IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium | 978-1-6654-2792-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/IGARSS46834.2022.9884819

DEEP ACTIVE CONTOUR MODELS FOR DELINEATING GLACIER CALVING FRONTS

Konrad Heidler [1,2], *Lichao Mou* [1,2], *Erik Loebel* [3], *Mirko Scheinert* [3], *Sébastien Lefèvre* [4], *Xiao Xiang Zhu* [1,2]

[1] Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Weßling, Germany
[2] Data Science in Earth Observation, Technical University of Munich (TUM), Munich, Germany
[3] Institut für Planetare Geodäsie, Technische Universität Dresden, Dresden, Germany
[4] IRISA UMR 6074, Université Bretagne Sud, Vannes, France

ABSTRACT

We present a deep active contour model for detecting and delineating glacier calving fronts from satellite imagery. Contrary to existing deep learning-based calving front detectors, our model does not perform an intermediate segmentation or pixel-wise edge detection, but instead directly predicts the contour parametrized by a fixed number of vertices. The model works by first deriving feature maps from an input image, and then updating an initial contour in an iterative fashion. Evaluating on the CALFIN dataset, which maps calving fronts in Greenland, our model outperforms existing approaches.

Code for the experiments and animated predictions can be found at `https://github.com/khdlr/deep-acm`

***Index Terms*—** Edge detection, Greenland, glacier front

1. INTRODUCTION

The location of marine-terminating glacier fronts is an important indicator for dynamic glacier changes and their response to long-term climate developments. Monitoring these front lines with deep learning has been studied both as a segmentation task, as well as an edge detection task [1].

With deep learning methods on the rise, the task has been approached using specialized convolutional neural networks (CNNs) for image segmentation. While these have proven viable for delineating calving fronts [2, 3], they do not solve the task directly. Instead, they take a detour by first predicting segmentation maps as a proxy for the actual delineation. In line with this observation, recent works show that using CNNs to directly classify edge pixels rather than segmentation masks is a promising approach [4, 5].

The detection of contours or edges is a task that predates the deep learning era. One way of formulating this task is as a dense prediction task, where each pixel is classified as either an "edge pixel" or a "non-edge pixel". Traditionally, simple local methods like the Roberts operator [6] have been used for such tasks. More recently, deep CNNs have shown to be very capable edge detectors. By building feature pyramids of increasing abstraction, they are better able to differentiate between true edges and noisy textures. One notable early work in this direction is holistically-nested edge detection [7].

In contrast to these works, we present a delineation framework that directly derives an explicit delineation from the input imagery. Inspired by Active Contour Models (ACMs) or Snakes [8], our model works by iteratively deforming an initial curve. For the task of building footprint detection, Marcos et al. [9] have proposed to predict ACM parameters using a CNN and backpropagate the error through the ACM iteration. In an inspiring study by Peng et al. [10], the authors showed that the snake evolution itself can be learned using a one-dimensional CNN, implementing a fully trainable Deep Active Contour model for instance segmentation. Transferring this idea to the delineation of glacier fronts requires great care, as the task is fundamentally different from instance segmentation in a number of ways. Instead of small, self-contained objects, glacier front detection requires drawing a boundary between two classes – glacier and ocean – which extend far beyond the given image. Further, the complex shape of calving fronts requires a more sophisticated loss function than the one used in [10].

Convinced by the theoretical advantages of an explicit curve-based delineation approach, we set out to devise such a framework for the delineation of glacier frontlines. The result is an end-to-end trainable model that first calculates high level features using a two-dimensional CNN backbone. In a second stage, an initial curve is iteratively deformed by a one-dimensional CNN that samples values from this feature map. The model is then trained to match the true glacier frontlines with its predictions.

2. METHOD

Active contour models [8] work directly on the brightness values of the given image. Therefore the method is very dependent on local contrast, and has no natural extension to RGB imagery, let alone multi- or hyperspectral imagery. Deep CNNs on the other hand use a large number of filters to extract meaningful information from the raw image pixels.

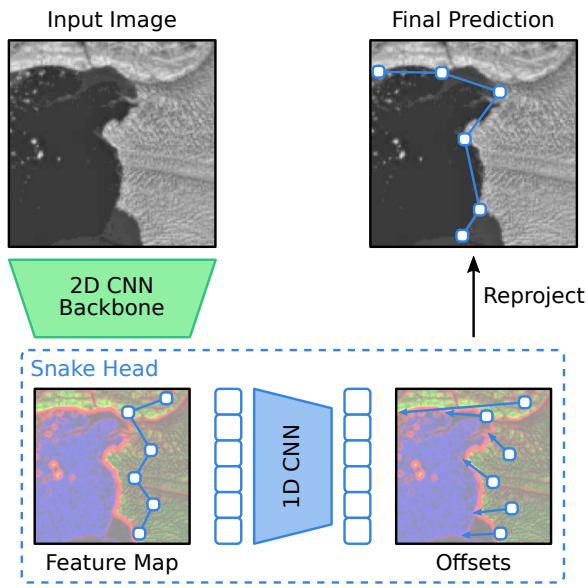

Fig. 1. Overview of our proposed model architecture. After extracting a feature map using the backbone network, features are extracted at initial vertex positions. Then, the snake head calculates offsets for each vertex based on these features. This step is repeated 4 times.

Combining the strengths of these two approaches therefore seems like a natural fit.

In their original formulation, ACMs iteratively evolve a curve by minimizing an energy functional. It is worthwhile to explore the possibilities of putting these two approaches back-to-back in a model that first derives features using a CNN and then applies an ACM to delineate regions of interest on these feature maps. This framework can then be trained end-to-end by backpropagating the loss through the ACM iterations [9].

In this simple concatenation of CNN and ACM, the transition between the two models quickly becomes an information bottleneck. While the CNN can output very rich feature maps, the ACM can only take a small number of features as its input. In order to remove this bottleneck, we replace the traditional snake iteration by a one-dimensional CNN, inspired by the model design proposed in [10].

2.1. Fully Convolutional ACM

Instead of relying on hand-crafted energy terms, our model learns the optimal deformations for a contour directly from the data.

2.1.1. Backbone

Like Peng *et al.* [10], we start by extracting two-dimensional feature maps using a convolutional backbone network. This step essentially replaces hand-crafted feature extraction methods that are sometimes applied prior to an ACM model. While a conventional ACM only samples a single scalar value per vertex, our method samples an entire feature vector of arbitrary size for each vertex, which drastically increases the amount of information available to the snake evolution.

The architecture used for the backbone model is an Xception network [11]. We find in our experiments, that this greatly improves training stability compared to the more popular ResNet architectures, which we attribute to the extensive use of depth-wise convolutions in the Xception architecture.

2.1.2. Snake Head

A critical step for the deep ACM is the transition from two-dimensional imagery to the output curve, which is a one-dimensional sequence of vertices. For each vertex in the current contour, the corresponding features are extracted at the vertex location via bilinear sampling. Then, a one-dimensional CNN is applied to the sequence of vertices in order to predict an offset for each vertex. After applying these offsets, the process of sampling and offsetting is repeated. In our experiments, we find that 4 iterations are enough to warrant satisfactory results.

The snake head can therefore also be viewed as a recurrent neural network, where the vertex locations are the model's recurrent state. Over the iteration, this state is updated and modified until arriving at the final output.

2.2. Reparametrization-Aware Objective Function

A simple method of training this model is to chose a fixed parametrization of the ground truth frontline represented by N vertices w_i and then minimizing the squared sum of the pairwise distances:

$$\mathcal{L}(v, w) = \sum_i \|v_i - w_i\|_2^2 \tag{1}$$

However, the model has no way of knowing how the original parametrization w_i was chosen, and will therefore be penalized even when it predicts the right curve in a wrong parametrization. This issue becomes particularly apparent with the complex outline shapes of glacier frontlines.

Similar issues have been studied in works on timeseries, where the general shape of the sequence is often more important than the exact location of the peaks. For this setting, Sakoe and Chiba [13] proposed a method for comparing timeseries that addresses these matters, called *Dynamic Time Warping* (DTW). It works by re-aligning two timeseries to match each other as closely as possible, and then calculating a distance based on this alignment. More formally, for two sequences v_i and w_j, the DTW loss is defined as

$$\mathcal{L}_{\text{DTW}}(v, w) = \min_{(i_k, j_k) \in \mathcal{K}} \sum_k d(v_{i_k}, w_{j_k}) \,, \tag{2}$$

where $\mathcal{K}$ denotes the set of all possible re-alignments (i_k, j_k) of the two sequences which are non-decreasing and injective.

Table 1. Model Comparison on CALFIN Test Set

Model	Forward MAE	Backward MAE	Forward RMSE	Backward RMSE
UNet [12]	173 m	175 m	248 m	260 m
CALFIN-NN [5]	103 m	115 m	163 m	189 m
Deep ACM (ours)	95 m	111 m	146 m	171 m

Forward = Predicted vertices to ground truth Backward = Ground truth vertices to predicted edge

The choice of the distance measure d in eq. 2 is arbitrary. For our use-case, we set $d(v, w) = \|v - w\|_2^2$.

DTW can also be understood as searching for a minimal-cost path from the top left corner to the bottom right corner of the distance matrix $D_{ij} = d(v_i, w_j)$. Dynamic programming is an efficient way of speeding up the calculation of DTW without explicitly evaluating all possible assignments.

For deep learning purposes however, using DTW directly as a loss function is not optimal, as this dissimilarity measure is not differentiable everywhere and the gradient is only backpropagated through the single optimal path. In order to improve upon these shortcomings, Cuturi and Blondel [14] formulate SoftDTW, which calculates a smooth approximation of DTW. This is achieved by replacing the minimum operator in eq. 2 with a softmin operator, which is defined by

$$\mathrm{softmin}^{\gamma}(x_1, \ldots, x_n) = -\gamma \log \sum_{k=1}^{n} \exp(-x_k/\gamma)\,. \quad (3)$$

The smoothing parameter γ determines the "softness" of the approximation. In the limit $\gamma \to 0$, the regular minimum operator is recovered. In our experiments, we observed the best results when setting $\gamma = 0.001$.

3. EXPERIMENTS & RESULTS

The Calving Front Machine (CALFIN) dataset introduced by Cheng et al. [5] provides a diverse benchmark of calving front locations, as well as an extensive test set. The training dataset contains near-infrared optical data for 1541 Landsat scenes from 1972 to 2019, covering 66 Greenlandic glaciers. To further encourage the model to generalize to other modalities, it also conains 232 single-polarization Sentinel-1 C-band SAR scenes from Antarctica. The corresponding validation dataset consists of 162 Landsat panchromatic scenes. For all of these scenes, manually delineated calving fronts are available.

3.1. Evaluated Models

On this dataset, we train and evaluate our proposed model as well as two competing models, namely a UNet model [12], as well as the CALFIN-NN model presented in [5]. The comparison to the latter is particularly interesting, as it was developed specifically for the dataset used, and incorporates a pixel-wise edge detection. We compare the frontlines obtained from our model to those obtained from the other models in table 1.

3.2. Training Details

All models were trained on the dataset for 500 epochs with a batch size of 16 on an NVIDIA GeForce RTX 3090. Optimization was done using an Adam optimizer with hyperparameters $\beta_1 = 0.5$, $\beta_2 = 0.9$ and an initial learning rate of 10^{-3}, decaying on a cosine schedule to 10^{-5} over 500 epochs.

While the performance of the original ACM is highly dependent on the initialization of the snake [8], we find that the deep ACM does not suffer from this limitation. In fact, the model will find the right contours even when initializing all curve vertices at the center of the image. To avoid leaking information about the ground truth to the model, all results presented in this paper were obtained using this constant initialization scheme.

Regarding the snake head, we find that the model trains more efficiently when stopping the gradients after applying the offsets to the vertices. At the same time, deep supervision is employed on the intermediate predictions, which means that additional loss terms encourage the model to match the ground truth more closely in earlier iterations. As can be seen in Fig. 2, the model already figures out fairly good delineation after the first step, and can thus use the remaining iterations to make adjustments to this contour and match the ground truth.

3.3. Results

Table 1 shows the performance of the three evaluated models on the CALFIN test set. As there is no standardized way of calculating the distance between two curves, we report average vertex-to-curve distances, as suggested in [15]. "Forward" denotes that the respective metric was calculated between the predicted vertices and the ground truth curve. Similarly, "Backward" denotes the opposite, namely calculating the metric between the ground truth vertices and the predicted curve. While the widely used UNet falls far behind the other two models, our proposed deep ACM slightly outperforms the CALFIN-NN, which we attribute to the fact that skipping the intermediate prediction of edge pixels allows the model to more thoroughly learn the nature of calving fronts.

4. CONCLUSION

In this study, we proposed a way of using deep learning to directly predict vectorized calving front lines instead of de-

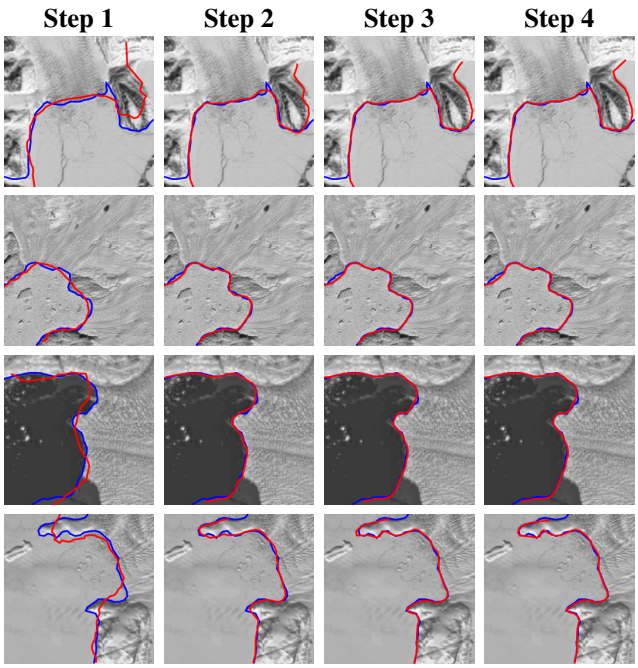

Fig. 2. Model predictions (red) and ground truth (blue) for tiles from the CALFIN test set. Shown are the iteration steps of the model, "Step 4" is the final model output.

riving them indirectly from a segmentation map. The results are promising and competitive with existing models based on segmentation and edge-detection.

5. REFERENCES

[1] Celia A. Baumhoer, Andreas J. Dietz, Stefan Dech, and Claudia Kuenzer, "Remote sensing of Antarctic glacier and ice-shelf front dynamics—A review," *Remote Sens.*, vol. 10, no. 9, pp. 1445, Sept. 2018.

[2] Celia A. Baumhoer, Andreas J. Dietz, C. Kneisel, and C. Kuenzer, "Automated extraction of antarctic glacier and ice shelf fronts from sentinel-1 imagery using deep learning," *Remote Sens.*, vol. 11, no. 21, pp. 2529, Jan. 2019.

[3] Yara Mohajerani, Michael Wood, Isabella Velicogna, and Eric Rignot, "Detection of glacier calving margins with convolutional neural networks: A case study," *Remote Sens.*, vol. 11, no. 1, pp. 74, Jan. 2019.

[4] Konrad Heidler, Lichao Mou, Celia Baumhoer, Andreas Dietz, and Xiao Xiang Zhu, "HED-UNet: Combined Segmentation and Edge Detection for Monitoring the Antarctic Coastline," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[5] Daniel Cheng, Wayne Hayes, Eric Larour, Yara Mohajerani, Michael Wood, Isabella Velicogna, and Eric Rignot, "Calving Front Machine (CALFIN): Glacial termini dataset and automated deep learning extraction method for Greenland, 1972–2019," *The Cryosphere*, vol. 15, no. 3, pp. 1663–1675, Apr. 2021.

[6] Lawrence G Roberts, *Machine Perception of Three-Dimensional Solids*, Ph.D. thesis, Massachusetts Institute of Technology, 1963.

[7] Saining Xie and Zhuowen Tu, "Holistically-nested edge detection," in *IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.

[8] Michael Kass, Andrew Witkin, and Demetri Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, Jan. 1988.

[9] Diego Marcos, Devis Tuia, Benjamin Kellenberger, Lisa Zhang, Min Bai, Renjie Liao, and Raquel Urtasun, "Learning deep structured active contours end-to-end," *ArXiv180306329 Cs*, Mar. 2018.

[10] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou, "Deep snake for real-time instance segmentation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. 2020, pp. 8530–8539, IEEE.

[11] Francois Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, July 2017, pp. 1800–1807, IEEE.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, Eds., Oct. 2015, pp. 234–241.

[13] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 26, no. 1, pp. 43–49, Feb. 1978.

[14] Marco Cuturi and Mathieu Blondel, "Soft-DTW: A Differentiable Loss Function for Time-Series," *ArXiv170301541 Stat*, Feb. 2018.

[15] Janja Avbelj, Rupert Muller, and Richard Bamler, "A Metric for Polygon Comparison and Building Extraction Evaluation," *IEEE Geosci. Remote Sensing Lett.*, vol. 12, no. 1, pp. 170–174, Jan. 2015.