Fast Ocean Front Detection Using Deep Learning Edge Detection Models

Violet Felt \textsuperscript{1}, Shreeyam Kacker \textsuperscript{1}, Joe Kusters \textsuperscript{1}, John Pendergast \textsuperscript{1}, and Kerri Cahoy \textsuperscript{1}

\textsuperscript{1}Affiliation not available

October 30, 2023

Abstract

Small-scale ocean fronts play a significant role in absorbing the excess heat and CO2 generated by climate change, yet their dynamics are not well understood. Existing in-situ and remote sensing measurements of the ocean have inadequate spatial and temporal coverage to map small-scale ocean fronts globally. Additionally, conventional algorithms to generate ocean front maps are computationally intensive and require data with long lead times. We propose machine learning (ML) models to detect temperature and chlorophyll ocean fronts from unprocessed and radiometrically uncorrected satellite imagery by transfer learning from existing models for edge detection. We use two separate datasets: one based on conventional approaches to ocean front detection, and a second based on human annotated ground truth\textsuperscript{1}. The deep learning front detection approach significantly reduces the resources and overall lead times needed for detecting ocean fronts. The deep learning models are developed with resource-constrained edge compute platforms like CubeSats in mind, as such platforms can address the spatial and temporal coverage challenges. The highest performing models achieve accuracies of 96\% and make predictions in milliseconds using unoptimized desktop CPUs and using less than 100 MB of storage; these capabilities are well-suited for CubeSat deployment.
Fast Ocean Front Detection Using Deep Learning Edge Detection Models

Violet Felt, Shreeyam Kacker, Joe Kusters, John Pendergrast, Kerri Cahoy, Member, IEEE

Abstract—Small-scale ocean fronts play a significant role in absorbing the excess heat and CO2 generated by climate change, yet their dynamics are not well understood. Existing in-situ and remote sensing measurements of the ocean have inadequate spatial and temporal coverage to map small-scale ocean fronts globally. Additionally, conventional algorithms to generate ocean front maps are computationally intensive and require data with long lead times. We propose machine learning (ML) models to detect temperature and chlorophyll ocean fronts from unprocessed and radiometrically uncorrected satellite imagery by transfer learning from existing models for edge detection. We use two separate datasets: one based on conventional approaches to ocean front detection, and a second based on human annotated ground truth. The deep learning front detection approach significantly reduces the resources and overall lead times needed for detecting ocean fronts. The deep learning models are developed with resource-constrained edge compute platforms like CubeSats in mind, as such platforms can address the spatial and temporal coverage challenges. The highest performing models achieve accuracies of 96% and make predictions in milliseconds using unoptimized desktop CPUs and using less than 100 MB of storage; these capabilities are well-suited for CubeSat deployment.

Index Terms—Deep learning, machine learning, ocean front, edge detection, Landsat, CubeSat.

I. INTRODUCTION

Ocean fronts are boundaries with strong gradients in the water properties, such as temperature, salinity, nutrients, and biological content. They are formed primarily due to the mixing of bodies of water and exist in a wide range of spatial and temporal scales. For example, the smallest ocean fronts can be meters long and last for days, whereas the largest fronts span thousands of kilometers and last for millions of years.

Ocean fronts play a vital role in the ecology of marine life, and changing ocean fronts are important indicators of climate change. The ocean absorbs much of the excess heat and carbon dioxide generated by fossil fuel emissions. The exact dynamics of this heat and carbon dioxide emission are yet to be fully understood but may be influenced by small-scale ocean fronts. Recent ocean front tracking indicates irregular and erratic activity, potentially caused by climate change, motivating the use of improved lower latency monitoring.

While ocean fronts can be extracted from a wide variety of properties, the two most commonly tracked types of ocean fronts are sea surface temperature (SST) and chlorophyll-a concentration (CHL) fronts. Traditional in-situ methods use buoys, gliders, and science cruises to track these properties. Fronto hand-drawn over extracted data in areas of sharp gradients.

Remote sensing has allowed for wide-spread monitoring of ocean fronts. For such large areas, ocean fronts are no longer hand-drawn, but instead, algorithms such as the Cayula-Cornillon algorithm (CCA) or the Belkin-O-Reilly algorithm (BOA) are used. These algorithms are computationally intensive and can rely on radiometrically calibrated data from multiple sources. In this work, we propose repurposing deep learning edge detection models to detect ocean fronts directly on L1 radiometrically uncalibrated data, allowing for fast, low-latency monitoring of ocean fronts.

Recent developments in commercial electronics components have increased CubeSats’ processing power and efficiency to the point where complex computations such as those required by some algorithms and even deep learning models can be completed on-orbit. For example, Nvidia and Xilinx are making parallel processing available in embedded forms that can be used onboard small satellites like CubeSats. Fast ocean front detection can then be performed directly on edge devices for global ocean front monitoring. The concept of operations for this involves processing the data on-orbit and using it as a kind of always-on monitoring system, only selecting the most valuable data for downlink. A diagram of this concept is shown in Fig. 1. This work is targeted towards MIT’s BeaverCube-2 mission, but the concepts and models can apply to any Earth-observing satellite.

All authors are affiliated with the Space Telecommunications, Astronomy and Radiation (STAR) Laboratory at the Massachusetts Institute of Technology in Cambridge, Massachusetts. Please address correspondence to Violet Felt (violetcfelt@gmail.com) or Shreeyam Kacker (shreeyam@mit.edu).

Available at https://github.com/MIT-STARLab/deep-learning-ocean-front-detection
with onboard compute capability.

Remote sensing data and algorithms have significantly lowered barriers to tracking ocean fronts. However, there are still issues in the spatial and temporal coverage of ocean fronts, especially small-scale ocean fronts. Large satellite missions often have poor spatial resolution for detecting small-scale ocean fronts or poor temporal resolution because of long revisit times [11]. Many satellites do not spend resources on ocean imaging, instead focusing on land mass imaging. Additionally, the time between when a satellite image is taken and when the ocean fronts in the image are identified can be weeks due to many intermediate processing levels and incorporating data from other sources [12]. This lag prevents real-time tracking of ocean fronts, important for fishermen and scientists alike.

II. BACKGROUND
A. CHL and SST Retrieval from Remote Sensing

Remote sensing offers a way to track ocean fronts globally. There are multiple active satellite missions that can be used to track both CHL and SST fronts, as shown in Table [1]. Primarily these missions have bands around 440 nm to measure CHL and thermal infrared bands between 8–14 μm to measure SST. This work utilizes Landsat 8 data and focuses on ocean fronts in coastal waters.

NASA uses established algorithms to calculate chlorophyll-a concentration and sea surface temperature from surface reflectance data (Landsat Level 2 data) [13–15]. A different set of coefficients is used for each satellite instrument.

CHL concentration is retrieved using a fourth-order polynomial relationship between Landsat Bands 1 (Coastal Aerosol), 2 (Blue), 3 (Green), given as:

\[
\log_{10}(\text{CHL}) = a_0 + \sum_{i=1}^{4} a_i \left( \log_{10} \left( \frac{\lambda_{\text{blue}}}{\lambda_{\text{green}}} \right) \right)^i
\]  (1)

where \(a_i\) are the coefficients, and \(\lambda\) is surface reflectance, returning the near-surface concentration of chlorophyll-a in units of mg/m\(^3\), calculated using relationships from in-situ chlorophyll-a measurements and corresponding satellite imagery. \(\lambda_{\text{blue}}\) represents max(Band 1, Band 2) while \(\lambda_{\text{green}}\) represents Band 3. \(a_0\) through \(a_4\) represent instrument-specific coefficients [13, 14].

SST is computed using a linear relationship between Landsat Band 10 (TIRS 1), given as

\[
\text{SST} = a_0 + a_1 \lambda_{\text{TIRS}}
\]  (2)

where \(a_0\) and \(a_1\) represent instrument-specific coefficients, and \(\lambda_{\text{TIRS}}\) represents Band 10 surface reflectance. This algorithm returns the skin sea surface temperature in units of °C, calculated using an empirical relationship derived from in-situ surface temperature measurements and corresponding satellite imagery [15].

B. Classical Ocean Front Detection

Classical ocean front detection is conventionally done on L2 surface reflectance data, which has already been radiometrically calibrated and atmospherically corrected. Two main algorithms exist for detecting ocean fronts in satellite data: the Cayula-Cornillon Algorithm (CCA) and the Belkin O’Reilly Algorithm (BOA). We utilize BOA in our work due to its applicability to small and meso-scale SST and CHL ocean fronts, and its usage of absolute rather than relative threshold for front intensity.

The Cayula-Cornillon Algorithm [7] is the first algorithm developed to detect temperature (SST) ocean fronts. CCA begins by patching the image into overlapping 32 km × 32 km windows and creating a histogram of pixel temperature values for each window. If the histogram is bimodal, then two distinct water populations are present in the window. A cohesion algorithm determines if these two water populations are spatially separate in the window, which implies the presence of an ocean front. If a window is found to have an ocean front, then it is examined on the pixel level. A contour following algorithm finds neighboring pixels belonging to distinct water populations. If a contour is longer than 15 pixels, it is labeled as an ocean front. CCA relies on the relative strength of an ocean front in a scene instead of the absolute strength, making front detection temperature-scale invariant.

While there are many widely accepted methods for detecting SST fronts, there are few established methods for detecting CHL fronts. Traditional noise-reducing median filters often clip important CHL spatial patterns such as blooms and ridges in early algorithmic stages [6].

The Belkin-O’Reilly Algorithm (BOA) is a newer algorithm to detect ocean fronts. While the CCA was developed to detect large-scale temperature fronts, BOA was developed to detect small-scale and mesoscale temperature and chlorophyll fronts. BOA accepts SST and log-normalized CHL satellite imagery and produces gradient magnitude images, where gradient magnitude corresponds to oceanic front strength [6]. BOA utilizes absolute magnitudes for ocean fronts, which makes it less scale invariant but more applicable to large datasets of satellite images.
1. Receive Command
   Ground command specifies location to capture image

2. Target Pointing
   Adjust spacecraft attitude to point camera at desired target

3. Capture Image
   Cameras capture and store images of target location

4. Process Image
   Onboard processing of image to identify features

5. Downlink
   Downlink data on overpasses of ground station

Fig. 1: Concept of Operations for a generic Earth observing CubeSat with edge compute capability. The system is able to extract features from images on-orbit [9, 10].

TABLE I: Active satellite missions that observe the ocean and capture the range of wavelengths necessary to retrieve CHL and SST [16–18].

<table>
<thead>
<tr>
<th>Agency</th>
<th>Operational Missions</th>
<th>Launch Years</th>
<th>Relevant Instruments</th>
<th>Band Count</th>
<th>Spectral Range [µm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>NASA</td>
<td>Terra, Aqua</td>
<td>1999, 2002</td>
<td>MODIS</td>
<td>36</td>
<td>0.41 - 14.2</td>
</tr>
<tr>
<td>NOAA</td>
<td>Suomi-NPP, JPSS-1</td>
<td>2011, 2017</td>
<td>VIIRS</td>
<td>22</td>
<td>0.41 - 12.0</td>
</tr>
<tr>
<td>USGS</td>
<td>Landsat 7, 8, 9</td>
<td>1999, 2013</td>
<td>OLI, TIRS</td>
<td>11</td>
<td>0.43 - 12.5</td>
</tr>
</tbody>
</table>

TABLE II: Landsat 8 spectral bands and corresponding ground sample distance (GSD) [18].

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Band Number</th>
<th>Band Name</th>
<th>Spectral Range [µm]</th>
<th>GSD [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLI</td>
<td>1</td>
<td>Coastal Aerosol</td>
<td>0.43-0.45</td>
<td>30</td>
</tr>
<tr>
<td>OLI</td>
<td>2</td>
<td>Blue</td>
<td>0.45-0.51</td>
<td>30</td>
</tr>
<tr>
<td>OLI</td>
<td>3</td>
<td>Green</td>
<td>0.53-0.59</td>
<td>30</td>
</tr>
<tr>
<td>OLI</td>
<td>4</td>
<td>Red</td>
<td>0.64-0.67</td>
<td>30</td>
</tr>
<tr>
<td>OLI</td>
<td>5</td>
<td>Near Infrared (NIR)</td>
<td>0.85-0.88</td>
<td>30</td>
</tr>
<tr>
<td>OLI</td>
<td>6</td>
<td>Short-Wave Infrared (SWIR) 1</td>
<td>1.57-1.65</td>
<td>30</td>
</tr>
<tr>
<td>OLI</td>
<td>7</td>
<td>Short-Wave Infrared (SWIR) 2</td>
<td>2.11-2.29</td>
<td>30</td>
</tr>
<tr>
<td>OLI</td>
<td>8</td>
<td>Panchromatic</td>
<td>0.50-0.68</td>
<td>15</td>
</tr>
<tr>
<td>OLI</td>
<td>9</td>
<td>Cirrus</td>
<td>1.36-1.38</td>
<td>30</td>
</tr>
<tr>
<td>TIRS</td>
<td>10</td>
<td>Thermal Infrared (TIRS) 1</td>
<td>10.60-11.19</td>
<td>100</td>
</tr>
<tr>
<td>TIRS</td>
<td>11</td>
<td>Thermal Infrared (TIRS) 2</td>
<td>11.50-12.51</td>
<td>100</td>
</tr>
</tbody>
</table>

BOA utilizes a recursive, shape-preserving, contextual median filter that removes noise while keeping CHL features. The filter operates on a 3 km × 3 km window while considering a larger 5 km × 5 km context, analyzing 1D slices of the window and context to determine whether or not to filter the window’s central pixel [6]. This recursive median filter is applied until convergence. Then, Sobel edge detection is applied, and the gradient magnitude is used to calculate pixel-wise ocean front intensity [19, 6].

BOA is computationally intensive. The number of iterations until convergence is $O(N)$, where $N$ is the number of pixels in the image [6]. Experimentally, it takes an average of 22 seconds to process a 224 × 224 pixel image with BOA using an nVidia Tesla T4 GPU on Google Colab.

Fig. 2 shows the overall pipeline going from L0 data to detection of CHL and SST fronts using the Belkin-O’Reilly algorithm. Data is pre-processed first by radiometrically calibrating it and deriving reflectances bringing it up to L2, after which the front detection algorithms can be run to produce a mask showing ocean front locations.

C. Deep Learning Edge Detection

Deep learning provides methods of performing edge detection with more fine tuning ability than classical computer vision methods. Transfer learning from other edge detection datasets to ocean front can allow for increased overall accuracy and a smaller required ocean front dataset [20].

The Berkeley Segmentation Dataset 500 (BSDS500) is
Fig. 2: The traditional image processing pipeline for detecting ocean fronts. The inputs are unprocessed Thermal Infrared (TIRS), Green, and Coastal Aerosol bands. The intermediate processing steps include Level 1 and Level 2 preprocessing, CHL and SST calculations, and the Belkin O’Reilly Algorithm (BOA). The outputs are CHL and SST fronts. This image is constructed with a Landsat training scene [5].

Fig. 3: Sample HED and CEDN model predictions on images from the BSDS500 dataset [22].

This work utilizes Holistically-Nested Edge Detection (HED) [23] and Convolutional Encoder-Decoder Network (CEDN) [24] architectures, which are both types of fully convolutional networks that provide a good balance of computational complexity and performance. The architectures of both models are shown in Fig. 4 and Fig. 5 and both utilize a Visual Geometry Group 16 (VGG-16) backbone [25]. Both models achieve an F_1 score of 0.79 on the BSDS500 dataset, achieving approximately the same performance as humans on the same dataset (0.80) [21].

D. Space Considerations

Space hardware is generally benchmarked for low size, weight, and power (SWaP), sometimes extending this metric to cost as well (SWaP-C). Real-time software has similar analogs, where it may be preferable for algorithms to balance resource usage and overall accuracy. The space environment imposes additional constraints as computational complexity is further limited by heat dissipation. Communications limits are also a consideration, as binary sizes are constrained by the uplink to the satellite. In practice, achieving all of these goals at once is difficult, and methods may excel at all metrics except one [10].

III. APPROACH

We propose replacing every image processing step between Level 0 Landsat data and BOA-detected CHL and SST fronts as shown in Fig. 2 with a deep learning model as shown in Fig. 6. This model will implicitly learn the atmospheric corrections needed to go from Level 0 data to Level 2 data, the NASA algorithms needed to go from Level 2 data to CHL and SST, and the Belkin O’Reilly Algorithm needed to detect ocean fronts. Fig. 6 shows our proposed image processing pipeline.

We structure our machine learning task as a binary classification task—we will assign each pixel a label of “ocean front” or “not ocean front.” In order to transfer learn from our existing state-of-the-art (SOTA) edge detection models, we must create a set of training data with annotated ocean fronts.

A. Creating Training Data

To create a training dataset, we first download 255 cloud-free Landsat 8 scenes using Google Earth Engine. These
To create ground truth ocean front data, we process Landsat Level 2 data using the pipeline detailed in Section II-B. We threshold the Belkin-O’Reilly output at 0.1 °C/km for SST ocean fronts and 0.05 mg/m³/km for CHL ocean fronts. There are no defined absolute thresholds for what constitutes an ocean front, so these thresholds are developed with the input of oceanographer Dr Yackar Mauzole [26].

In parallel, we also process Landsat Level 2 data using the NASA CHL and SST algorithms, apply the Belkin-O’Reilly algorithm, then use a human annotator to label the ocean fronts. This method of creating ground truth data aligns with traditional ML methods of creating ground truth edge detection data.

B. Model Architectures

The HED and CEDN model architectures detailed in Fig. 4 and Fig. 5 are SOTA models for edge detection on natural images. However, our on-orbit ocean front detection task is narrower in scope and more SWaP-constrained than generalized edge detection. Because edge detection generally happens in the first few layers of a computer vision model, it is feasible that decreasing the number of layers in our models could preserve the edge detection capabilities of the models. We also experiment with two “small” model architectures based on HED and CEDN in Fig. 8 and Fig. 9, decreasing the number of model parameters over eight-fold.

C. Training Parameters

To train our machine learning models, we use focal cross entropy loss [27]. This loss function weights examples based on class imbalance, and thus is ideal for our dataset where only 6% of pixels are labeled as ocean fronts.

To prepare our dataset for training, we augment by mirroring and rotation, and subtract the per-channel mean pixel values calculated on the training dataset.

We build our four ML models in Tensorflow, and apply pre-trained VGG weights to the relevant backbone layers. We train our models on the MIT SuperCloud for 100 epochs with a learning rate of 1e-4 [28].

1 Dataset available at https://github.com/MIT-STARLab/deep-learning-ocean-front-detection
Note that because we are using pre-trained VGG weights, we only have three input channels (traditionally R, G, and B). We use the coastal aerosol, green, and TIRS channels as model inputs, as those are the main bands used in traditional methods of detecting ocean fronts [13, 15].

IV. RESULTS

We analyze the performance of our four model architectures across two ground truth datasets using qualitative and quantitative metrics. Qualitatively, we visually compare model outputs to ground truth data. Quantitatively, we calculate binary metrics, F1 scores, and a precision-recall curve for each model. Section A describes model performance on the BOA ground truth dataset, while Section B describes model performance on the human annotated ground truth dataset. Section C compares the quality of SST and CHL front predictions, and Section D delves into model resource utilization.

A. BOA Ground Truth

1) Qualitative Comparison: Fig. [10] provides two examples of model input/output. Note that Small HED captures the mild striping caused by inaccurate calibration of the Landsat pushbroom sensors [18], even though it is not present in the ground truth data. CEDN is blurrier than the other outputs, failing to delineate between separate fronts in some predictions. This blurriness is due to the small size of the center convolutional layer in the encoder-decoder architecture, and could potentially be mitigated by adding skip connections across the encoding and decoding layers [24].

Fig. [11] provides an additional example of model input/output, and shows that the BOA ground truth data is not always perfect. In this figure, the ground truth only captures part of the swirls evident in the input data, while the model outputs capture all of the swirls.

This emphasizes the importance of our pretrained VGG weights; the models already have some robust concept of edge detection that only needs to be refined for ocean front detection. The disparity between ground truth data and model predictions—where the models seem to outperform the ground truth—underscores the generalization capabilities of the models.

2) Quantitative Metrics for CHL: The precision-recall curve for CHL is shown in Fig. [12] and describes model performance across all thresholds. The best threshold for each model and the corresponding F1 score and binary metrics are detailed in Table III. HED performs best across the board, especially in the category of precision.

3) Quantitative Metrics for SST: The precision-recall curve for SST is shown in Fig. [13] and describes model performance across all thresholds. The best threshold for each model and the corresponding F1 score and binary metrics are detailed in Table IV. HED again performs well across the board, but is less dominant in SST front detection than in CHL front detection.

B. Human Annotated Ground Truth

1) Qualitative Comparison: Next, we train our four models on the human annotated ground truth dataset. Fig. [14] provides two examples of model input/output. Small HED continues to capture the mild striping caused by inaccurate Landsat sensor calibration [18], but the striping is also occasionally visible in the Small CEDN and HED outputs. CEDN SST front predictions occasionally “bleed” into CHL front predictions, such as the SST front in the bottom right corner of the first prediction. This “bleeding” could be mitigated by adding skip connections, as discussed in Section A.

Our human annotated ground truth data is still not perfect. The intent of creating human annotated ground truth data was to produce images with long, smooth ocean fronts that can be captured by the human eye but not by BOA. While the human annotated ground truth fronts are longer and smoother than the BOA ground truth fronts, there is still room for improvement. See Fig. [15] where there is a long, smooth-ish ocean front in the bottom right corner of the TIRS input and all model predictions, but a series of short, jagged fronts in the human annotated ground truth. This problem
TABLE III: CHL metrics for models trained with BOA ground truth data. The bolding highlights the highest score for each metric.

<table>
<thead>
<tr>
<th>Model</th>
<th>Best Threshold</th>
<th>F&lt;sub&gt;1&lt;/sub&gt; score</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Specificity</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>HED</td>
<td>0.35</td>
<td>0.67</td>
<td>0.99</td>
<td>0.72</td>
<td>0.99</td>
<td>0.63</td>
</tr>
<tr>
<td>Small HED</td>
<td>0.3</td>
<td>0.56</td>
<td>0.98</td>
<td>0.51</td>
<td>0.99</td>
<td>0.61</td>
</tr>
<tr>
<td>CEDN</td>
<td>0.35</td>
<td>0.55</td>
<td>0.98</td>
<td>0.54</td>
<td>0.99</td>
<td>0.57</td>
</tr>
<tr>
<td>Small CEDN</td>
<td>0.3</td>
<td>0.59</td>
<td>0.98</td>
<td>0.62</td>
<td>0.99</td>
<td>0.56</td>
</tr>
</tbody>
</table>

TABLE IV: SST metrics for models trained with BOA ground truth data. The bolding highlights the highest score for each metric.

<table>
<thead>
<tr>
<th>Model</th>
<th>Best Threshold</th>
<th>F&lt;sub&gt;1&lt;/sub&gt; score</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Specificity</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>HED</td>
<td>0.35</td>
<td>0.82</td>
<td>0.96</td>
<td>0.81</td>
<td>0.97</td>
<td>0.84</td>
</tr>
<tr>
<td>Small HED</td>
<td>0.35</td>
<td>0.74</td>
<td>0.94</td>
<td>0.70</td>
<td>0.96</td>
<td>0.78</td>
</tr>
<tr>
<td>CEDN</td>
<td>0.35</td>
<td>0.76</td>
<td>0.95</td>
<td>0.77</td>
<td>0.97</td>
<td>0.74</td>
</tr>
<tr>
<td>Small CEDN</td>
<td>0.35</td>
<td>0.81</td>
<td>0.96</td>
<td>0.80</td>
<td>0.98</td>
<td>0.81</td>
</tr>
</tbody>
</table>

could be mitigated by increasing the number of human annotators per image from n=1 to n=3 (the number of annotators per image in the BSDS500 dataset) and averaging the annotations [21].

2) **Quantitative Metrics for CHL:** The precision-recall curve for CHL is shown in Fig. [16] and describes model performance across all thresholds. The best threshold for each model and the corresponding F<sub>1</sub> score and binary metrics are detailed in Table [V]. Small HED performs well across the board, as does Small CEDN.

3) **Quantitative Metrics for SST:** The precision-recall curve for SST is shown in Fig. [17] and describes model performance across all thresholds. The best threshold for each model and the corresponding F<sub>1</sub> score and binary metrics are detailed in Table [VI]. HED and Small CEDN are the highest performers.

With this final set of metrics, we can draw some general conclusions about the comparative performance of our four model architectures across our two testing datasets:

- Models are generally better at capturing SST fronts than CHL fronts (discussed further in Section C).
- Models are generally better at producing predictions that align with BOA than predictions that align with human annotation (human annotation as approached in this paper–there may be better ways of doing human annotation as discussed in [21].
- The models that capture intermediate levels of frontal detail (HED, Small CEDN) are the most robust performers quantitatively across all predictions.

### C. Comparing CHL and SST Predictions

Overall, the models are quantitatively better at detecting SST fronts than CHL fronts. There are a couple of reasons why this could be the case:

1) In our ground truth datasets where only 6% of pixels are ocean fronts, only 16% of these ocean front pixels are CHL ocean fronts. This could be because our CHL ocean front threshold value is too high, or just reflect the fact that there are fewer CHL fronts than SST fronts in the oceanic regions our training data comes from. Either way, the models have very few examples of what constitutes a CHL ocean front, so they do not learn how to detect them very well.

2) The existing processing pipeline from input Coastal Aerosol/Green data to output CHL fronts is more complex than the existing processing...
Fig. 10: Sample HED, Small HED, CEDN, and Small CEDN model predictions on images from the BOA ground truth dataset. The far left column is the Level 0 input data. Note the Green channel (not shown) is also a model input, but is excluded from this diagram for simplicity (it is visually similar to the Coastal Aerosol channel). The next column is the thresholded BOA ground truth data. The final four columns are unthresholded outputs from our four models: HED, Small HED, CEDN, and Small CEDN.

Fig. 11: Sample HED, Small HED, CEDN, and Small CEDN model predictions on an image with a poor BOA ground truth. The high-quality model predictions underscore ML’s robustness to low-quality ground truth.
TABLE V: CHL metrics for models trained with human annotated ground truth data. The bolding highlights the highest score for each metric.

<table>
<thead>
<tr>
<th>Model</th>
<th>Best Threshold</th>
<th>F1 score</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Specificity</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>HED</td>
<td>0.25</td>
<td>0.34</td>
<td>0.96</td>
<td>0.27</td>
<td>0.97</td>
<td>0.45</td>
</tr>
<tr>
<td>Small HED</td>
<td>0.25</td>
<td>0.36</td>
<td>0.96</td>
<td>0.29</td>
<td>0.98</td>
<td>0.46</td>
</tr>
<tr>
<td>CEDN</td>
<td>0.25</td>
<td>0.34</td>
<td>0.96</td>
<td>0.27</td>
<td>0.97</td>
<td>0.46</td>
</tr>
<tr>
<td>Small CEDN</td>
<td>0.25</td>
<td>0.34</td>
<td>0.96</td>
<td>0.27</td>
<td>0.97</td>
<td>0.46</td>
</tr>
</tbody>
</table>

TABLE VI: SST metrics for models trained with human annotated ground truth data. The bolding highlights the highest score for each metric.

<table>
<thead>
<tr>
<th>Model</th>
<th>Best Threshold</th>
<th>F1 score</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Specificity</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>HED</td>
<td>0.3</td>
<td>0.46</td>
<td>0.96</td>
<td>0.44</td>
<td>0.98</td>
<td>0.49</td>
</tr>
<tr>
<td>Small HED</td>
<td>0.3</td>
<td>0.46</td>
<td>0.96</td>
<td>0.42</td>
<td>0.98</td>
<td>0.52</td>
</tr>
<tr>
<td>CEDN</td>
<td>0.25</td>
<td>0.35</td>
<td>0.93</td>
<td>0.25</td>
<td>0.94</td>
<td>0.58</td>
</tr>
<tr>
<td>Small CEDN</td>
<td>0.3</td>
<td>0.47</td>
<td>0.96</td>
<td>0.39</td>
<td>0.97</td>
<td>0.59</td>
</tr>
</tbody>
</table>

Fig. 12: CHL precision-recall curve for models trained with BOA ground truth data.

Fig. 13: SST precision-recall curve for models trained with BOA ground truth data.

pipeline from input TIRS data to output SST fronts. For example, Landsat Level 2 processing includes correcting for the scattering and absorbing effects of aerosols, which affect the Coastal Aerosol band more than the TIRS band [12]. Furthermore, the NASA algorithm for computing chlorophyll concentration uses a fourth-order polynomial and three Landsat bands [13], more elaborate than the NASA algorithm for computing sea surface temperature which uses a linear equation and one Landsat band [11]. This additional complexity could make it harder for the models to learn to detect CHL ocean fronts.

3) There is more developed literature around SST measurements and SST fronts than CHL measurements and CHL fronts [1, 29], so the processes surrounding front detection could work better for SST than CHL. While the unique features of CHL fronts are specifically targeted by the contextual filter in BOA [6], there could be other parts of the processing pipeline—such as the satellite instruments—that are better optimized for SST [15].

It is likely a combination of all of these factors that cause the discrepancy between SST front prediction quality and CHL front prediction quality.

D. Resource Utilization

We also analyze the performance of our four model architectures by computing the speed of inference, as well as the storage requirements of inference. The time and memory requirements of our ocean front detection models are a good indicator that these models could be deployed on a CubeSat without exceeding resource constraints.
Fig. 14: Sample HED, Small HED, CEDN, and Small CEDN model predictions on images from the human annotated ground truth dataset.

Fig. 15: Sample HED, Small HED, CEDN, and Small CEDN model predictions on an image with a “poor” human annotated ground truth. The high-quality model predictions underscore ML’s robustness to low-quality ground truth.
1) Speed of Inference: We calculate the speed of inference for each model using Google Colab. Google Colab offers CPUs (Intel(R) Xeon(R) @ 2.30 GHz), GPUs (nVidia Tesla T4), and TPUs (TPUv4) at runtime [30]. Models are benchmarked with no additional runtime optimizations to create a fair comparison. In practice, optimizations such as quantization or mixed precision would be used to speed up inference.

The inference speeds for each model are shown in Table VII:

<table>
<thead>
<tr>
<th>Model</th>
<th>CPU [ms]</th>
<th>GPU [ms]</th>
<th>TPU [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>HED</td>
<td>110</td>
<td>6</td>
<td>110</td>
</tr>
<tr>
<td>Small HED</td>
<td>50</td>
<td>2</td>
<td>70</td>
</tr>
<tr>
<td>CEDN</td>
<td>80</td>
<td>3</td>
<td>70</td>
</tr>
<tr>
<td>Small CEDN</td>
<td>50</td>
<td>1</td>
<td>50</td>
</tr>
</tbody>
</table>

These times represent the average time for one inference, computed over 2000 inferences. Note that GPU speeds are on average 30 times faster than CPU speeds, while TPU speeds are about the same as CPU speeds. The strong performance of GPU makes sense, as GPUs are better suited for ML model inference.

Small CEDN performs the fastest inferences, followed by Small HED, then CEDN, then HED. This roughly matches the number of parameters in each model, but the HED and Small HED models perform slower than expected. This could be explained by the concatenation layer in the HED and Small HED architectures, a non-standard ML model layer for which the processing units may not be optimized [30].

2) Storage Requirements: We calculate the storage requirements for each model by converting our TensorFlow models to TensorFlow Lite models. TensorFlow Lite models are quantized TensorFlow models, optimized for efficient inference in SWaP-constrained embedded systems [31].

The disk space required for storing models and performing inference with models is detailed in Table VIII. The memory needed for storage corresponds to when the model is not performing inference, while the memory needed for inference corresponds to when the model is performing inference.

The memory required for storage correlates directly with the number of parameters in each model, but the memory required for inference is more interesting. Small CEDN requires the least space, followed by Small HED, then HED and CEDN. This could perhaps also be explained by the non-standard concatenation layer in HED and Small HED, for which TensorFlow Lite may not be optimized [31].

V. CONCLUSIONS & FUTURE WORK

We present machine learning models for detecting temperature and chlorophyll ocean fronts from satellite imagery. These models can be deployed on CubeSats to help address the existing spatial, temporal, and computational challenges of detecting ocean fronts.
We discuss the importance of detecting ocean fronts and emerging tools for doing so more effectively. We examine the traditional non-ML ocean front detection pipeline, and use this existing pipeline to create a set of training data. We explore existing ML models and datasets for edge detection, adapting architectures and techniques to fit our task of ocean front detection. We train our models and analyze them across a broad range of performance and resource metrics to determine the best fit for CubeSat deployment.

The HED and Small CEDN models achieve accuracies of at least 96% for detecting CHL and SST fronts on both BOA data and human annotated data. The ML models Small HED and Small CEDN use the least resources: utilizing less than 120 MB to make predictions in less than 0.002 seconds on a Google Colab GPU. This inference speed contrasts sharply with the traditional pipeline speed—which takes up to 16 days to process imagery from Level 0 to Level 2 [12] and 22 seconds to detect ocean fronts using BOA on a Google Colab GPU.

Overall, Small CEDN seems to have the most promise for CubeSat deployment. The mini-encoder-decoder architecture makes fairly accurate predictions while consuming relatively few resources, a good fit for a SWaP-constrained CubeSat.

Detecting ocean fronts on-orbit is a means to an end; a way to queue images for downlink. These relatively simple on-orbit ML models serve as an initial demonstration and pave the way for future, more complex, on-orbit ML work.

A. Future Work

Additional avenues for future work in this area include analyzing the training data to understand the spatial and temporal relationships between CHL and SST ocean fronts, and fine-tuning the trained models on-orbit.

1) Relationship between CHL and SST Fronts: There is future work that must be done to fully characterize the spatial and temporal relationship between CHL and SST fronts. There has been little work done in this area before, because Earth-observing satellites often do not observe the bands necessary to generate high-quality CHL and SST data at the same spatial and temporal resolutions [17] [32].

The training dataset developed in this work provides an opportunity to more fully explore the relationship between CHL and SST fronts, including quantifying the CHL/SST offset in different oceanic regions during different seasons.

2) On-Orbit Fine-Tuning: Our models are trained with Landsat 8 data, and intended for deployment on a CubeSat. If the CubeSat cameras are significantly different from the Landsat 8 cameras, the models could perform worse than expected [18] [33]. We expect this will not be a large concern because our models depend on image-wide gradients, instead of individual pixel values. Additionally, previous work has shown that models trained on Landsat transfer well to on-orbit data but can require additional white balancing and calibration [10]. However, if the shift from Landsat to CubeSat does present a problem, we could explore on-orbit fine-tuning. This process would involve computing approximate ground truths on-orbit (likely using the Sobel operator [19]) and fine-tuning the models on-orbit. The resources needed for on-orbit training are considerably greater than the resources needed for on-orbit inference, so this fine-tuning would have to be resource-conscious.

ACKNOWLEDGMENTS

The authors acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing computing resources that have contributed to the research results reported within this paper.

The authors would like to thank Paul Fucile, Vivian Menezes, Sean McCarthy, and Yackar Mauzole for their input.

The work was developed in relation to BeaverCube-2 hardware development sponsored by the Northrop Grumman Corporation.

REFERENCES


**Biography**

Violet Felt is a machine learning engineer in Kihei, Hawaii. She received her B.S. and M.Eng. degrees in Electrical Engineering Computer Science from MIT in 2022.
**Shreyaam Kacker** is a current graduate student in the MIT Space Telecommunications, Astronomy, and Radiation Laboratory led by Prof. Kerri Cahoy. He received the SM degree from MIT in Aeronautics and Astronautics in 2022, and the MEng degree in Aeronautical Engineering from Imperial College London in 2020. His research specializes in laser communications and machine learning applied to remote sensing.

**Joe Kusters** is a graduate student in the MIT Space Telecommunications, Astronomy, and Radiation Laboratory, specializing in laser communications, flight software, electronics design and autonomous systems. He received his M. Eng. in Electrical Engineering and Computer Science from MIT.

**John Pendergast** is an undergraduate student in the Department of Aeronautics and Astronautics at MIT. He is a candidate for S.B. in Aerospace Engineering, expected May 2023.

**Kerri Cahoy** is an Associate Professor of Aeronautics and Astronautics at MIT and leads the Space Telecommunications, Astronomy, and Radiation (STAR) Laboratory. Cahoy received a B.S. (2000) in Electrical Engineering from Cornell University, and M.S. (2002) and Ph.D. (2008) in Electrical Engineering from Stanford University. Dr. Cahoy previously worked as a Senior RF Communications Engineer at Space Systems Loral, and as a postdoctoral fellow at NASA Ames. Cahoy currently works on nanosatellite atmospheric and ionospheric sensing missions (MicroMAS, NASA TROPICS, AERO/VISTA), optical communications (NASA CLICK), and exoplanet technology demonstration (DARPA DeMi) missions.