Smart Tracking of Internal Layers of Ice in Radar Data via Multi-Scale Learning

Masoud Yari, Maryam Rahnemoonfar
Computer Vision and Remote Sensing Laboratory
University of Maryland Baltimore County
Baltimore, MD
yari@umbc.edu; maryam@umbc.edu

John Paden, Ibikunle Oluwanisola
Center for Remote Sensing of Ice Sheets (CReSIS)
University of Kansas
Lawrence, KS
paden@ku.edu; ibikunle.oluwanisola@ku.edu

Lora Koenig , Lynn Montgomery
National Snow and Ice Data Center
University of Colorado
Boulder, CO
lkoenig@nsidc.org; lynn.montgomery@colorado.edu

***Abstract*—Artificial intelligence (AI) techniques have displayed impressive success in many practical fields. Deep neural networks (DNNs) owe their success to the availability of massive labeled data. However, in many real-world problems, even when a large dataset is available, deep learning methods have shown less success, due to causes such as lack of large labeled dataset, presence of noise in data, or missing data. In the present work, we intend to examine the application of deep learning methods on radar data gathered from polar regions. Our goal is to track internal ice layers in radar imagery. In such data, the presence of noise is one of the main obstacles in utilizing popular deep learning methods such as transfer learning. Our experiments show that if the neural network is trained to detect contours of objects in electro-optical imagery, it can only track a low percentage of contours in radar data. Fine-tuning and further training do not provide any better results. However, we will show that selecting the right model and training the model on the radar imagery from the base, is going to yield far better results. We also discuss another possible learning approach that can save us time for data annotation.**

***Index Terms*—hidden layers, ice layer tracking, deep learning, multi-scale learning**

I. Introduction

The advancement of artificial intelligence (AI) techniques in recent years has had a great impact on our approaches to data analysis. Deep learning, in particular, has shown great success in many areas of practical interest such as classification [1]–[4], object recognition [5], [6], object tracking [7], counting [8]–[10], and semantic segmentation [11]–[14]. Despite their progress, these algorithms are limited mainly to optical imagery. Non-optical sensors such as radar present a great challenge due to coherent noise in the data. The goal of this work is to test the capability of deep neural network in tracking the internal layers of ice.

We analyze snow Radar [15] images produced by the Center of Remote Sensing of Ice Sheets for NASA Operation IceBridge. The snow radar is a profiling instrument which produces vertical sounding images of snow layers over ice sheets and ice caps. The radar signal is sensitive to annual density changes that occur due to the seasonal transitions from summer to winter; this density change interface scatters the radar signal which is measured by the radar's receiver. Several semi-automated and automated methods exist for surface and bottom tracking in radar images [16]–[23]. Tracking internal layers is a significantly more difficult task because of the large number of layers in close proximity. Panton [24] provided a semi-automatic method for tracing internal layers in radio echograms, but did not apply the algorithm on a large dataset. MacGregor et al [25] developed a semi-automated layer tracker for the CReSIS radar depth sounder, in which the operator could use either a phase coherent method based on signal processing or the method described in [30]. MacGregor's team applied this semi-automated approach to several seasons of data in the first large scale effort to do internal layer tracking. Even using the semi-automated method, the task took several years to complete. Koenig et al. [26] and Medley et al. [27] have tracked these interfaces and used the tracked layers to measure annual snow fall over a large area. Since annual snow fall is critical to understanding how the ice sheet has and will respond to a changing climate, tracking the layers provides a valuable dataset which can be used to understand and improve atmospheric models. Koenig et al [26] applied a semi-automated layer tracker to several seasons of CReSIS snow radar data. Layers that were visible yet less detectable were too difficult to task, but even when constrained to the easier layers the task is still time consuming. Since the techniques are not able to detect all internal layers and are not fully automatic, they cannot be easily scaled up to big dataset for routine application.

Most traditional approaches to edge and contour detection problems fundamentally make use of image spatial derivative operators. Since the derivative operators possess high-pass characteristics of the image, they can effectively enhance edges in an image and make them more pronounced. The downfall of the derivative operator, however, is that they are susceptible to noise. Now in order to reduce the sensibility of

978-1-7281-0858-2/19/$31.00 ©2019 IEEE

derivative operators to noise, one can employ regularization filters, such as a Gaussian filter. Traditional edge detectors, such as canny [28] and the Marr-Hildretch [29], are prime examples of edge detectors that combine regularization with derivative operators. Such methods can generate edge maps in different spatial scales as they apply different bandwidths. Consequently, in large scales, the edge maps tend to capture the low frequency characteristics and global information, while being less sensitive to noise. However, in smaller scales, they are more sensitive to noise as the edge maps tend to capture details of an image or high frequency information.

Pioneer learning based methods utilize supervised models with hand-crafted features. We refer the reader to [30] and references therein.

In this article we experiment with a deep learning technique that has shown great success in other areas of computer vision. We are particularly interested in Holistically-Nested Edge Detection (HED) models, introduced in [31]. We will discuss this in more details in Section II. The basic framework of HED is the same as VGGNet architecture, and it uses the parameter of VGGNet trained on the ImageNet dataset. One difference between the two architectures is that HED does not use the final fully convolution layers. This will allow us to use various sizes images for our training. However, the fundamental difference of HED, with regard to its architecture, is that it introduces the so-called side-outputs which are outputs of some intermediate layers. During training, it uses the side-outputs of five convolutional layers, and a fusion output. The latter, which is generated by fusing the five side-outputs, provides the final predictions of the model. In the next section we will describe the mathematical foundation of the multi-scale learning. In section III, and IV we will describe the experimental results and evaluation of our results and finally we draw the conclusion in section IV.

II. Method

We denote the original data in our training dataset by $X = \{X_n : n = 1 \dots N\}$, where N is the size of the dataset; we also denote the corresponding boundary data by $Y = \{Y_n : n = 1 \dots N\}$. The HED model pulls out M side-outputs by $Y_n^{(m)}$for $m = 1 \dots M$, and a final output of the weighted-fusion layer, denoted by $\widetilde{Y}_n$. The model includes M image-level loss function at each side-output layer, dentoed by ℓ_m for side-output m, and a loss function at the fusion layer, dented by ℓ_f. We denote all parameters of the classifier associated with the m^{th}side-output by θ_m. Then the loss function ℓ_m is defined as a class-balanced cross-entropy function as in equation 1

$$\begin{aligned}\ell_m = \ell_m(\theta_0, \theta_m) = -\beta \sum_{j \in Y^+} logPr(y_j = 1|X; \theta_0, \theta_m) \\ - (1-\beta) \sum_{j \in Y_-} logPr(y_j = 0|X; \theta_0, \theta_m), \quad (1)\end{aligned}$$

where θ_0 represents all other standard network layer parameters, $Y_-$and Y_+ are the edge and non-edge labels respectively and $\beta = |Y_-|/|Y|$. The loss function for the final fusion layer is defined by

$$\ell_f(\theta, w) = CE(Y, \widetilde{Y}), \quad (2)$$

where CE is a cross-entropy loss function that measures dissimilarities of the fused prediction and the ground truth label; $\theta = (\theta_0, \theta_1, \dots, \theta_m)$, and $w = (w_1, \dots, w_m)$ represents the fusion weights. Putting everything together, the goal is to minimize the following objective function via standard (back-propagation) stochastic gradient descent:

$$(\theta, w)^* = \text{argmin} \left(\ell_f + \sum_{m=1}^{M} \ell_m \right) \quad (3)$$

We use a mini-batch gradient descent that computes the gradient of the cost function with respect to the parameters θ for the entire training dataset: $\theta = \theta - \eta \nabla_\theta \mathcal{L}(\theta, x^I, y^I)$ Here we used the symbol θ for all parameters. This minimization approach is based on Nesterov accelerated gradient technique as discussed in [32]:

$$\begin{aligned} v_t &= \mu v_{t-1} + \eta \nabla_\theta \mathcal{L}(\theta_{t-1} - \mu v_{t-1}), \quad (4) \\ \theta_t &= \theta_{t-1} - v_t \end{aligned}$$

where $\mu \in [0, 1]$ is the momentum and $\eta > 0$ is the learning rate, see [32].

Input images are not resized for training or testing. Since we get the side-ouputs right before applying the max pooling, the size of the first output matches with the original input size. But after applying the max pooling in the step, the second side-output is half the size of the first side-output; likewise, each subsequent side-output is going to be half the size of the previous side-output. Therefore, each side-output is generated at a different scale.

III. Experimental Results

Our labeled data set of ice radar imagery, ICE2012 consists of 2360 train images and 260 test images. In the training process, we have used the following parameters $\gamma = 0.1$, learning rate $\eta = 10^{-6}$ and the momentum $\mu = 0.9$ (see [33]). We have also used weight decay rate of 2×10^{-4}.

We present the results of three successful experiments that we carried out:

1) transferred VGG16 parameters, continued training on BSDS500 with augmentation (for instance, see Fig.1)
2) train the model on ICE2012 with a normal distribution initialization (for instance, see Fig.1)
3) train the model on synthetic ice dataset (SYNT_ICE)

In the first experiment, we trained the network on BSDS500 data and its augmentation. The model initialization was based on VGGNet which was trained on ImageNet. We used the same hyper parameters as used in [31] and trained for 10 epochs. We achieved almost the same accuracy as the authors on BSDS test set. Then we applied the trained model on our test data. We used a similar approach for evaluation of our results as in [31]. As shown in Fig.1b the results are not very impressive.

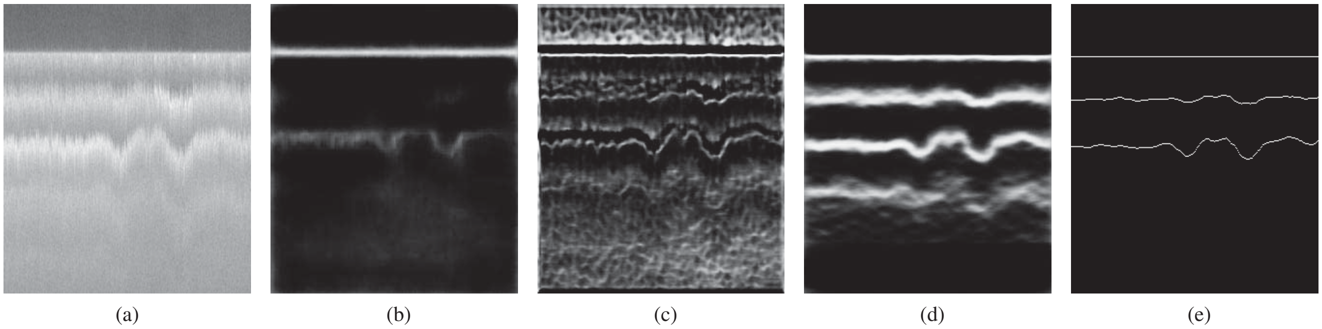


Fig. 1. (a) The original image. (b) The result of training on augmented BSDS500. (c) The result of training on synthetic data. (d) The result of training from scratch on ICE2012. (e) Human labeled edges.

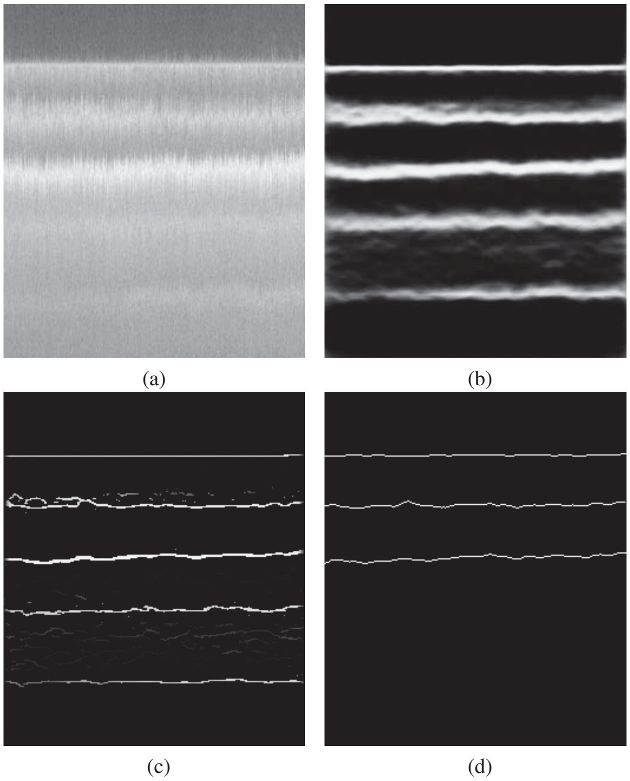


Fig. 2. The test result of the model trained on ICE2012: (a) the original Image (b) the prediction result (c) the non-maximal suppression result (d) the ground Truth.

We argue that the presence of noise in radar imagery, such as our data; makes it almost impossible to use transfer learning methods to detect the edges in our dataset. Fig.1b shows the output of the HED model which is trained on BSDS500 with VGGNet initialization. We trained HED on BSDS500 to achieve the same accuracy as the authors in [31]. However, as one can see in Fig.1b, the results are not good. The quantitative results are presented in Table I. we did further experiments with transfer learning techniques, but all of them failed to converge. For instance, we transferred VGG16 parameters and continued training on our own dataset, ICE2012. In another experiment , we transferred VGGNet parameters, continued training on BSDS500 with augmentation, and continued training it further on ICE2012. Both experiments failed to converge. In fact both algorithms diverged in very early stages. This is due to the presence of noise in images.

The second experiment produced a relatively better result. We trained the model on a synthetic data set of 1000 images.

The synthetic dataset uses a simple linear superposition radar model for the scattering. Each image is produced using a layer thickness model generated with a smoothed Gaussian random process. The thicknesses are summed one of on top of another starting from a flat surface. Each layer is created by summing the contribution of 100 complex (magnitude and phase) randomized targets spread slightly in range to create thickness to the layer. The slight spreading in range is generated from an exponential distribution to simulate the backscatter fall-off from each layer as a function of incidence angle. The signal power of each layer follows an exponentially decay with depth so that deeper layers have weaker signals. The simulator has not been tuned to match the actual snow radar data statistics, but these preliminary results demonstrate that if we can generate more synthetic data with a more complicated noise generator which better matches the snow radar data, we should be able to achieve much better results. This is a subject of our future work.

The third experiment is conducted on our real dataset, ICE2012, with a random initialization. We notice a considerable improvement in our results. Fig.1 d,e show a sample data and the manually annotated ground truth contours side by side of the results of our three experiments. The evaluation results for approach1 through 3 are summarized in Table I.

As discussed in some other resent works, in fact, deep learning models are unstable in the presence of noise. We can witness the same phenomena once again in our experiments. As shown in Table I, training from scratch provides much better results even with respect to the human labeling.

In Fig.II, the original image has different layers, but only 3 top layers are annotated in the ground truth image. With our training strategy, our model has been able to detect 5 layers.

As it was pointed out in the previous sections, HED model produces side outputs in different scales. Fig.3 shows all side outputs of the model and the final fusion for two data. It is apparent that the first side outputs contain more details of the image while the later side outputs project the general structure

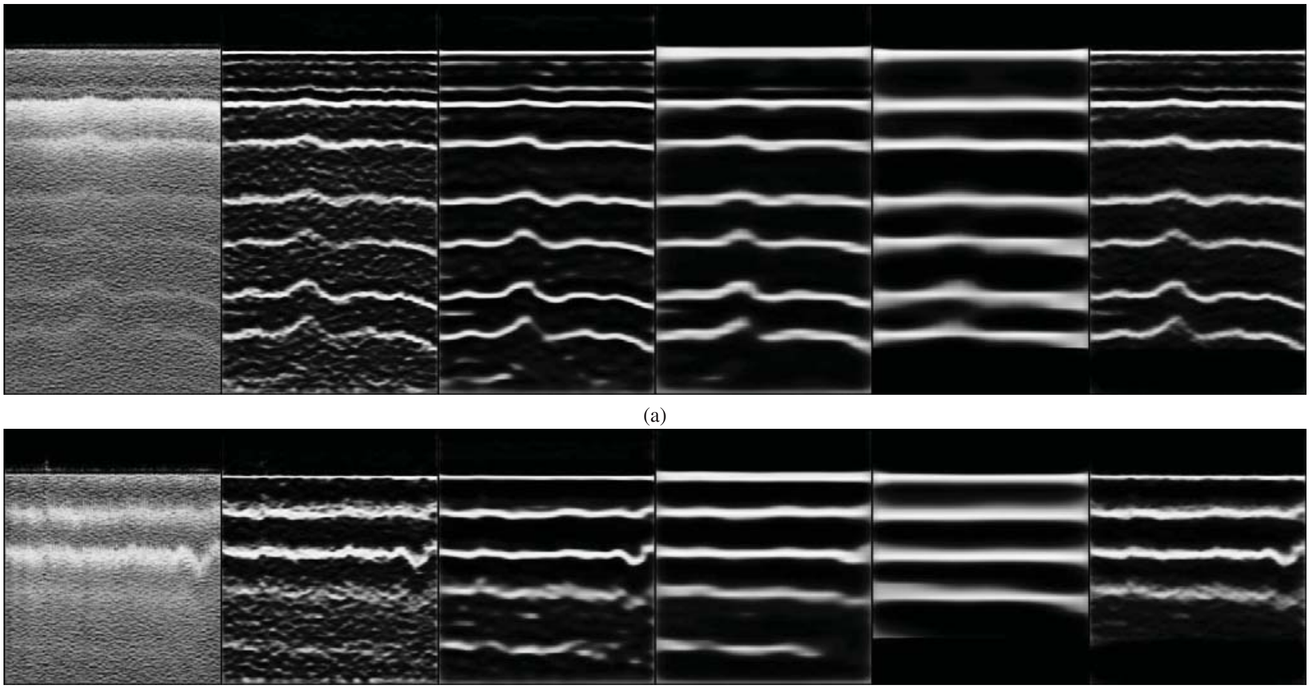


Fig. 3. Form left to right: the first image is the original image. The second image is the first side output which is the same size of the image. The third image is the second side output which is half the size of the first side-output; likewise the third side-output is half the size of the second output, which is the fourth image, so on. The utmost right image is the fusion of the five side-outputs.

of the image.

Fig.4 provides a sample of an image with more fluctuations in the layer boundaries. Comparing to the human level detection, our result shows more layers; thus, it is more accurate.

As one further example of how well the model works in the case of images that contain many internal layers, we have included another sample of our results in Fig.7. The original image contains many layer data. Towards the bottom of the image, the human label annotation has missed some layer information, particularly, it has detected only some parts of one of the layers. However in our case, the model has been able to detect more layers than the human annotation; but it has failed to predict the very last bottom layer.

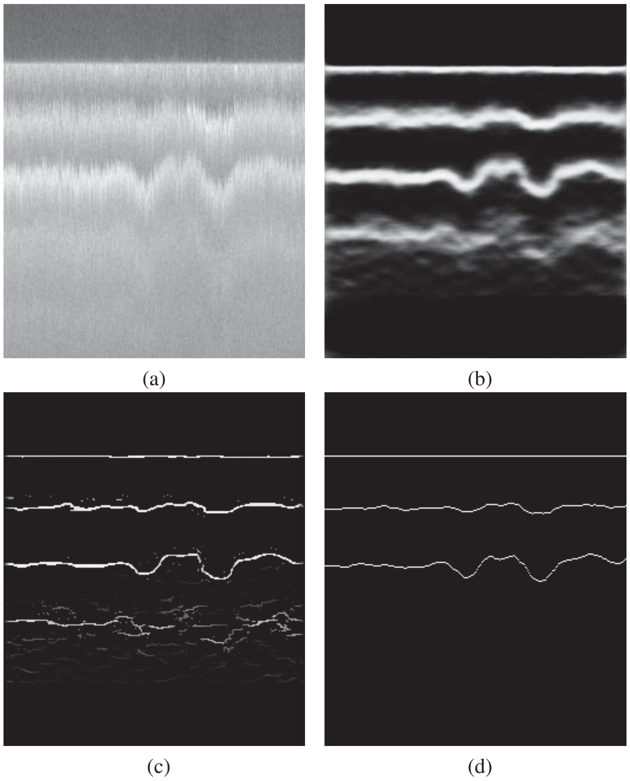


Fig. 4. The test result of the model trained on ICE2012: (a) the original with sharper fluctuations in the layer boundaries (b) the prediction result (c) the non-maximal suppression result (d) the ground-truth.

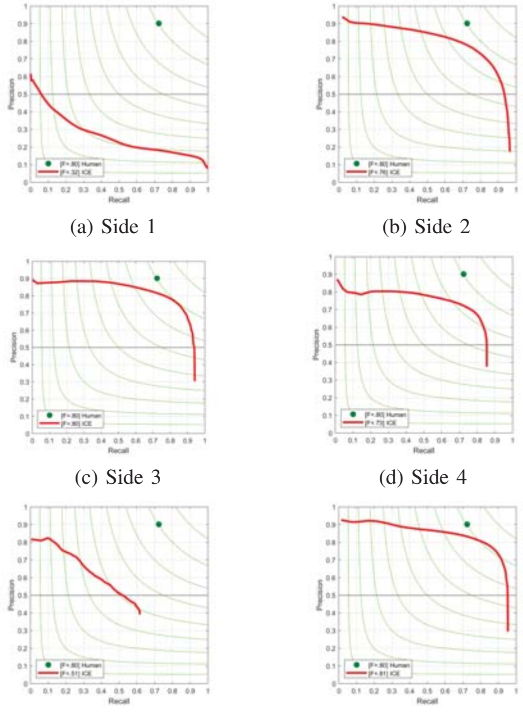
IV. Evaluation

We tested our model on 260 images. The first column in Table I presents the results for transfer learning in which the model is trained on BSDS, and tested on our test set. The second column shows the result of the model trained on the synthetic ice data, and tested on our test dataset. The final column shows the result of the train model on real data ICE2012, and tested on our test dataset.

Here we report three different quantities for an algorithm; the Optimal Dataset Scale (ODS) or best F-measure on the dataset for a fixed scale, the Optimal Image Scale (OIS) or aggregate F-measure on the dataset for the best scale in each image, and the Average Precision (AP) on the full recall range (equivalently, the area under the precision-recall curve), see [34].

TABLE I
EVALUATION RESULTS FOR THE TEST DATA SET

	Transfer learning:Trained on BSDS	Trained on synthetic data tested on ice	Our Results
Side 1	ODS=0.130 OIS=0.111 AP=0.073	ODS=0.175 OIS=0.173 AP=0.123	ODS=0.320 OIS=0.465 AP=0.261
Side 2	ODS=0.162 OIS=0.186 AP=0.082	ODS=0.415 OIS=0.503 AP=0.234	ODS=0.763 OIS=0.779 AP=0.760
Side 3	ODS=0.199 OIS=0.202 AP=0.075	ODS=0.614 OIS=0.628 AP=0.491	ODS=0.796 OIS=0.824 AP=0.786
Side 4	ODS=0.170 OIS=0.196 AP=0.055	ODS=0.385 OIS=0.382 AP=0.206	ODS=0.732 OIS=0.769 AP=0.645
Side 5	ODS=0.276 OIS=0.295 AP=0.138	ODS=0.448 OIS=0.496 AP=0.364	ODS=0.512 OIS=0.572 AP=0.399
Fuse	ODS=0.139 OIS=0.164 AP=0.040	ODS=0.292 OIS=0.379 AP=0.217	**ODS=0.815 OIS=0.854 AP=0.815**



(a) Side 1

(b) Side 2

(c) Side 3

(d) Side 4

(e) Side 5

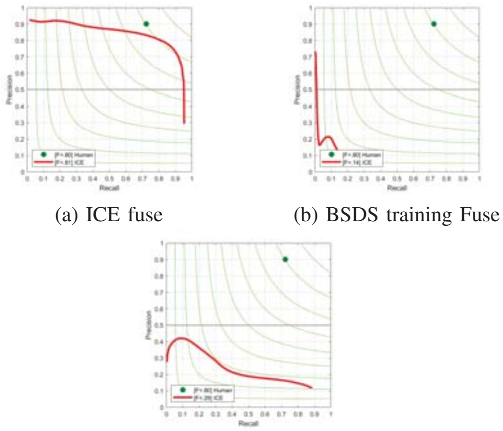
(f) Fuse

Fig. 5. Precision-recall curve for each side-outputs and the fusion of them on the test data.

The accuracy of various method that we discussed in Sec.II are evaluated using the aforementioned standard measures. We also apply a standard non-maximal suppression (NMS) technique to our edge maps to obtain thinned edges for evaluation purposes. The results are shown in Fig.IV, Fig. IV and Table I.

V. CONCLUSIONS

In this work, we have studied a multi-scale deep learning model and various approaches to implement it for detecting ice layers in radar imagery. It is important to note that most of the well-known deep learning approaches work very well on normal images, but can not produce acceptable results in the presence of noise. The fact that deep learning models are not robust with respect to noise are discussed in various works (see [35] for a recent report). In our experiments we have shown that transfer learning approaches do not work well for radar images, while training from scratch yields far better results. However, the latter requires annotated data provided by the domain experts. One way to avoid this would be to generate synthetic data. The current result of synthetic data is not good, but it looks promising. In our future work we are going to try various methods to create synthetic data. We think the right approach to these type of applications is to look into multi-scale models.



(a) ICE fuse

(b) BSDS training Fuse

(c) Synt Fuse

Fig. 6. Evaluation comparison: precision-recall curve for the fuse result on the test set where the model is trained on (a) ICE2012 (b) BSDS500 and (c) Synthetic data

ACKNOWLEDGMENT

This work is supported by NSF BIGDATA award, IIS-1838024, IBM, and Amazon. We also acknowledge the use of data and/or data products from CReSIS generated with support from the University of Kansas, NASA Operation IceBridge grant NNX16AH54G, NSF grants ACI-1443054, OPP-1739003, and IIS-1838230, Lilly Endowment Incorporated, and Indiana METACyt Initiative.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1–9.

[4] C. Sheppard and M. Rahnemoonfar, "Real-time scene understanding for UAV imagery based on deep convolutional neural networks," in *Geoscience and Remote Sensing Symposium (IGARSS), 2017 IEEE International*. IEEE, 2017, pp. 2243–2246.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 580–587.

[6] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, *Simultaneous Detection and Segmentation*. Cham: Springer International Publishing, 2014, pp. 297–312.

[7] M. Rahnemoonfar and H. Alkittawi, "Spatio-temporal convolutional neural network for elderly fall detection in depth video cameras," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 2868–2873.

[8] M. Rahnemoonfar and C. Sheppard, "Deep count: fruit counting based on deep simulated learning," *Sensors*, vol. 17, no. 4, p. 905, 2017.

[9] ——, "Real-time yield estimation based on deep learning," in *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping II*, vol. 10218. International Society for Optics and Photonics, 2017, p. 1021809.

[10] M. Rahnemoonfar, D. Dobbs, M. Yari, and M. J. Starek, "Discountnet: Discriminating and counting network for real-time counting and localization of sparse objects in high-resolution UAV imagery," *Remote Sensing*, vol. 11, no. 9, p. 1128, 2019.

[11] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, Aug 2013.

[12] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3376–3385.

[13] M. Rahnemoonfar, M. Robin, M. V. Miguel, D. Dobbs, and A. Adams, "Flooded area detection from UAV images based on densely connected recurrent neural networks," in *Geoscience and Remote Sensing Symposium (IGARSS), 2017 IEEE International*. IEEE, 2018, pp. 3743–3746.

[14] M. Rahnemoonfar and D. Dobbs, "Semantic segmentation of underwater sonar imagery with deep learning," in *Geoscience and Remote Sensing Symposium (IGARSS), 2017 IEEE International*. IEEE, 2019, pp. 9455–9458.

[15] F. Rodriguez-Morales, D. G.-G. Alvestegui, E. J. Arnold, R. D. Hale, S. Keshmiri, C. J. Leuschen, J. Li, J. D. Paden, and C. Cardenas, "Radar systems for ice and snow measurements onboard manned and unmanned aircraft," *IEEE Latin America Transactions*, vol. 16, no. 9, pp. 2473–2480, 2018.

[16] D. J. Crandall, G. C. Fox, and J. D. Paden, "Layer-finding in radar echograms using probabilistic graphical models," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 1530–1533.

[17] S. Lee, J. Mitchell, D. J. Crandall, and G. C. Fox, "Estimating bedrock and surface layer boundaries and confidence intervals in ice sheet radar imagery using mcmc," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 111–115.

[18] J. E. Mitchell, D. J. Crandall, G. C. Fox, M. Rahnemoonfar, and J. D. Paden, "A semi-automatic approach for estimating bedrock and surface layers from multichannel coherent radar depth sounder imagery," in *SPIE Remote Sensing*. International Society for Optics and Photonics, 2013, Conference Proceedings, pp. 88 921–88 926.

[19] J. E. Mitchell, D. J. Crandall, G. Fox, and J. Paden, "A semi-automatic approach for estimating near surface internal layers from snow radar imagery," in *IGARSS*, 2013, Conference Proceedings, pp. 4110–4113.

[20] M. Rahnemoonfar, M. Yari, and G. C. Fox, "Automatic polar ice thickness estimation from sar imagery," in *SPIE Defense+ Security*. International Society for Optics and Photonics, 2016, pp. 982 902–982 902.

[21] M. Rahnemoonfar, G. C. Fox, M. Yari, and J. Paden, "Automatic ice surface and bottom boundaries estimation in radar imagery based on level-set approach," *IEEE Transactions on Geoscience and Remote Sensing*, 2017.

[22] M. Rahnemoonfar, , A. Abbassi, and F.-G. C. Paden, John, "Automatic ice thickness estimation in radar imagery based on charged particle concept," *IEEE International Geoscience and Remote Sensing Symposium*, 2017.

[23] H. Kamangir, M. Rahnemoonfar, D. Dobbs, J. Paden, and G. Fox, "Deep hybrid wavelet network for ice boundary detection in radar imagery," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 3449–3452.

[24] C. Panton, "Automated mapping of local layer slope and tracing of internal layers in radio echograms," *Annals of Glaciology*, vol. 55, no. 67, pp. 71–77, 2014.

[25] J. A. MacGregor, M. A. Fahnestock, G. A. Catania, J. D. Paden, S. P. Gogineni, S. K. Young, S. C. Rybarski, A. N. Mabrey, B. M. Wagman, and M. Morlighem, "Radiostratigraphy and age structure of the greenland ice sheet," *Journal of Geophysical Research: Earth Surface*, vol. 120, no. 2, pp. 212–241, 2015.

[26] L. S. Koenig, A. Ivanoff, P. M. Alexander, J. A. MacGregor, X. Fettweis, B. Panzer, R. R. Forster, I. Das, J. R. McConnell, M. Tedesco *et al.*, "Annual greenland accumulation rates (2009–2012) from airborne snow radar," *The Cryosphere*, vol. 10, no. 4, 2016.

[27] B. Medley, I. Joughin, B. Smith, S. B. Das, E. J. Steig, H. Conway, S. Gogineni, C. Lewis, A. S. Criscitiello, J. R. McConnell *et al.*, "Constraining the recent mass balance of pine island and thwaites glaciers, west antarctica, with airborne observations of snow accumulation," *The Cryosphere*, vol. 8, no. 4, pp. 1375–1392, 2014.

[28] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[29] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

[30] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 8, pp. 1558–1570, 2014.

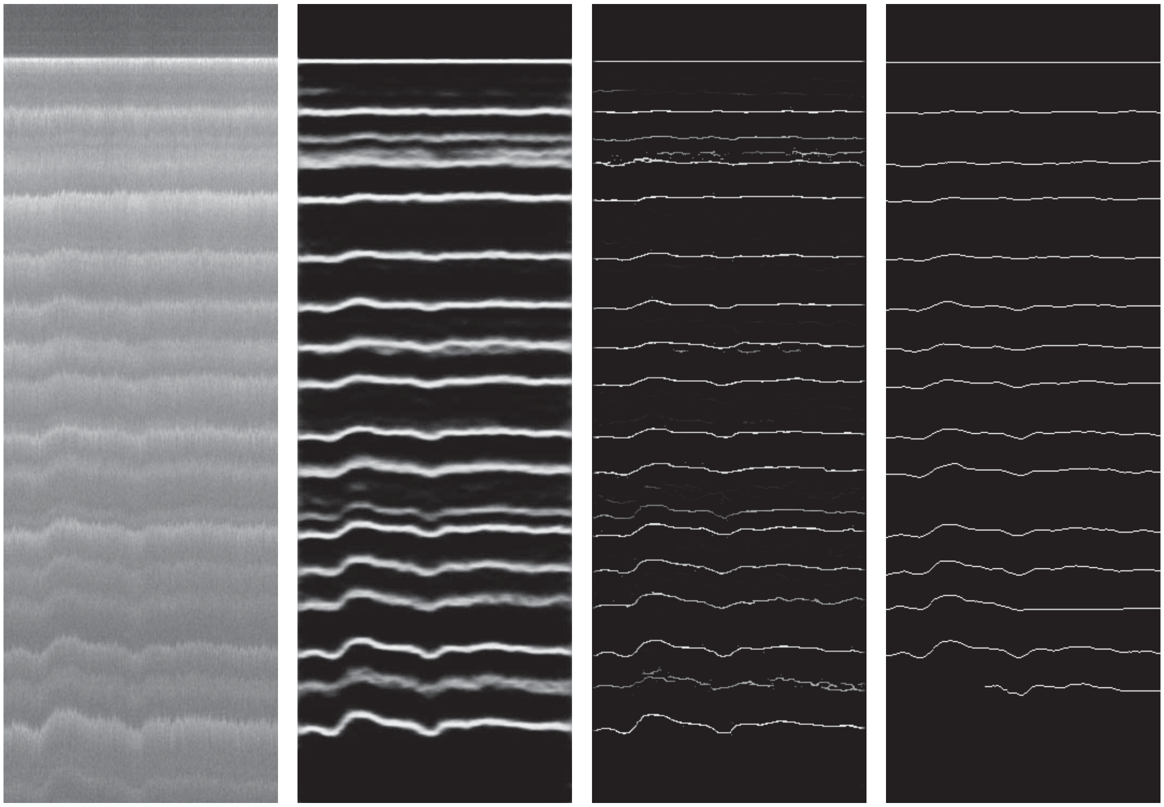
[31] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[32] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

[33] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu, "Statistical edge detection: Learning and evaluating edge cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, pp. 57–74, 2003.

[34] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[35] D. Heaven, "Why deep-learning ais are so easy to fool," pp. 163–166, 2019.



(a) Original Image (b) Prediction (c) NMS Result (d) Ground Truth

Fig. 7. Another sample of our experiment where the image contains high number of layer boundaries. The model is trained and tested on ICE2012.