AI-DRIVEN SYSTEM FOR OIL SPILL IDENTIFICATION AND MONITORING

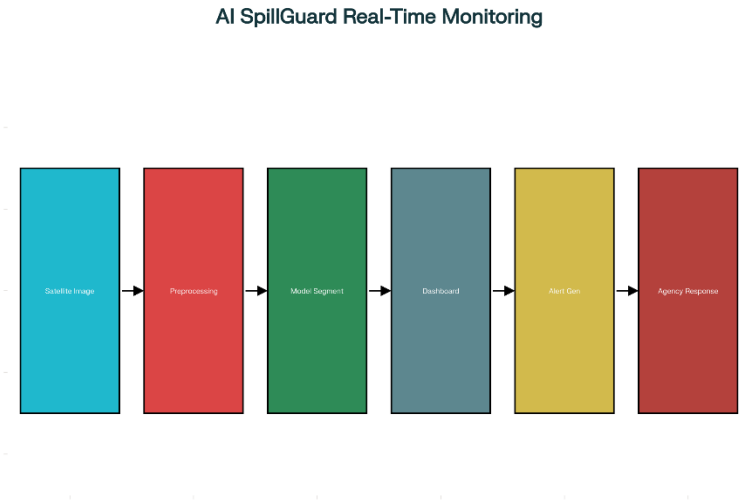
BY: JEEVIETHA

1. Introduction

Oil spills cause significant environmental and economic damage by contaminating marine ecosystems and affecting coastal economies. Conventional detection methods, such as manual satellite image inspection and physical patrolling, are slow and labor-intensive, often delaying critical interventions.

The AI SpillGuard project builds an automated, AI-powered system leveraging satellite imagery and deep learning for accurate, real-time oil spill detection and segmentation. Using the U-Net convolutional network architecture, the system performs semantic segmentation of satellite images, producing pixel-wise mask predictions that highlight oil-contaminated regions.

This documentation presents the technical details of the system pipeline, model design, training and evaluation methodologies, visualization strategies, and deployment via a user-interactive Streamlit application.



2. Dataset Acquisition and Preprocessing

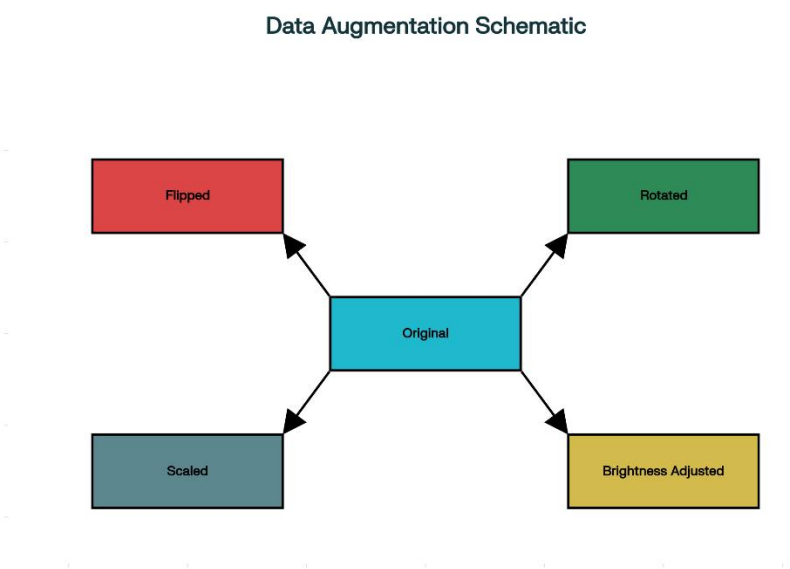
The project utilizes satellite images from publicly available sources such as Sentinel-1 Synthetic Aperture Radar (SAR), MODIS, and NOAA. Ground truth annotations specifying oil spill regions are gathered from environmental datasets or manually labeled to form segmentation masks.

The dataset is organized into structured folders for training, validation, and testing images and masks.

The preprocessing pipeline handles:

- Image resizing to a uniform resolution of 256×256 pixels to fit the U-Net input requirements.
- Normalization scaling pixel intensities to.image.jpg
- Speckle noise reduction tailored for SAR images to remove sensor artifacts.
- Data augmentation through flips, rotations, zooms, and brightness adjustments to increase model robustness.

These essential preprocessing steps prepare high-quality input data, essential for effective deep learning model performance.



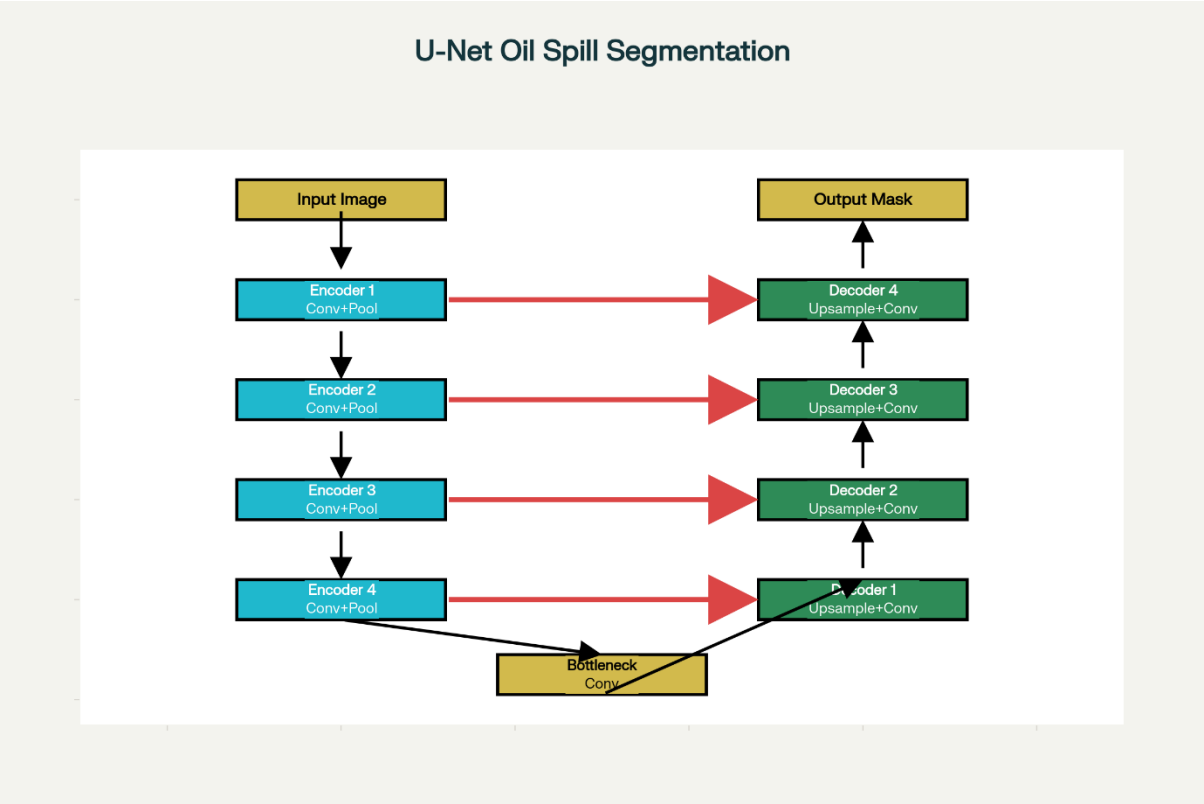
3. Model Architecture: U-Net

AI SpillGuard adopts the U-Net architecture, widely recognized for biomedical and satellite image segmentation.

- The **encoder** compresses information via convolutional and max-pooling layers to learn multiple feature hierarchies.
- The **decoder** upsamples these encoded features, reconstructing spatial details and precise segmentation masks.
- Skip connections allow low-level features to bypass deeper layers, improving localization.

The network outputs a pixel-wise probability map indicating the presence or absence of oil spill contamination.

Custom loss functions combining Binary Cross-Entropy and Dice Loss aid tackling class imbalance and encourage accurate segmentation.



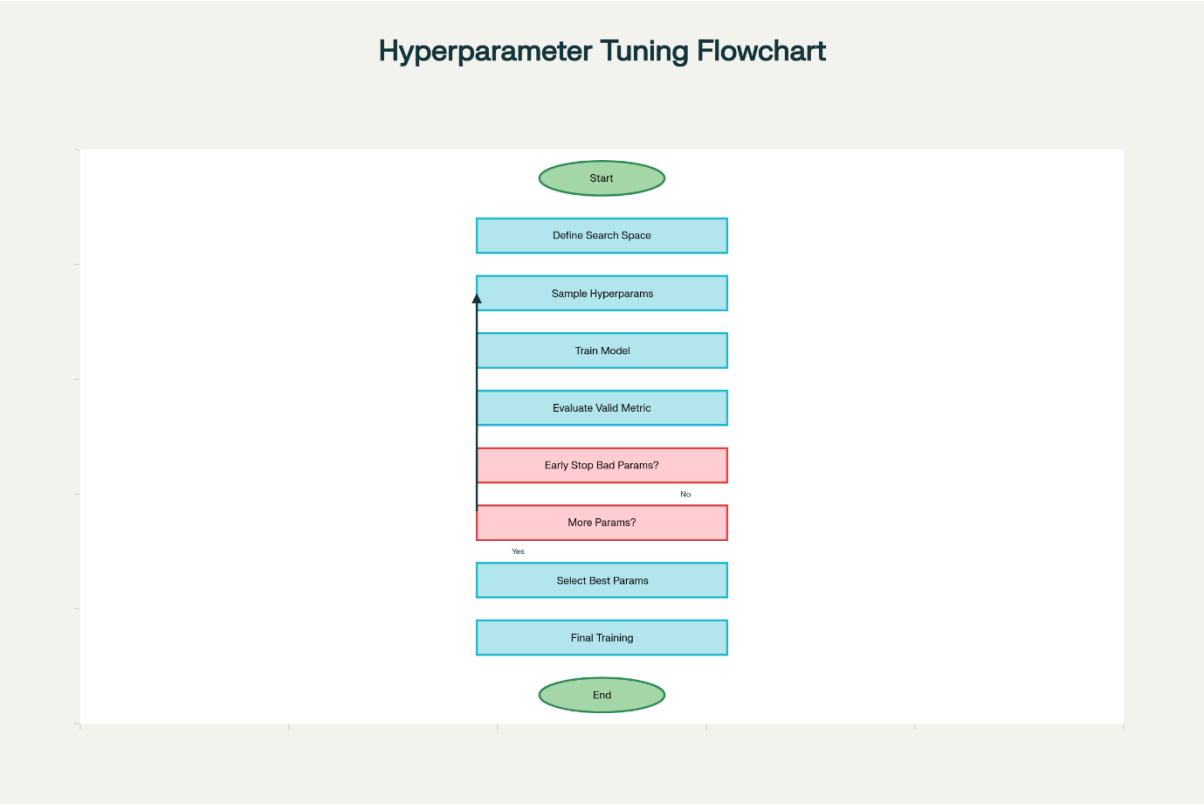
4. Training and Evaluation Framework

The model is trained on the preprocessed datasets using the Adam optimizer, with batch sizes, learning rates, and other hyperparameters tuned automatically through Keras Tuner's Hyperband algorithm.

Metrics for performance assessment:

- **Pixel Accuracy:** Overall pixel-wise classification ratio.
- **Dice Coefficient:** Reflects the spatial overlap between predicted and ground truth masks.
- **Intersection over Union (IoU):** Measures commonality between predicted and actual spill regions.
- **Precision & Recall:** Evaluate false positives/negatives respectively.

Early stopping based on validation performance, weight checkpointing, and learning rate scheduling help prevent overfitting and speed convergence.

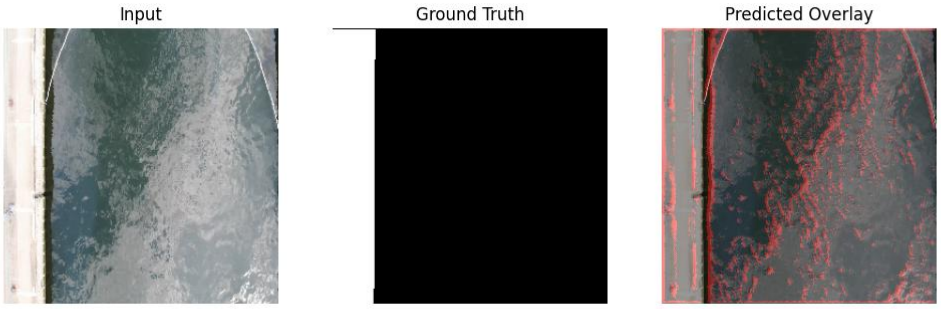


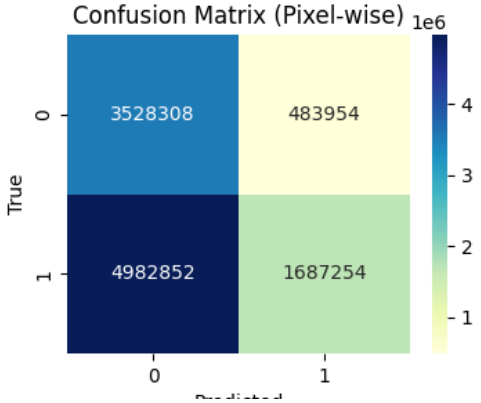
5. Visualization and Results

The system supports multiple visualization techniques for qualitative and quantitative evaluation:

- Side-by-side comparisons of satellite images, ground truth masks, and predicted segmentation masks.
- Mask overlays on input images using red color highlights with adjustable transparency.
- Confusion matrices depicting pixel-wise prediction results with heatmaps sized to exactly match input and mask images.
- Plots of training and validation loss histories for model diagnostics.

Result images, masks, overlays, and plots are saved systematically to output directories for review and reporting.



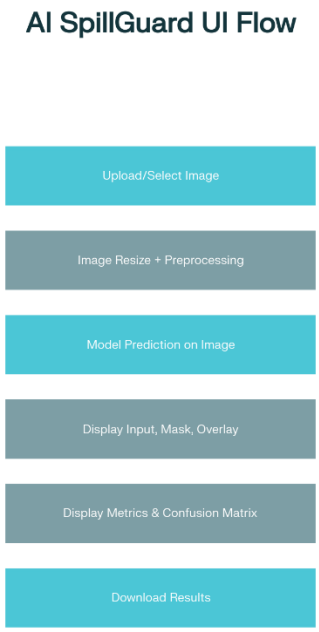


6. Deployment: Interactive Streamlit Application

A dedicated Streamlit-based web app enables stakeholders to:

- Upload new satellite images or select from test datasets.
- Input images are resized to 256×256 internally before prediction.
- Display input image, predicted binary mask, and overlay—all rendered at consistent 256×256 dimensions for visual clarity.
- Show computed metrics (Dice, IoU, pixel accuracy).
- Present confusion matrix plots at fixed size for image-to-mask comparison.
- Downloadable results (input, mask, overlay) for offline use.
- Sidebar alerts and progress bars enhance user experience.

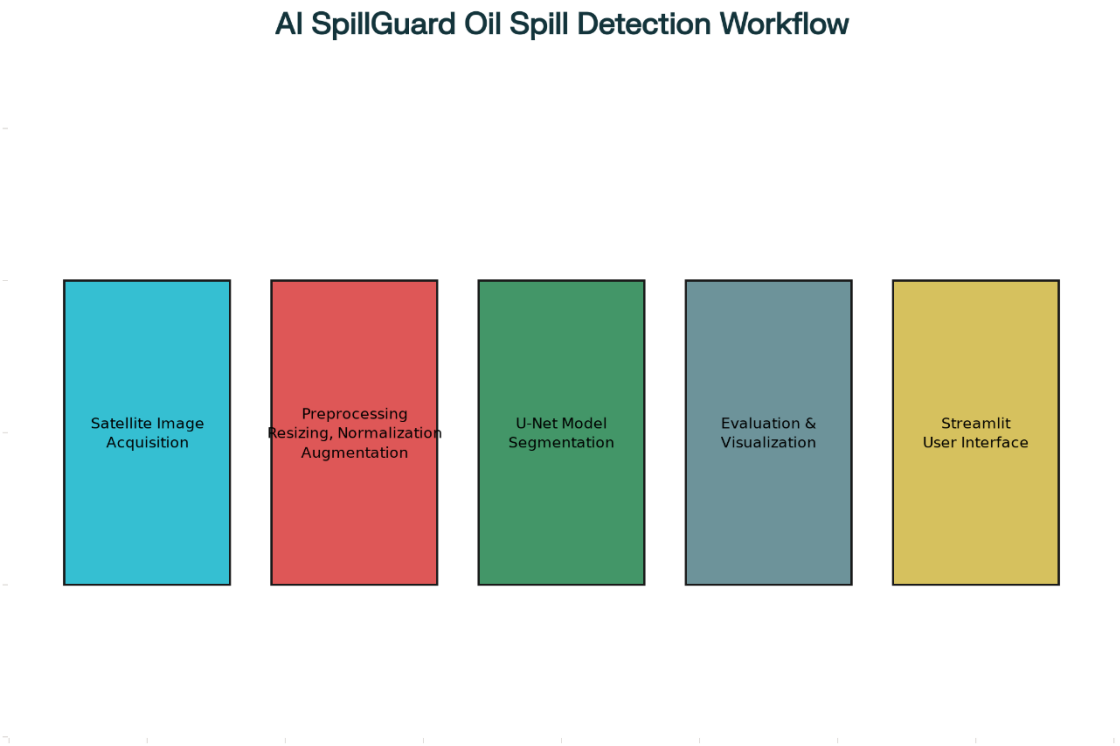
This app demonstrates a user-friendly interface that supports rapid and transparent oil spill monitoring, adaptable to real-time workflows.



7. Modules Included in Codebase

The codebase comprises:

- **Dataset Preparation:** setup_and_load.py handles data parsing and preprocessing.
- **Model Development and Training:** model_define_train.py builds the U-Net, runs training with checkpointing.
- **Hyperparameter Optimization:** hyperband_tuner.py automates tuning of learning parameters.
- **Evaluation & Visualization:** evaluate_and_visualize.py for metric calculation, overlay creation, plotting.
- **Streamlit App:** streamlit_app_launcher.py integrates front-end with backend inference, user inputs, and visualization.
- **Artifacts:** Saved model architecture (unet_architecture.json), trained weights (unet_best_weights.weights.h5), training history (unet_history.json), and visualization plots.



8. Conclusion and Future Work

AI SpillGuard provides an end-to-end solution for automatic, accurate oil spill identification from satellite images. This has critical implications for early environmental impact prevention and efficient crisis management.

Potential future enhancements include real-time alert systems, scalable API deployment, integration with additional satellite platforms, and expansion to other environmental monitoring tasks.