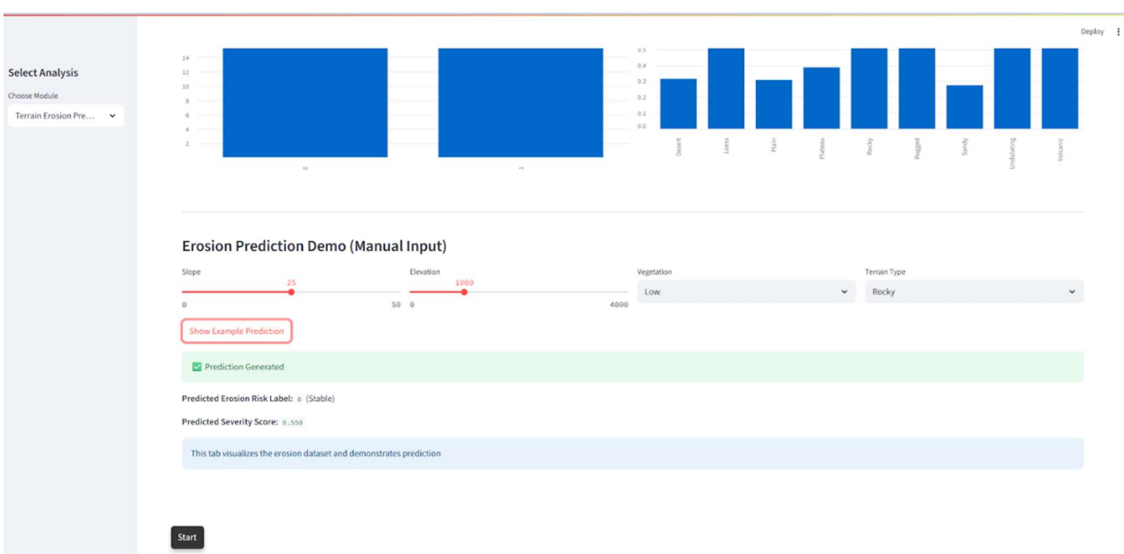
Filters

Vegetation Level	Terrain Type	Erosion Risk Label
All	All	All



Final Epoch Summary

- ✅ Final Epoch: `50`
- ✅ metrics/mAP50(B): 0.343
- ✅ metrics/mAP50-95(B): 0.161
- ✅ metrics/precision(B): 0.974
- ✅ metrics/recall(B): 0.286



11. Challenges Faced

- Limited number of labeled satellite images for ruins.
- Visual confusion between vegetation and terrain regions.
- Low segmentation performance due to dataset size and CPU training constraints.
- YOLO recall was low due to fewer labeled detection samples.
- Dashboard integration required path handling and prediction output management.

---

12. Conclusion

This project successfully demonstrates an end-to-end AI workflow for archaeological site monitoring using satellite imagery. The system includes:

- Semantic segmentation using U-Net for identifying ruins, vegetation and terrain.
- Object detection using YOLOv8 for detecting archaeological structures.
- Terrain erosion prediction using Random Forest models for stable/prone classification and severity estimation.
- Interactive visualization using Streamlit dashboard for final demonstration.

The results prove the feasibility of applying AI techniques in archaeology for improved monitoring and mapping.

---

13. Future Improvements

If more time and resources are available, the project can be improved by:

- Increasing dataset size and improving annotation quality.
- Using GPU-based training for better model performance.
- Trying advanced segmentation models like DeepLabV3+.
- Adding multi-class YOLO detection (walls, pillars, temples).
- Integrating map overlays for real geospatial visualization.

---

14. References

- Google Earth Pro (satellite imagery collection)
- Labelbox (annotation tool)
- PyTorch (deep learning framework)
- Ultralytics YOLOv8 documentation
- Streamlit documentation