Shoreline Monitoring Using a Network of Webcams

Mona Zhao[1], Joe Long[2], Borja Reguero[3], David Gutierrez[3], Greg Dusek[4], Jon Warrick[5], Alex Pang[1]

[1]Department of Computer Science and Engineering, UCSC, [2]Department of Earth and Ocean Sciences, NC Wilmington, [3]Center for Coastal Climate Resilience, UCSC, [4]NOAA, [5]USGS

Objectives

This project aims to investigate the use of machine learning methods for shoreline extraction, offering the potential to analyze shoreline changes across different timescales—from swash zone dynamics to tidal, seasonal change and longer-term events.

Background

Ocean observing networks, such as those managed by webcoos.org, NWLON, and USGS-sponsored platforms, offer valuable tools for shoreline monitoring. Traditional methods, like using brightest pixel images, help determine beach runups but often fall short in capturing the complexity of shoreline dynamics. Many communities need cost-effective ways to track shoreline changes and flooding. Regular webcams offer a valuable but underused resource. Our machine learning algorithm fills this gap, providing an efficient alternative to traditional methods.

Shoreline Definition

In this work, we define the shoreline as the instantaneous interface between water and non-water (e.g., beach, cliff). Unlike the brightest pixel method, which requires video over a wave cycle, our approach allows real-time shoreline extraction from a single frame.

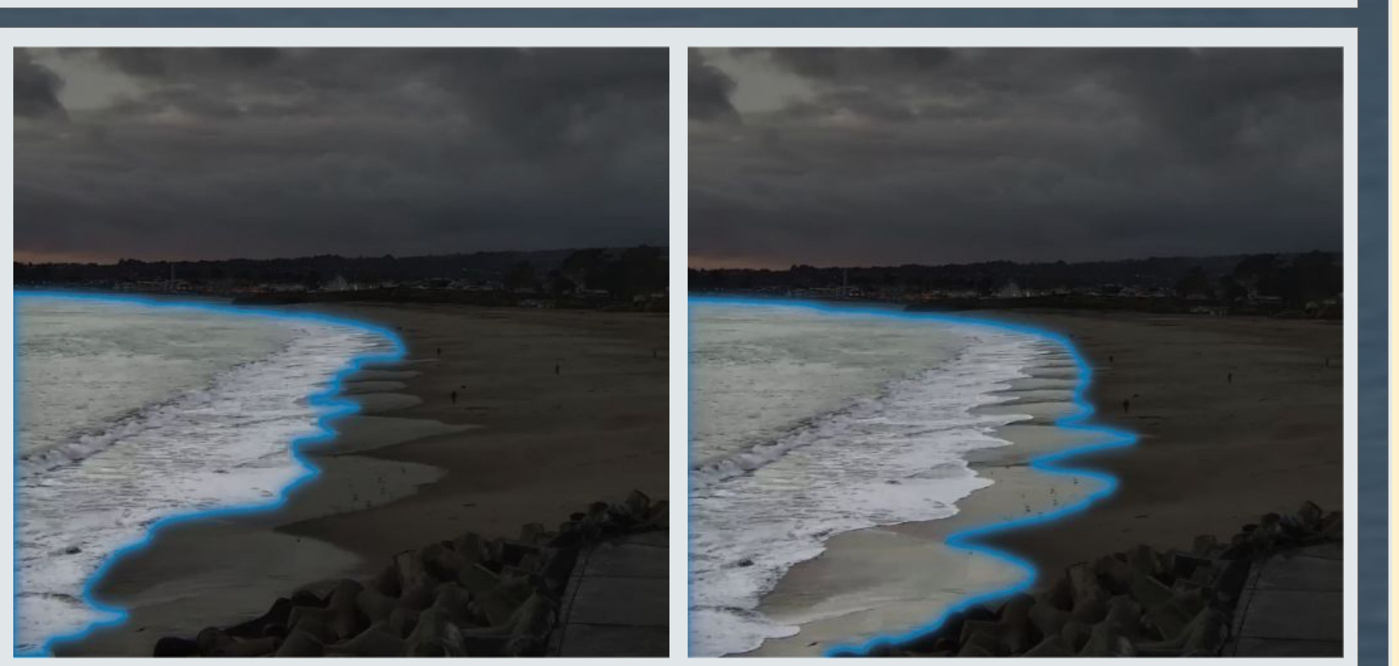


Figure 1: Left: instantaneous shoreline. Right: beach runup

Challenges

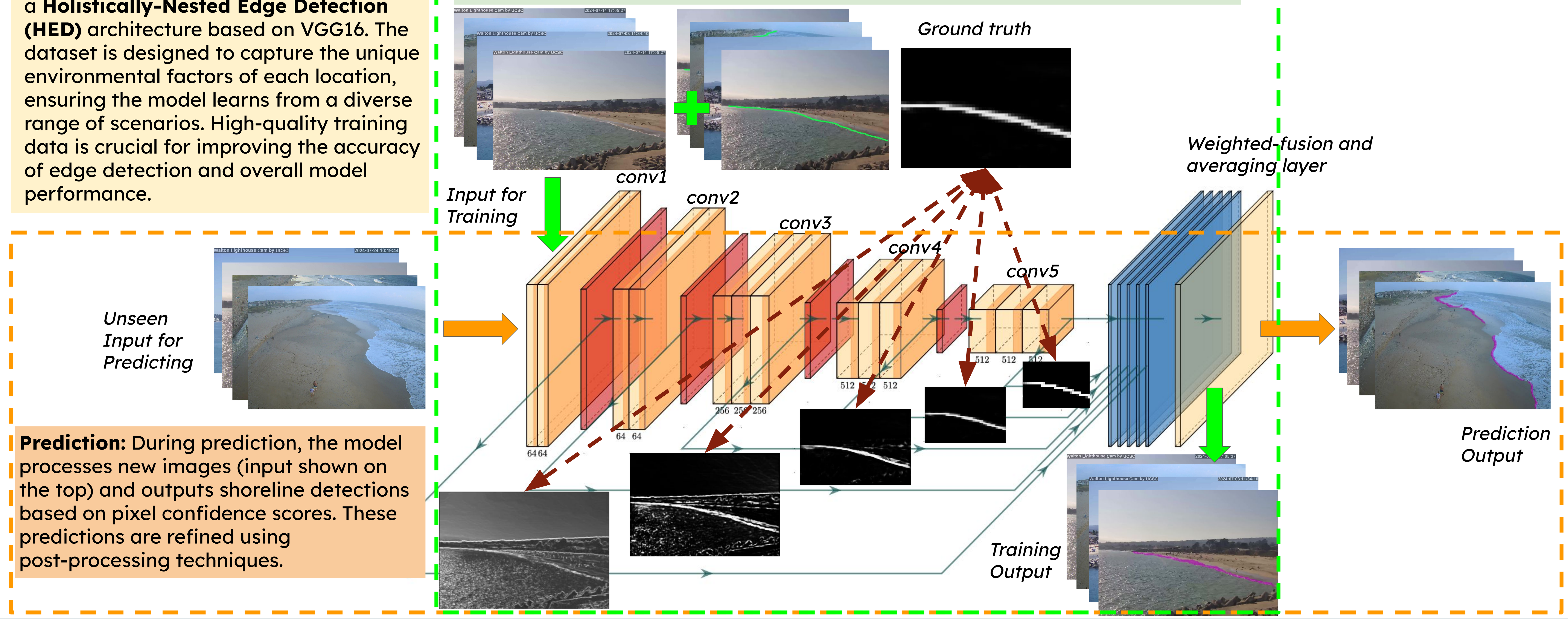
Several factors complicate accurate shoreline monitoring using webcams:

- **Environment Condition**: Stormy weather, fog, and snow affect shoreline appearance.
- **Time of Day**: Different lighting conditions and glare impact visibility.
- **Lens Condition**: Dirt, moisture, and smudges on the lens can obscure the view.

Methodology and Results

We propose a systematic approach to creating a training dataset optimized for **different locations and conditions**, using a **Holistically-Nested Edge Detection (HED)** architecture based on VGG16. The dataset is designed to capture the unique environmental factors of each location, ensuring the model learns from a diverse range of scenarios. High-quality training data is crucial for improving the accuracy of edge detection and overall model performance.

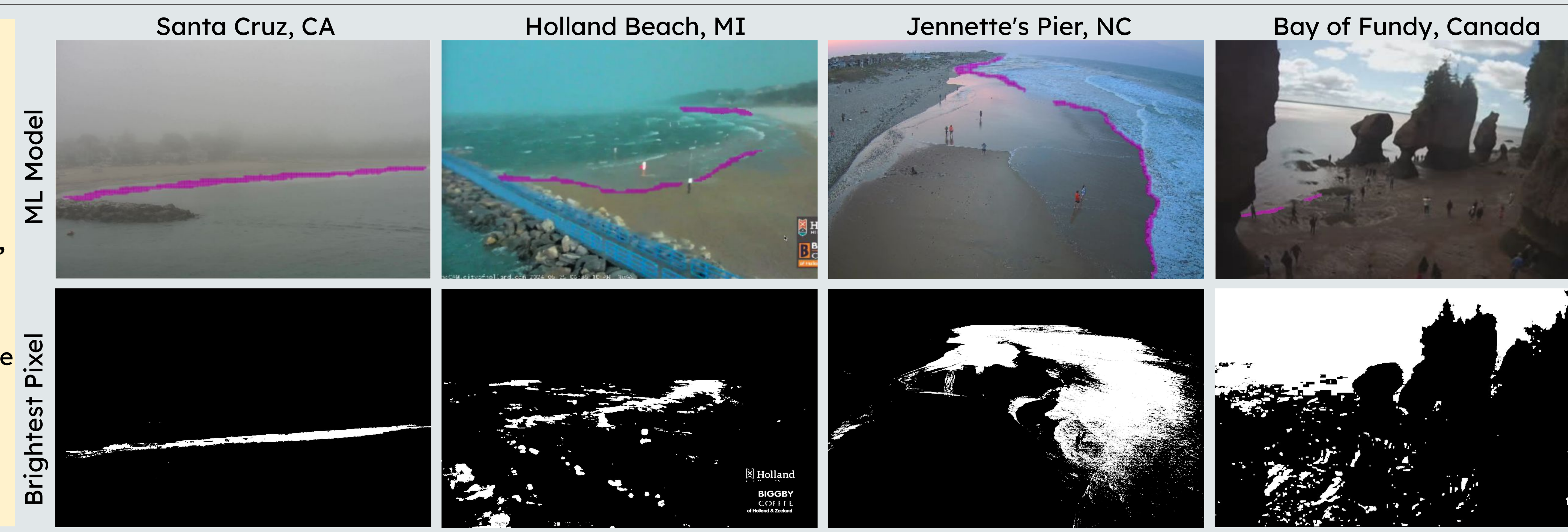
Training: During training, batches of labeled images (example shown below with green shoreline labels) are used. Ground truth edge maps (example in binary) are compared to predictions from each DSN using a loss function.



Prediction: During prediction, the model processes new images (input shown on the top) and outputs shoreline detections based on pixel confidence scores. These predictions are refined using post-processing techniques.

Figure 2: Diagram of the ML model pipeline, illustrating the preprocessing, multi-scale edge detection, and postprocessing steps, with visualized examples at each stage.

This comparison shows our detection method alongside the traditional brightest pixel method, applied to four different sites: **California, Michigan, North Carolina, and Canada.** Each site presents unique conditions such as **cloudiness, storms, water puddles, dirty camera lenses,** and other obstacles. Our results demonstrate that the ML model delivers more consistent performance across these diverse scenarios. We can generate time series data of water marks along the beach profile, providing detailed measurements that help track shoreline changes over time.



Conclusion

- Our machine learning approach enables real-time, frame-by-frame shoreline extraction, even under challenging conditions.
- Regular webcams and location-specific training datasets provide a cost-effective and scalable solution for shoreline monitoring.
- The model can generate time series data of water marks, offering detailed insights into shoreline changes over various timescales.

References:
1. Rogers, M. S. J., Bithell, M., Brooks, S. M., & Spencer, T. (2021). VEdge_Detector: automated coastal vegetation edge detection using a convolutional neural network. International Journal of Remote Sensing. https://doi.org/10.1080/01431161.2021.1897185
2. Simmons, J. (n.d.). Simmonsja/CNN-shoreline-detect: Shoreline detection on oblique images of beaches using a hed CNN approach. GitHub. https://github.com/simmonsja/cnn-shoreline-detect
3. Using video imagery to study coastal change: Whidbey Island. Using Video Imagery to Study Coastal Change: Whidbey Island | U.S. Geological Survey. (n.d.). https://www.usgs.gov/centers/pacific-coastal-and-marine-science-center/science/using-video-imagery-study-coastal-1
4. HarisIqbal88. (n.d.). Harisiqbal88/PlotNeuralNet: Latex Code for making Neural Networks diagrams. GitHub. https://github.com/HarisIqbal88/PlotNeuralNet?tab=readme-ov-file

More Results

Connect with Me!