

Fast Ocean Front Detection Using Deep Learning Edge Detection Models

Violet Felt¹, Shreeyam Kacker², Joe Kusters, John Pendergrast, and Kerri Cahoy³, *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. The existing in situ and remote sensing measurements of the ocean have inadequate spatial and temporal coverage to map small-scale ocean fronts globally. In addition, 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 the 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, such as CubeSats in mind, as such platforms can address the spatial and temporal coverage challenges. The highest performing models achieve the accuracies of 96% and make predictions in milliseconds using unoptimized desktop CPUs and less than 100 MB of storage; these capabilities are well suited for CubeSat deployment.

Index Terms—CubeSat, deep learning, edge detection, Landsat, machine learning (ML), ocean front.

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 [1].

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 [2]. Recent ocean front tracking indicates irregular and erratic activity [3], [4], potentially caused by climate change, motivating the use of improved lower latency monitoring [5].

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. Fronts are hand-drawn over extracted data in areas of sharp gradients [1].

Remote sensing has allowed for widespread 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 [6], [7]. 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, such as CubeSats [8]. 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 toward Massachusetts Institute of Technology (MIT)’s BeaverCube-2 mission [9], [10], but the concepts and models can apply to any Earth-observing satellite 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.

Manuscript received 28 November 2022; revised 23 March 2023; accepted 19 April 2023. Date of publication 15 May 2023; date of current version 6 June 2023. This work was supported in part by the European Regional Development Fund (ERDF) through the Operational Program for Competitiveness and Internationalization (COMPETE 2020) within the Project “AEROS,” in part by the ERDF through the North Portugal Regional Operational Program (NORTE 2020) within the Project “AEROS,” in part by the BeaverCube 2 project funded by the Northrop Grumman Corporation, and in part by the Portuguese Foundation for Science and Technology (FCT) through the Massachusetts Institute of Technology (MIT) Portugal Partnership 2030 under Grant MPP2030. This work was developed in relation to BeaverCube-2 hardware development sponsored by the Northrop Grumman Corporation, Falls Church, VA, USA. (Corresponding authors: Violet Felt; Shreeyam Kacker.)

The authors are with the Space Telecommunications, Astronomy, and Radiation (STAR) Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: violetfelt@gmail.com; shreeyam@mit.edu). Digital Object Identifier 10.1109/TGRS.2023.3276374

1558-0644 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See <https://www.ieee.org/publications/rights/index.html> for more information.

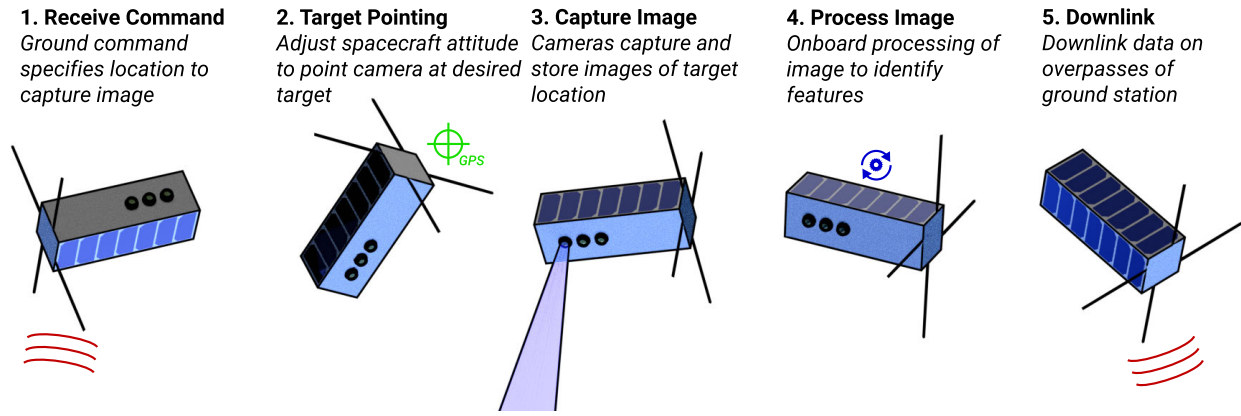


Fig. 1. Concept of operations for a generic Earth-observing CubeSat with edge compute capability. The system is able to extract features from images on-orbit [9], [10].

In addition, 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 I. Primarily, these missions have bands around 440 nm to measure CHL and thermal infrared (TIRS) bands between 8 and 14 μm to measure SST. This work utilizes Landsat 8 data and focuses on ocean fronts in coastal waters, as that is the primary oceanic region Landsat 8 images. The spectral bands and corresponding ground sample distance (GSD) of the Landsat 8 operational land imager (OLI) and thermal infrared sensor (TIRS) are shown in Table II.

National aeronautics and space administration (NASA) uses established algorithms to calculate CHL and SST from surface reflectance data (Landsat Level-2 data) [13], [14], [15]. A different set of coefficients is used for each satellite instrument.

CHL concentration is retrieved using a fourth-order polynomial relationship among Landsat Bands 1 (coastal aerosol), 2 (blue), and 3 (green), given as follows:

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

where the values of a_i are the coefficients and λ 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. λ_{blue} represents $\max(\text{Band 1, Band 2})$, while λ_{green} represents Band 3. a_0 – a_4 represent instrument-specific coefficients [13], [14].

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

$$\text{SST} = a_0 + a_1 \lambda_{\text{TIRS}} \quad (2)$$

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

B. Ocean Front Detection

Classical ocean front detection is conventionally done on L_2 surface reflectance data, which has already been radiometrically calibrated and atmospherically corrected. Two main algorithms exist for detecting ocean fronts in satellite data: the CCA and the BOA. We utilize BOA in our work due to its applicability to small and mesoscale SST and CHL ocean fronts [6], and its usage of absolute rather than relative threshold for front intensity.

The CCA [7] is the first algorithm developed to detect temperature (SST) ocean fronts. CCA begins by patching the image into overlapping 32×32 -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 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

TABLE I
ACTIVE SATELLITE MISSIONS THAT OBSERVE THE OCEAN AND CAPTURE THE RANGE OF WAVELENGTHS
NECESSARY TO RETRIEVE CHL AND SST [16], [17], [18]

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

TABLE II
LANDSAT 8 SPECTRAL BANDS AND CORRESPONDING GROUND SAMPLE DISTANCE (GSD) [18]

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

utilizes absolute magnitudes for ocean fronts, which makes it less scale-invariant but more applicable to large datasets of satellite images.

BOA utilizes a recursive, shape-preserving, contextual median filter that removes noise while keeping CHL features. The filter operates on a 3×3 -km window while considering a larger 5×5 -km context, analyzing 1-D 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 pixelwise ocean front intensity [6], [19].

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 s 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 L_0 data to detection of CHL and SST fronts using the BOA. Data are preprocessed first by radiometrically calibrating it and deriving reflectances bringing it up to L_2 , after which the front detection algorithms can be run to produce a mask showing ocean front locations [20].

Rule-based algorithms may use temporal information to improve detection, as seen in successful applications for tracking ocean fronts in the Bay of Bengal using change-point detection [20].

Deep learning techniques, such as U-Nets, have also been effective for ocean front detection. Li et al. [21] trained a U-Net from scratch on human-labeled data with L_2 imagery as inputs, while Lima et al. [22] utilized ImageNet transfer learning on national oceanic and atmospheric administration (NOAA) datasets for L_2 imagery.

Our work is distinct from previous deep learning techniques in several ways. First, we train a model to identify ocean fronts using L_0 imagery, which is considerably more noisy

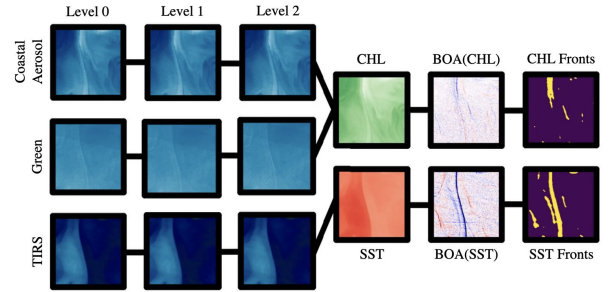


Fig. 2. Traditional image processing pipeline for detecting ocean fronts. The inputs are unprocessed TIRS, green, and coastal aerosol bands. The intermediate processing steps include Level-1 and Level-2 preprocessing, CHL and SST calculations, and the BOA. The outputs are CHL and SST fronts. This image is constructed with a Landsat training scene [5].

than L_2 imagery. Second, we take into account the inference speed and storage requirements of our models to gain a more comprehensive understanding of their efficiency in real-world scenarios. Finally, we adopt a novel deep learning approach from previous work: transfer learning from edge detection models instead of training models from scratch.

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 [23].

The Berkeley Segmentation Dataset 500 (BSDS500) is a popular dataset for training contour/edge detection models. The dataset is composed of 500 images of natural scenes and their corresponding human-annotated edge maps. Each image is annotated by at least three humans, and edge maps are binary images of “edge” and “not edge” pixels [24]. BSDS500 is a benchmark dataset for edge detection models, and models are compared on this dataset using F_1 scores.

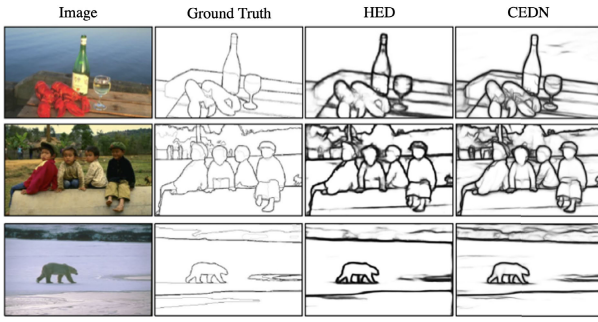


Fig. 3. Sample HED and CEDN model predictions on images from the BSDS500 dataset [25].

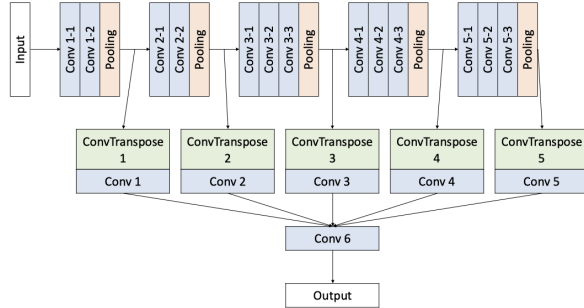


Fig. 4. HED network architecture [26].

Fig. 3 provides three examples of images from the BSDS500 dataset and their human-annotated ground truths, as well as corresponding HED and CEDN model predictions.

This work utilizes holistically nested edge detection (HED) [26] and convolutional encoder-decoder network (CEDN) [27] 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 Figs. 4 and 5, and both utilize a visual geometry group 16 (VGG-16) backbone [28]. Both models achieve an F_1 score of 0.79 on the BSD500 dataset, achieving approximately the same performance as humans on the same dataset (0.80) [24]. We create trained versions of both models (with labeled layers for future transfer learning) using our own TensorFlow implementation, the BSDS500 dataset, and the optimized hyperparameters determined in the papers [26], [27].

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. Despite the limitations of using only three bands for top-of-atmosphere radiometric corrections, the model can compensate by leveraging contextual and spatial features derived from the NASA algorithms that derive CHL and SST, as well as the labeled ocean front data. Fig. 6 shows our proposed image processing pipeline. In addition, the model is only proposed to work on cloud-free input data, as it is not trained to be able to distinguish cloud from ocean areas, which could be provided by having an upstream cloud cover estimation algorithm [10].

We structure our deep learning task as a binary classification task: we 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 Landsat 8 scenes with <5% cloud cover using Google Earth Engine. These scenes are temporally and spatially variant, capturing various frontal structures, such as large-scale water mass fronts, mesoscale fronts around eddies, and local chlorophyll blooms and ridges. The locations of these scenes are detailed in Fig. 7, and all were captured between 2017 and 2021. These scenes consist largely of coastal regions, as those are the predominant oceanic regions imaged by Landsat 8. The specific Landsat scenes used in this dataset are detailed in [5].

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 $0.1^\circ\text{C}/\text{km}$ for SST ocean fronts and $0.05\text{ mg}/\text{m}^3/\text{km}$ 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, i.e., Mauzole [29].

In parallel, we also process Landsat Level-2 data using the NASA CHL and SST algorithms, apply the BOA, and then use a human annotator to label the ocean fronts. This method of creating ground-truth data aligns with traditional deep learning methods of creating ground-truth edge detection data.¹

B. Model Architectures

The HED and CEDN model architectures detailed in Figs. 4 and 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 Figs. 8 and 9, decreasing the number of model parameters over eightfold.

¹Dataset available at <https://github.com/MIT-STARLab/deep-learning-ocean-front-detection>

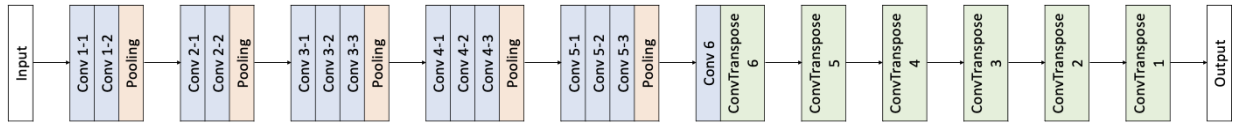


Fig. 5. CEDN network architecture [27].

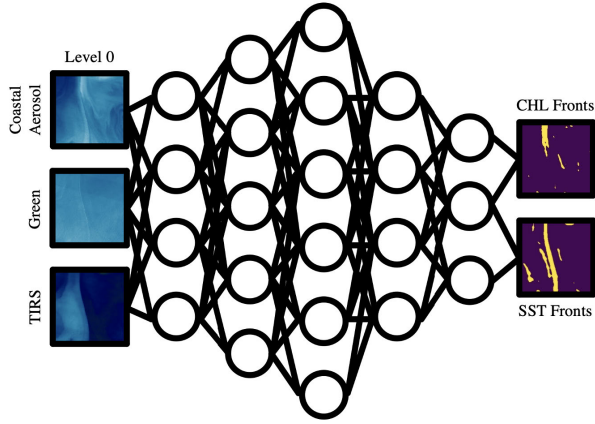


Fig. 6. Proposed image processing pipeline for detecting ocean fronts, utilizing a deep learning model. The inputs are unprocessed TIRS, green, and coastal aerosol bands. The intermediate circles represent neurons in a CNN. The outputs are CHL and SST fronts. This image is constructed with a Landsat training scene [5].

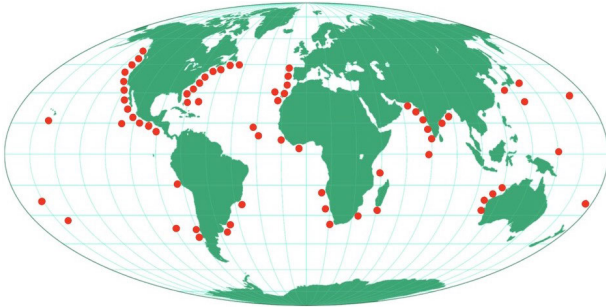


Fig. 7. Locations of Landsat training data.

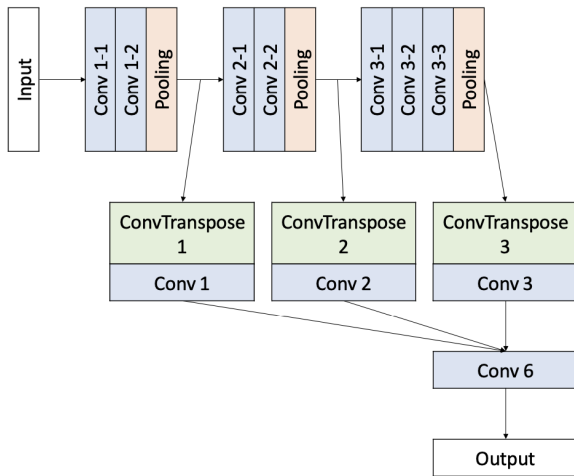


Fig. 8. Smaller version of the HED network architecture as compared with Fig. 4.

C. Training Parameters

To train our deep learning models, we use focal cross-entropy loss [30]. 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 randomly split the dataset into training, testing, and validation partitions using an 80:10:10 ratio. The partitions are kept consistent across both the BOA and the human-annotated datasets.

We build our four deep learning models in TensorFlow and apply pretrained model weights to the matching model layers using TensorFlow's built-in function `get_layer()`. The code for constructing each model architecture is included in [5, Appendix].

We train our models on the MIT SuperCloud for 100 epochs with Adam optimization, a learning rate of $1e-4$, an early stopping patience of ten epochs, and a batch size of 10 [31]. These parameters follow those used to train HED [26] and CEDN [27] after hyperparameter studies. Additional hyperparameter tuning could increase the performance of our models, and we leave that to future work.

Note that because we are using pretrained 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 and a precision-recall curve for each model. Section IV-A describes model performance on the BOA ground-truth dataset, while Section IV-B describes model performance on the human-annotated ground-truth dataset. Section IV-C compares the quality of SST and CHL front predictions, and Section IV-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 push-broom 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. Skip connections are data paths that “skip” intermediate layers of the neural network and use the

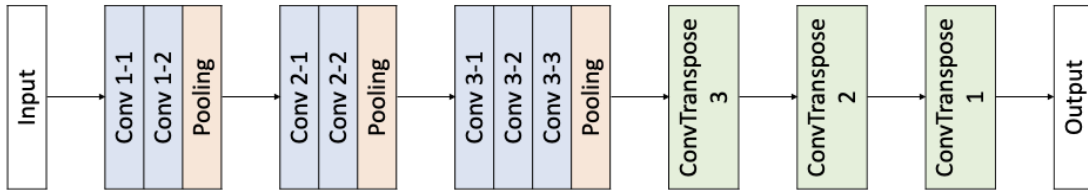


Fig. 9. Smaller version of the CEDN network architecture as compared with Fig. 5.

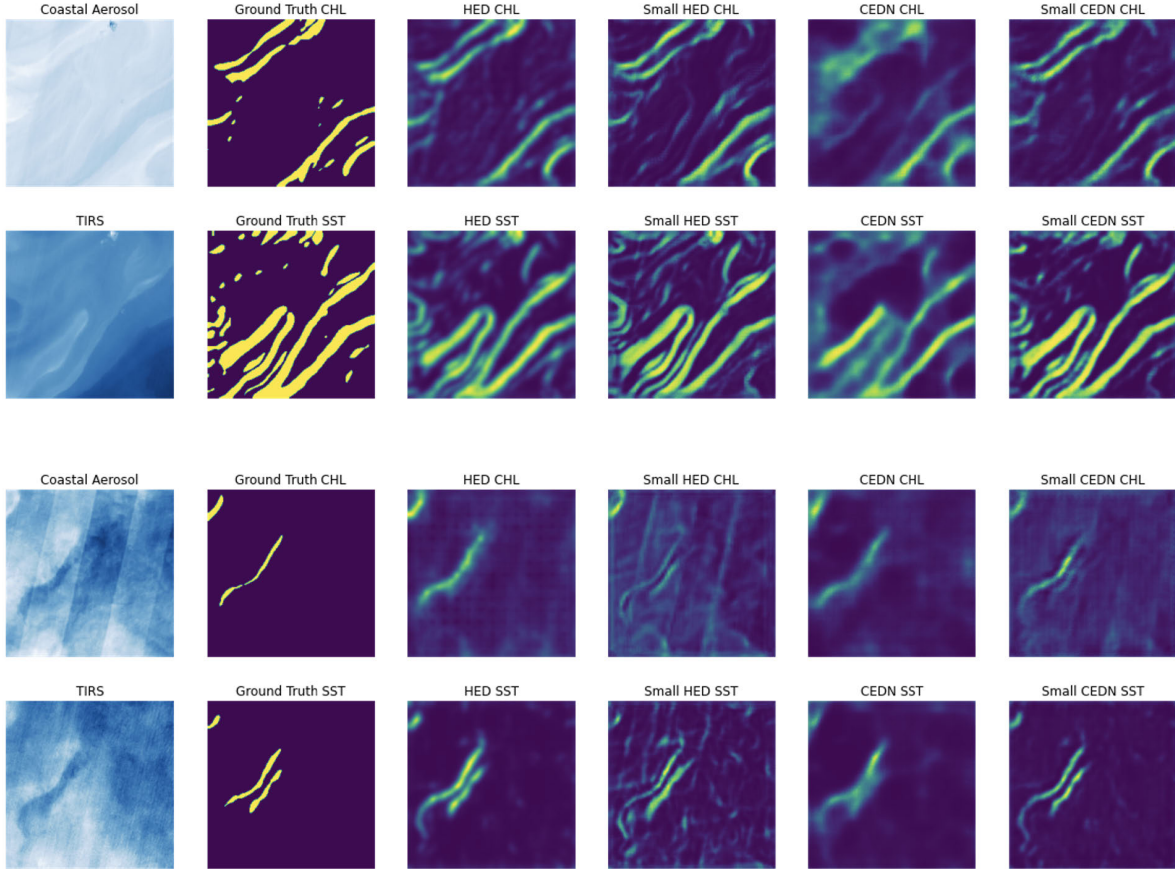


Fig. 10. Sample HED, small HED, CEDN, and small CEDN model predictions on images from the BOA ground-truth dataset. The far left column is the Level-0 input data. Note the green channel (not shown) is also a model input, but is excluded from this diagram for simplicity (it is visually similar to the 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.

output of an earlier layer of the model as an input for a later layer of the model, allowing fine-grained details learned in the first part of the network to propagate to subsequent parts of the network [27].

Fig. 11 provides an additional example of model input–output and shows that the BOA ground-truth data are 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 weights; the models already have some robust concept of edge detection that only needs to be refined for ocean front detection.

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 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 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 IV-A.

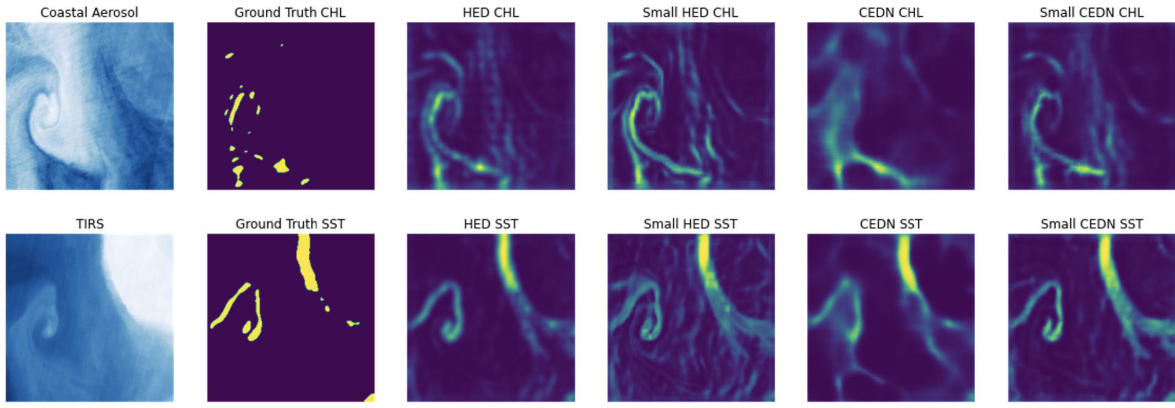


Fig. 11. Sample HED, small HED, CEDN, and small CEDN model predictions on an image with a poor BOA ground truth.

TABLE III

CHL METRICS FOR MODELS TRAINED WITH BOA GROUND-TRUTH DATA. THE BOLDING HIGHLIGHTS THE HIGHEST SCORE FOR EACH METRIC

| Model | Best Threshold | F ₁ score | Accuracy | Precision | Specificity | Recall | IoU |
|------------|----------------|----------------------|-------------|-------------|-------------|-------------|-------------|
| HED | 0.35 | 0.67 | 0.99 | 0.72 | 0.99 | 0.63 | 0.75 |
| Small HED | 0.3 | 0.56 | 0.98 | 0.51 | 0.99 | 0.61 | 0.68 |
| CEDN | 0.35 | 0.55 | 0.98 | 0.54 | 0.99 | 0.57 | 0.68 |
| Small CEDN | 0.3 | 0.59 | 0.98 | 0.62 | 0.99 | 0.56 | 0.70 |

TABLE IV

SST METRICS FOR MODELS TRAINED WITH BOA GROUND-TRUTH DATA. THE BOLDING HIGHLIGHTS THE HIGHEST SCORE FOR EACH METRIC

| Model | Best Threshold | F ₁ score | Accuracy | Precision | Specificity | Recall | IoU |
|------------|----------------|----------------------|-------------|-------------|-------------|-------------|-------------|
| HED | 0.35 | 0.82 | 0.96 | 0.81 | 0.97 | 0.84 | 0.83 |
| Small HED | 0.35 | 0.74 | 0.94 | 0.70 | 0.96 | 0.78 | 0.76 |
| CEDN | 0.35 | 0.76 | 0.95 | 0.77 | 0.97 | 0.74 | 0.78 |
| Small CEDN | 0.35 | 0.81 | 0.96 | 0.80 | 0.98 | 0.81 | 0.82 |

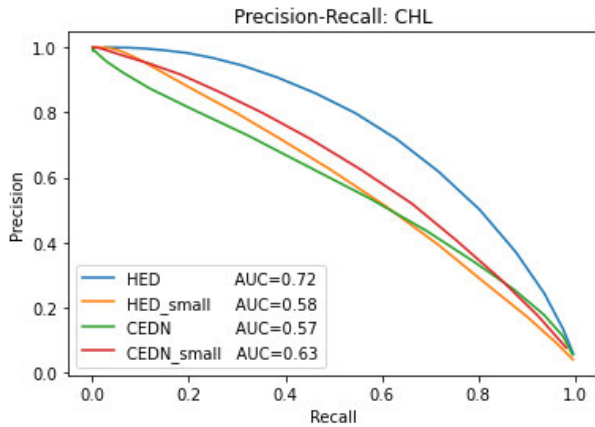


Fig. 12. CHL precision-recall curve for models trained with BOA ground-truth data.

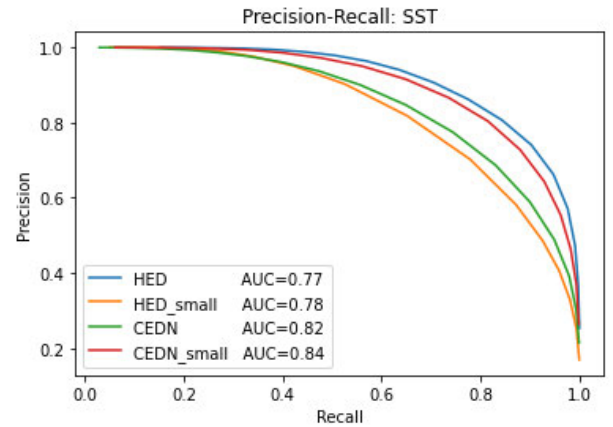


Fig. 13. SST precision-recall curve for models trained with BOA ground-truth data.

Our human-annotated ground-truth data are 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, somewhat smooth 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 could be

mitigated by increasing the number of human annotators per image from $n = 1$ to 3 (the number of annotators per image in the BSDS500 dataset) and averaging the annotations [24], but we leave this time-intensive solution to future work.

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 binary metrics are detailed in Table V. Small HED performs well across the board.

3) *Quantitative Metrics for SST*: The precision-recall curve for SST is shown in Fig. 17 and describes model performance

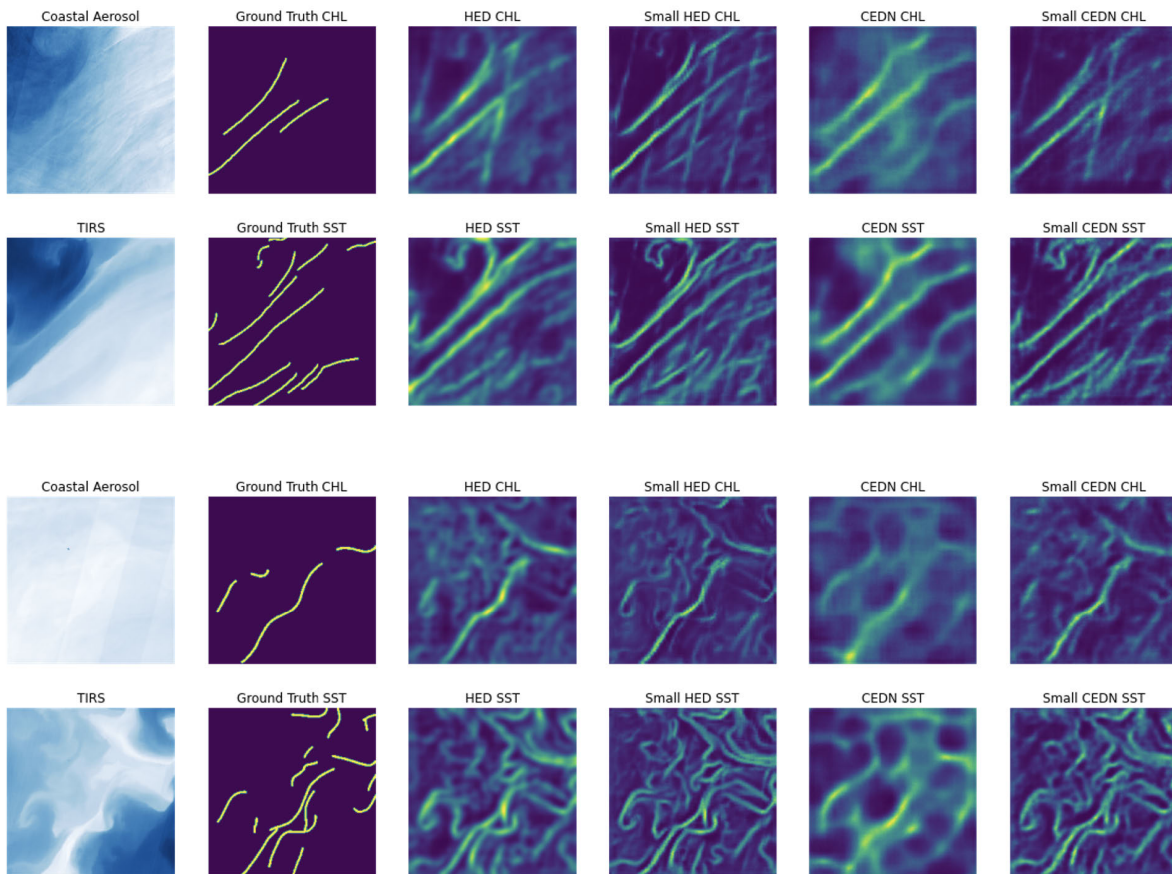


Fig. 14. Sample HED, small HED, CEDN, and small CEDN model predictions on images from the human annotated ground-truth dataset.

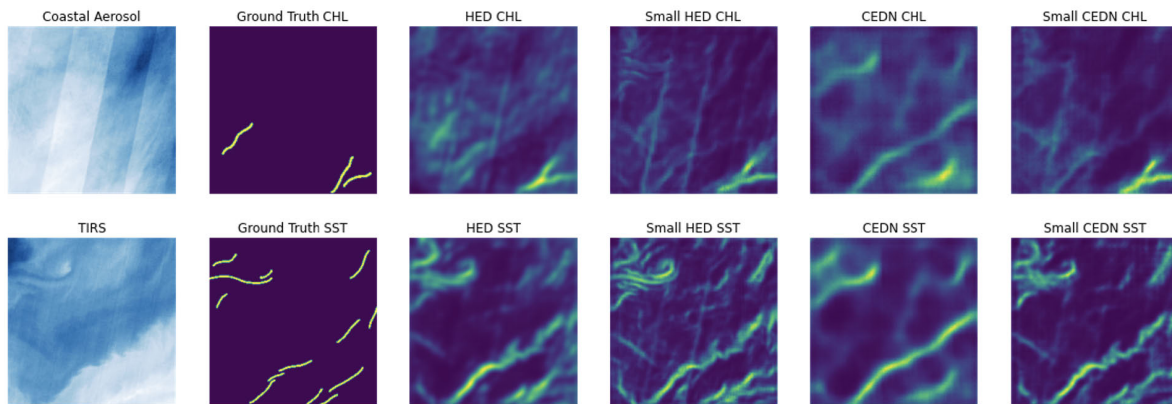


Fig. 15. Sample HED, small HED, CEDN, and small CEDN model predictions on an image with a “poor” human-annotated ground truth.

across all thresholds. The best threshold for each model and the corresponding binary metrics are detailed in Table VI. HED and small CEDN are the highest performers.

4) *BOA Performance on Human-Annotated Ground Truth:* It is also interesting to note the performance of the BOA against the fine-tuned models. Tables V and VI include a final row evaluating the performance of the classical BOA against the trained deep learning models on the human-annotated ground-truth dataset, following standard deep learning practices. Every fine-tuned model outperforms the BOA baseline in F_1 score, the harmonic mean of precision, and recall.

In general, training on human-annotated data leads to degraded model performance metrics likely due to significant label differences between human annotation and BOA. Thinner human-annotated ground-truth labels result in an even larger “ocean front” versus “not ocean front” class imbalance, causing variability in metrics, such as intersection over union (IOU) and difficulty in achieving convergence during backpropagation despite smoother labeled fronts.

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.

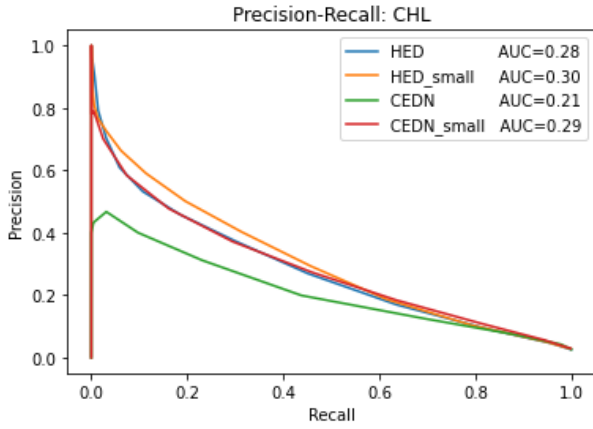


Fig. 16. CHL precision–recall curve for models trained with human-annotated ground-truth data.

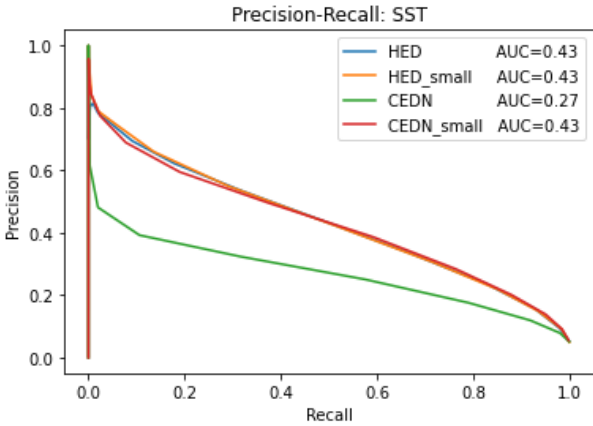


Fig. 17. SST precision–recall curve for models trained with human-annotated ground-truth data.

- 1) SST fronts are easier for models to capture than CHL fronts (see Section IV-C).
- 2) Models tend to align better with BOA than with human annotation, although multiple annotators could potentially improve human annotation quality [24].
- 3) The best-performing and most robust models across all predictions are those that capture intermediate levels of frontal detail (HED and small CEDN).
- 4) Focal loss does not fully address the issue of extreme class imbalance, especially in the human-annotated data case.

C. Comparing CHL and SST Predictions

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

- 1) In our ground-truth datasets, only 6% of pixels are ocean fronts, and of those, only 16% are CHL ocean fronts. This may be due to a high CHL ocean front threshold value or the scarcity of CHL fronts in the oceanic regions used for our training data. Consequently, the models have few examples of CHL ocean fronts, which impedes their ability to detect them.
- 2) The processing pipeline from input coastal aerosol/green data to output CHL fronts is more complex than the processing 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 SST, 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], [32], 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.

1) *Speed of Inference*: We calculate the speed of inference for each model using Google Colab. Google Colab offers CPUs (Intel² Xeon² @ 2.30 GHz), GPUs (NVIDIA Tesla T4), and TPUs (TPUv4) at runtime [33]. 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.

Table VII displays mean inference times and standard deviations for each model, averaged over 2000 inferences. GPU speeds are typically 30 times faster than CPU speeds, while tensor processing unit (TPU) speeds are similar to CPU speeds. GPUs are better suited to deep learning model inference, explaining their superior performance.

Small CEDN performs the fastest inferences, followed by small HED, then CEDN, and 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 nonstandard deep learning model layer for which the processing units may not be optimized [33].

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 and optimized for efficient inference in SWaP-constrained systems [34].

²Registered trademark.

TABLE V

CHL METRICS FOR MODELS TRAINED WITH HUMAN-ANNOTATED GROUND-TRUTH DATA. THE BOLDING HIGHLIGHTS THE HIGHEST SCORE FOR EACH METRIC. BOA IS INCLUDED AS A CLASSICAL BASELINE ALGORITHM FOR COMPARISON

| Model | Best Threshold | F ₁ score | Accuracy | Precision | Specificity | Recall | IoU |
|----------------|----------------|----------------------|-------------|-------------|-------------|-------------|-------------|
| HED | 0.25 | 0.34 | 0.96 | 0.27 | 0.97 | 0.45 | 0.54 |
| Small HED | 0.25 | 0.36 | 0.96 | 0.29 | 0.98 | 0.46 | 0.54 |
| CEDN | 0.25 | 0.27 | 0.95 | 0.20 | 0.96 | 0.44 | 0.52 |
| Small CEDN | 0.25 | 0.34 | 0.96 | 0.27 | 0.97 | 0.46 | 0.54 |
| BOA (baseline) | - | 0.22 | 0.97 | 0.36 | 0.99 | 0.25 | 0.55 |

TABLE VI

SST METRICS FOR MODELS TRAINED WITH HUMAN-ANNOTATED GROUND-TRUTH DATA. THE BOLDING HIGHLIGHTS THE HIGHEST SCORE FOR EACH METRIC. BOA IS INCLUDED AS A CLASSICAL BASELINE ALGORITHM FOR COMPARISON

| Model | Best Threshold | F ₁ score | Accuracy | Precision | Specificity | Recall | IoU |
|----------------|----------------|----------------------|-------------|-------------|-------------|-------------|-------------|
| HED | 0.3 | 0.46 | 0.96 | 0.44 | 0.98 | 0.49 | 0.47 |
| Small HED | 0.3 | 0.46 | 0.96 | 0.42 | 0.98 | 0.52 | 0.44 |
| CEDN | 0.25 | 0.35 | 0.93 | 0.25 | 0.94 | 0.58 | 0.43 |
| Small CEDN | 0.3 | 0.47 | 0.96 | 0.39 | 0.97 | 0.59 | 0.47 |
| BOA (baseline) | - | 0.27 | 0.85 | 0.18 | 0.86 | 0.77 | 0.50 |

TABLE VII

MODEL INFERENCE SPEEDS USING CPU, GPU, AND TPU BACK ENDS ON GOOGLE COLAB

| Model | CPU [ms] | GPU [ms] | TPU [ms] |
|------------|----------|----------|----------|
| HED | 110 ± 24 | 6 ± 3 | 110 ± 22 |
| Small HED | 50 ± 7 | 2 ± 1 | 70 ± 6 |
| CEDN | 80 ± 12 | 3 ± 2 | 70 ± 11 |
| Small CEDN | 50 ± 9 | 1 ± 1 | 50 ± 8 |

TABLE VIII

STORAGE METRICS FOR TENSORFLOW LITE MODELS

| Model | Storage [MB] | Inference Memory [MB] |
|------------|--------------|-----------------------|
| HED | 64.7 | 352.6 |
| Small HED | 7.8 | 119.6 |
| CEDN | 160.0 | 354.4 |
| Small CEDN | 11.1 | 70.1 |

The disk space required for storing models and performing inference with models is detailed in Table VIII. The memory needed for storage corresponds to disk space, while the memory needed for inference corresponds to peak memory usage during 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 also be explained by the nonstandard concatenation layer in HED and small HED, for which TensorFlow Lite may not be optimized [34].

V. CONCLUSION AND FUTURE WORK

We present deep learning models for detecting temperature and chlorophyll ocean fronts from uncorrected satellite imagery, addressing spatial, temporal, and computational challenges when deployed on CubeSats. We discuss the importance of detecting ocean fronts and emerging tools

for doing so more effectively. We explore the existing deep learning models and datasets for edge detection, adapting architectures to fit our task of ocean front detection and comparing with human-annotated data and classical algorithms. We train our models and analyze them across a range of performance and resource metrics to determine the best fit for CubeSat deployment.

The HED and small CEDN models achieve the accuracies of at least 96% for detecting CHL and SST fronts on both BOA data and human-annotated data. The deep learning models, such as small HED and small CEDN, use the least resources: utilizing less than 120 MB to make predictions in less than 0.002 s 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 s to detect ocean fronts with BOA on an NVIDIA Tesla T4 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 deep learning models serve as an initial demonstration and pave the way for more complex on-orbit deep learning.

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, fine-tuning the trained models on-orbit, and assessing model performance by geographic region.

1) *Relationship Between CHL and SST Fronts:* More study is needed to understand the spatial and temporal connection between CHL and SST fronts. This is challenging due to limited satellite data availability—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, inhibiting further study [17], [35]. However, the developed training dataset can facilitate exploration of the CHL/SST offset in various ocean regions and seasons.

2) *On-Orbit Fine-Tuning*: Our Landsat 8-trained models may perform worse on CubeSat cameras that differ significantly from Landsat 8 cameras [18]. However, our models rely on image-wide gradients rather than individual pixel values, reducing this concern. Previous research indicates that Landsat-trained models generally transfer well to on-orbit data with some additional white balancing and calibration [10], although on-orbit fine-tuning may be necessary, which requires more data labeling resources.

3) *Assessment Over Different Geographical Regions*: Ocean front detection difficulty varies across geographical regions, including between coastal and open ocean. While the dataset in this work consists of coastal imagery, future work should involve comparing deep learning model performance to classical algorithm performance in both coastal and open ocean regions, as open ocean may be a region where classical algorithms struggle [29].

ACKNOWLEDGMENT

The authors thank Paul Fucile, Viviane Menezes, Sean McCarthy, and Yackar Mauzole for their input. They acknowledge the SuperCloud and Lincoln Laboratory Supercomputing Center, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, for providing computing resources that have contributed to the research results reported within this article.

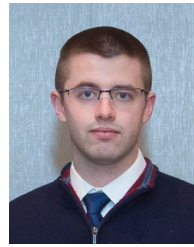
REFERENCES

- [1] I. M. Belkin, "Remote sensing of ocean fronts in marine ecology and fisheries," *Remote Sens.*, vol. 13, no. 5, p. 883, Feb. 2021.
- [2] C. Heinze et al., "The ocean carbon sink-impacts, vulnerabilities and challenges," *Earth Syst. Dyn.*, vol. 6, no. 1, pp. 327–358, Jun. 2015.
- [3] C. Garcia-Soto et al., "An overview of ocean climate change indicators: Sea surface temperature, ocean heat content, ocean pH, dissolved oxygen concentration, Arctic Sea ice extent, thickness and volume, sea level and strength of the AMOC (Atlantic Meridional Overturning Circulation)," *Frontiers Mar. Sci.*, vol. 8, Sep. 2021, Art. no. 642372.
- [4] O. Hoegh-Guldberg et al., "The ocean as a solution to climate change: Five opportunities for action," World Resour. Inst., Washington, DC, USA, Tech. Rep., 2019. [Online]. Available: <https://oceanpanel.org/publication/the-ocean-as-a-solution-to-climate-change-five-opportunities-for-action/>
- [5] V. C. Felt, "Machine learning models for on-orbit detection of temperature and chlorophyll ocean fronts," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2022.
- [6] I. M. Belkin and J. E. O'Reilly, "An algorithm for oceanic front detection in chlorophyll and SST satellite imagery," *J. Mar. Syst.*, vol. 78, no. 3, pp. 319–326, Oct. 2009.
- [7] J.-F. Cayula and P. Cornillon, "Edge detection algorithm for SST images," *J. Atmos. Ocean. Technol.*, vol. 9, no. 1, pp. 67–80, Feb. 1992.
- [8] *System-Level Benefits of the Versal Platform*, Xilinx, San Jose, CA, USA, 2022.
- [9] S. Kacker, A. Meredith, J. Kusters, H. Tomio, K. Cahoy, and V. Felt, "On-orbit rule-based and deep learning image segmentation strategies," in *Proc. AIAA SciTech*, 2022, p. 646.
- [10] S. Kacker, A. Meredith, K. Cahoy, and G. Labréche, "Machine learning image processing algorithms onboard OPS-SAT," in *Proc. AIAA/USU Conf. Small Satell.* Logan, UT, USA: Utah State Univ., 2022.
- [11] K. A. Kilpatrick, G. P. Podesta, and R. Evans, "Overview of the NOAA/NASA advanced very high resolution radiometer Pathfinder algorithm for sea surface temperature and associated matchup database," *J. Geophys. Res., Oceans*, vol. 106, no. C5, pp. 9179–9197, May 2001.
- [12] N. E. Young, R. S. Anderson, S. M. Chignell, A. G. Vorster, R. Lawrence, and P. H. Evangelista, "A survival guide to Landsat preprocessing," *Ecology*, vol. 98, no. 4, pp. 920–932, Apr. 2017.
- [13] C. Hu, Z. Lee, and B. Franz, "Chlorophyll a algorithms for oligotrophic oceans: A novel approach based on three-band reflectance difference," *J. Geophys. Res., Oceans*, vol. 117, no. C1, Jan. 2012, Art. no. C01011.
- [14] J. E. O'Reilly et al., "Ocean color chlorophyll algorithms for SeaWiFS," *J. Geophys. Res., Oceans*, vol. 103, no. C11, pp. 24937–24953, Oct. 1998.
- [15] J.-C. Jang and K.-A. Park, "High-resolution sea surface temperature retrieval from Landsat 8 OLI/TIRS data at coastal regions," *Remote Sens.*, vol. 11, no. 22, p. 2687, Nov. 2019.
- [16] M. M. Abbas, A. M. Melesse, L. J. Scinto, and J. S. Rehage, "Satellite estimation of chlorophyll-a using moderate resolution imaging spectroradiometer (MODIS) sensor in shallow coastal water bodies: Validation and improvement," *Water*, vol. 11, no. 8, p. 1621, Aug. 2019.
- [17] *Visible Infrared Imaging Radiometer Suite (VIIRS) Sensor Data Record (SDR) User's Guide*, NOAA, Washington, DC, USA, 2017.
- [18] R. Morfitt et al., "Landsat-8 operational land imager (OLI) radiometric performance on-orbit," *Remote Sens.*, vol. 7, no. 2, pp. 2208–2237, Feb. 2015.
- [19] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE J. Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, Apr. 1988.
- [20] V. S. Reddem, R. Muthalagu, V. R. Bekkam, P. R. R. Eluri, V. Jampana, and K. Nimit, "Ocean fronts detection over the Bay of Bengal using changepoint algorithms—A non-parametric approach," *Oceanologia*, vol. 63, no. 4, pp. 438–447, Oct./Dec. 2021.
- [21] Y. Li, J. Liang, H. Da, L. Chang, and H. Li, "A deep learning method for ocean front extraction in remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 1502305.
- [22] E. Lima, X. Sun, J. Dong, H. Wang, Y. Yang, and L. Liu, "Learning and transferring convolutional neural network knowledge to ocean front recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, pp. 354–358, Mar. 2017.
- [23] Y. Dar, D. LeJeune, and R. G. Baraniuk, "The common intuition to transfer learning can win or lose: Case studies for linear regression," 2021, *arXiv:2103.05621*.
- [24] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vision. (ICCV)*, Jul. 2001, pp. 416–423.
- [25] X. Soria, E. Riba, and A. Sappa, "Dense extreme inception network: Towards a robust CNN model for edge detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1912–1921.
- [26] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.
- [27] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang, "Object contour detection with a fully convolutional encoder-decoder network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 193–202.
- [28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, 2015. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [29] Y. Mauzole, "Dynamical typology of sea surface temperature (SST) fronts based on satellite observations," Ph.D. dissertation, Univ. Rhode Island, South Kingstown, RI, USA, 2017.
- [30] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.
- [31] A. Reuther et al., "Interactive supercomputing on 40,000 cores for machine learning and data analysis," in *Proc. IEEE High Perform. Extreme Comput. Conf. (HPEC)*, Sep. 2018, pp. 1–6.
- [32] D. S. Ullman and P. C. Cornillon, "Evaluation of front detection methods for satellite-derived SST data using in situ observations," *J. Atmos. Ocean. Technol.*, vol. 17, no. 12, pp. 1667–1675, Dec. 2000.
- [33] Y. E. Wang, G.-Y. Wei, and D. Brooks, "Benchmarking TPU, GPU, and CPU platforms for deep learning," *CoRR*, 2019. [Online]. Available: <https://arxiv.org/abs/1907.10701>
- [34] L. Shuangfeng, "TensorFlow lite: On-device machine learning framework," *J. Comput. Res. Develop.*, vol. 57, no. 9, pp. 1839–1853, 2020.
- [35] D. T. Lauer, S. A. Morain, and V. V. Salomonson, "The Landsat program: Its origins, evolution, and impacts," *Photogramm. Eng., Remote Sens.*, vol. 63, no. 7, pp. 831–838, 1997.



Violet Felt received the B.S. and M.Eng. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2022.

She is currently a machine learning engineer in Kihei, HI, USA.



John Pendergrast is currently pursuing the S.B. degree in aerospace engineering with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA.



Shreeyam Kacker received the M.Eng. degree in aeronautical engineering from the Imperial College London, London, U.K., in 2020, and the S.M. degree in aeronautics and astronautics from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2022, where he is currently pursuing the Ph.D. degree with the Space Telecommunications, Astronomy, and Radiation Laboratory, under the supervision of Prof. Kerri Cahoy.

His research interests include laser communications and machine learning applied to remote sensing.



Kerri Cahoy (Member, IEEE) received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, USA, in 2000, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2002 and 2008, respectively.

She previously worked as a Senior RF Communications Engineer with Space Systems Loral, Palo Alto, CA, USA, and a Post-Doctoral Fellow with National Aeronautics and Space Administration (NASA) Ames, Mountain View, CA, USA.

She is currently Full Professor of aeronautics and astronautics with the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, where she leads the Space Telecommunications, Astronomy, and Radiation (STAR) Laboratory. She works on nanosatellite atmospheric and ionospheric sensing missions (Micro-sized Microwave Atmospheric Satellite (MicroMAS), NASA TROPICS, and AERO/VISTA), optical communications (NASA CLICK), and exoplanet technology demonstration Defense Advanced Research Projects Agency Deformable Mirror Demonstration Mission (DARPA DeMi) missions.



Joe Kusters received the M.Eng. degree in electrical engineering and computer science and the Ph.D. degree in spacecraft systems with the Space Telecommunications, Astronomy, and Radiation Laboratory, specializing in laser communications, flight software, electronics design, and autonomous systems from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA.