WEAKLY SUPERVISED AUTOMATIC OBJECT MASKING FOR SYNTHETIC APERTURE SONAR

MS Emigh — Naval Surface Warfare Center, Panama City, FL, USA
H Baker — University of Delaware, Newark, DE, USA
CH Mendoza-Cardenas — University of Delaware, Newark, DE, USA
AJ Brockmeier — University of Delaware, Newark, DE, USA

1 INTRODUCTION

Automatic object masking can be defined, in the context of sonar seafloor imagery, as the process of generating a pixelwise mask that separates objects from the seafloor. It is a challenging problem because conventional segmentation techniques require drawing and labeling masks, which is extremely time-intensive. Segmentation training sets consist of tens of thousands of hand-labeled images. We propose a weakly supervised approach by exploiting the relatively large number of labels contained within a typical synthetic aperture sonar (SAS) automatic object recognition dataset. These datasets typically provide the object center location and object type, but do not contain pixelwise labels. We learn the object masks by optimizing a deep network to mask a subset of pixels in an object image such that when the un-masked part is replaced with a seafloor-only image, the result is statistically indistinguishable from an actual object image. The network is trained using a cost function that estimates the statistical divergence between the real and "counterfactual" images generated by the masks. Notably, this approach ignores the dependence between the statistics of the object and the surrounding seafloor. Nonetheless, it will allow us to gather joint statistics on objects and the seafloor they lie on. Accurate object masking allows the gathering of statistics of objects versus the seafloor they lie on, estimation of signal-to-noise ratio (SNR), and estimation of objects' dimensions. It also allows for synthetically generating new object imagery on other seafloors. This has the potential of augmenting object recognition datasets.

2 RELATED WORK

Although no works directly related to weakly or unsupervised sonar object masking have been found, a number of related approaches have recently been applied to natural imagery. Remez *et al.*[1] proposed a GAN-based, weakly-supervised approach to segment objects output by the faster R-CNN object detector. The GAN generator attempts to learn a mask which is then used to cut and paste the object to a different location. The discriminator then attempts to distinguish between real images and the ones generated by the cut-and-paste masks. They train an independent model per class (e.g., car, person). To prevent degenerate solutions, they added a pre-trained classifier with a classification loss to ensure the object is still in the copy-paste image.

Bielski *et al.*[2] proposed a GAN-based, unsupervised approach. They trained a GAN to produce mask, foreground, and background images and then alpha-blended the foreground and background based on the mask. To prevent degenerate solutions (e.g., an empty mask), they perturbed the masked foreground by randomly shifting it before blending the images. Finally, they trained an autoencoder model where the generator served as a decoder to segment images. Their method, however, is class-dependent and the performance degrades when training the model for more than two classes (e.g., horses and cars). Similarly, Yang *et al.*[3] used layered GANs-based to generate foreground, mask and background images. They trained a segmentation network based on the generated data by the GAN model.

Savarese *et al.*[4] proposed a method to segment foreground and background through masking in order to maximize the inpainting errors in both the region of foreground removed from the background and the complement. The inpainting is performed using a pre-trained inpainting network. The intuition

is that if correctly masked such that no background remains in the foreground and vice versa, the inpainter will fill the regions with distinct imagery. Their initial masking method does not train a model, instead, it iteratively adjusts the mask using an optimization procedure for each image, assuming the initial mask is a small centered square. They then explore using the image and estimated mask pairs that resulted in the highest inpainting error (assuming they are correct) to train a masking neural network; this achieved moderate improvements.

3 METHODOLOGY

We consider the set of probability distributions $\mathcal{P}_\mathcal{I}$ over the domain $\mathcal{I}$. Let $\mu, \nu \in \mathcal{P}_\mathcal{I}$ denote probability measures on $\mathcal{I}$. Here, we consider $\mathcal{I}$ to be a subset of d-dimensional real-valued vector space $\mathcal{I} \subseteq \mathbb{R}^d$, i.e., the space of SAS images with d pixels. The measures μ and ν represent distributions of different SAS image types, e.g., various seafloor textures or objects. Let $X \sim \mu$ and $Y \sim \nu$ denote random variables $X, Y \in \mathcal{I}$ representing SAS images distributed according to μ and ν.

A statistical divergence is a function that quantifies the dissimilarity between two probability distributions. Let $D(\mu, \nu)$ denote a divergence $D : \mathcal{P}_\mathcal{I} \times \mathcal{P}_\mathcal{I} \to [0, \infty)$. It is a distance metric (a probability metric) if all of the following statements hold: (i) $\mu = \nu \Rightarrow D(\mu, \nu) = 0$, (ii) $D(\mu, \nu) = 0 \Rightarrow \mu = \nu$, (iii) $D(\mu, \nu) = D(\nu, \mu)$, (iv) $D(\mu, \nu) \leq D(\mu, \xi) + D(\nu, \xi)$.

In this paper, we consider variants of the Wasserstein metric. It is a divergence, and as should be clear from its name, satisfies the requirements to be a distance metric. The Wasserstein-2 distance $W_2(\mu, \nu)$ is

$$W_2(\mu, \nu) \triangleq \sqrt{\inf_{\gamma \in \Gamma(\mu,\nu)} \mathbb{E}_{(X,Y)\sim\gamma} \|X - Y\|_2^2}, \tag{1}$$

where $\Gamma(\mu, \nu)$ defines the set of all joint distributions with marginals μ and ν. Wasserstein-2 distance is a type of "earth-mover's" distance; if distributions μ and ν are visualized as piles of dirt, it measures the cost to transform one distribution into the other in the most efficient way possible.

Unfortunately, computation of Wasserstein distance is too demanding in high-dimensional spaces such as SAS imagery. The sliced Wasserstein distance[5,6,7], max-sliced Wasserstein distance[8], and further generalizations[9] are computed along one-dimensional linear or non-linear subspaces. They can be expressed by the Radon transform of integrable functions[10].

The linear max-sliced Wasserstein-2 (MSW-2) distance (squared) is

$$D^2_{\mathsf{MSW}_2}(\mu, \nu) = \sup_{W \in \mathbb{S}^{d-1}} \inf_{\gamma \in \Gamma(\mu,\nu)} \mathbb{E}_{(X,Y)\sim\gamma}[\langle X - Y, W\rangle^2], \tag{2}$$

where $\mathbb{S}^{d-1}$ is the d-dimensional unit hypersphere. Comparing equations (1) and (2), linear MSW-2 is simply the Wasserstein-2 distance computed on an (optimal) one-dimensional projection. Slicing is motivated by the relative ease of computing the Wasserstein-p distance in one dimension, since it has a closed form[11] and in the case of two samples only requires a sorting procedure versus computing the full optimal transport plan $\gamma^* \in \Gamma(\mu, \nu)$.

While not as simple to interpret as the general case, the recently introduced distributional form of the sliced Wasserstein distance[12], which optimizes the distribution over *multiple* slices, can be used to compute discrepancies across different subspaces. Let $\mathbb{M}_C \subset \mathcal{P}_{\mathbb{S}^{d-1}}$ denote the family of distributions over the unit hypersphere $\mathbb{S}^{d-1}$ such that for any $\xi \in \mathbb{M}_C$, $\mathbb{E}_{U,W\sim\xi}[|\langle U, W\rangle|] \leq C$. This constraint ensures the slices are not too concentrated. The distributional sliced Wasserstein distance is

$$D_{\mathsf{DSW}_2^\xi}(\mu, \nu) \triangleq \sup_{\xi \in \mathbb{M}_C} \left(\mathbb{E}_{W\sim\xi} E^2_{\mathsf{SW}_2}(\mu, \nu; W)\right)^{\frac{1}{2}}, \tag{3}$$

$$E^2_{\mathsf{SW}_2}(\mu, \nu; W) \triangleq \inf_{\gamma \in \Gamma(\mu,\nu)} \mathbb{E}_{(X,Y)\sim\gamma}[\langle X - \mathbf{y}, W\rangle^2]. \tag{4}$$

Comparing equations (1) and (4), E_{SW_2} is simply the (squared) Wasserstein-2 distance for slice W, while equation (3) maximizes the divergence by adjusting the distribution of slices over $\mathbb{M}_C$. Notably, as $C \to 1$, $D_{\mathsf{DSW}_2^C}(\mu,\nu) \to D_{\mathsf{MSW}_2}(\mu,\nu)$. For unconstrained optimization, the constraint on $\mathbb{M}_C$ can converted to an equivalent regularized form with parameter $\lambda > 0$, such that for a given $0 \geq C \geq 1$, there exists a $\lambda_C > 0$ such that

$$D_{\mathsf{DSW}_2^C}(\mu,\nu) = \sup_{\xi \in \mathcal{P}_{\mathbb{S}^{d-1}}} \left\{ \left(\mathbb{E}_{W\sim\xi} E^2_{\mathsf{SW}_2}(\mu,\nu;W)\right)^{\frac{1}{2}} - \lambda_C \mathbb{E}_{U,W\sim\xi}[|\langle U,W\rangle|] \right\} + \lambda_C C. \tag{5}$$

Following Nguyen *et al.*[12], we limit the search for ξ to a family of distributions. We consider the push-forward measure $\xi = f_\sharp \xi^{d-1}$, where ξ^{d-1} is uniform on $\mathbb{S}^{d-1}$ and $f : \mathbb{S}^{d-1} \to \mathbb{S}^{d-1}$ is a parameterized function $f_A(z) = \frac{1}{\|Az\|_2} Az \quad \forall z \in \mathbb{R}^d \setminus \{0\}$, where A is a square matrix and f_A a normalized linear projection. Given a random vector Z distributed according to the uniform unit hypersphere ξ^{d-1} (or equivalently, from an isotropic Gaussian distribution), the random slice $W = f_A(Z)$ is distributed according to ξ. Intuitively, the eigenstructure of A shapes the distribution of the slices ξ. If A is rank-1, then distributional slicing corresponds to max-slicing (since the sign of the slice does not affect the divergence). If A is a scaled orthogonal matrix, then the distributional slicing corresponds to uniform slicing.

In practice, we have two samples/batches of SAS images $\{\mathbf{x}_i\}_{i=1}^m$ and $\{\mathbf{y}_i\}_{i=1}^n$ (real or synthetic) drawn from μ and ν of size m and n, respectively, which can be expressed as empirical measures $\hat{\mu} = \frac{1}{m}\sum_{i=1}^m \delta_{\mathbf{x}_i}$ and $\hat{\nu} = \frac{1}{n}\sum_{i=1}^n \delta_{\mathbf{y}_i}$. Likewise, during the optimization of (5), we can sample multiple batches of n_w slices $\{\mathbf{z}_i\}_{i=1}^{n_w}$. This leads to an unconstrained stochastic optimization that is an estimate of (5)

$$\hat{D}_{\mathsf{DSW}_2^C}(\hat{\mu},\hat{\nu}) = \max_{A\in\mathbb{R}^{d\times d}} \mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_{n_w}\sim\xi^{d-1}} \left\{ \left(\frac{1}{n_w}\sum_{i=1}^{n_w} E^2_{\mathsf{SW}_2}(\hat{\mu},\hat{\nu};A\mathbf{z}_i) \right)^{\frac{1}{2}} - \frac{\lambda_C}{n_w}\sum_{i\neq j} |\mathbf{z}_i^\top A \mathbf{z}_j| \right\} + \lambda_C C. \tag{6}$$

For equal sized batches, $m = n$, $E^2_{\mathsf{SW}_2}(\hat{\mu},\hat{\nu};\mathbf{w}) = \frac{1}{m}\sum_{j=1}^m (\mathbf{w}^\top \mathbf{x}_{\pi_j} - \mathbf{w}^\top \mathbf{y}_{\sigma_j})^2$, where the permutations $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}$ ensure that $\mathbf{w}^\top \mathbf{x}_{\pi_1} \leq \mathbf{w}^\top \mathbf{x}_{\pi_2} \leq \cdots \leq \mathbf{w}^\top \mathbf{x}_{\pi_m}$ and $\mathbf{w}^\top \mathbf{y}_{\sigma_1} \leq \mathbf{w}^\top \mathbf{y}_{\sigma_2} \leq \cdots \leq \mathbf{w}^\top \mathbf{y}_{\sigma_m}$, respectively. Intuitively, $\mathbf{w}$ defines a subspace and the sorting ensures the shortest distances between the pairs in the subspace.

4 APPROACH

Synthetic aperture sonar automatic object recognition datasets typically provide, at a minimum, object center locations. Using this information, we produced a foreground (object) dataset by collecting square snippets of centered object imagery. For each foreground image, we also collected background (seafloor) images by taking square snippets of the same size centered at a random nearby locations. Since seafloor imagery is typically sparse, the vast majority of the collected background images do not contain (foreground) objects. To produce a test set, we repeated this process; but instead of using known object center locations, we manually curated the set of images output from an anomaly detector run on a SAS dataset. Only images that were deemed to contain objects were selected.

The automatic masking network, which generates masks that segment the foreground and background, is trained in the context of counterfactual image generation. A counterfactual image, or rather an image with a counterfactual background, has the foreground taken from foreground image and the background taken from another background image. Following section 3, images with (foreground) objects X are assumed to have distribution μ. Similarly, the background-only images are denoted as $R \sim \nu$ (i.e., reference images).

The counterfactual image generator is denoted by $G^\theta : \mathcal{I} \times \mathcal{I} \to \mathcal{I}$, and consists of two modules: the masking network and the alpha blender. The masking network M^θ : $mathcall \to \mathcal{I}$ takes a SAS image as input and attempts to produce a mask containing *zeroes* at foreground pixel locations and *ones*

at background pixel locations. The alpha blending block takes mask M, image X, and background image R as input and generates new SAS images according to the equation

$$\alpha(M, X, R) = (1 - M) \circ X + M \circ R. \tag{7}$$

The counterfactual image $\tilde{X}$ is then defined as

$$\tilde{X} = G^{\theta}(X, R) = \alpha(M^{\theta}(X), X, R). \tag{8}$$

A block diagram of our approach is shown in Figure 1.

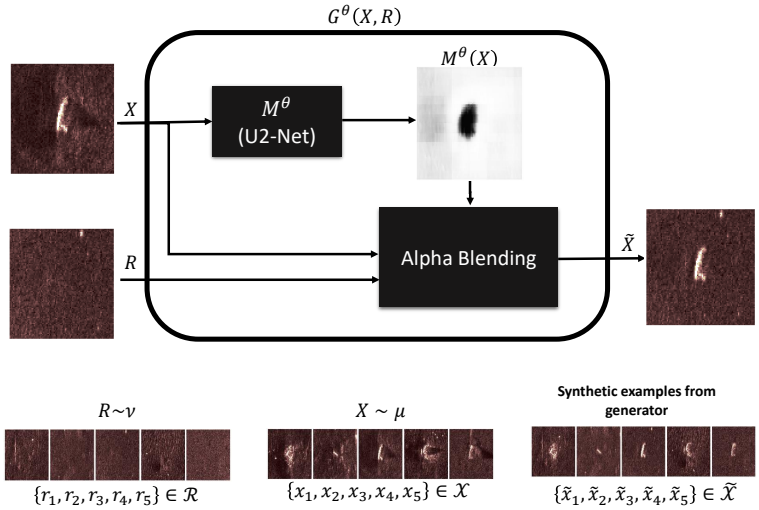

Figure 1: Block diagram of our approach. X, R, and $\tilde{X}$ denote the random variables, while $\mathbf{x}_m$, $\mathbf{r}_m$, and $\tilde{\mathbf{x}}_m$ denote the realizations of foreground, background, and counterfactual images respectively.

We chose the U2-Net[13] as the architecture for the masking network. The masking network's parameters θ are stochastically optimized using three losses. Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, $\mathcal{R} = \{\mathbf{r}_1, \ldots, \mathbf{r}_m\}$, and $\tilde{\mathcal{X}}^{\theta} = \{\tilde{\mathbf{x}}_1^{\theta}, \ldots, \tilde{\mathbf{x}}_m^{\theta}\}$ denote realizations of the foreground, background, and counterfactual images, where $\tilde{\mathbf{x}}_i^{\theta} = G^{\theta}(\mathbf{x}_i, \mathbf{r}_i) = \alpha(M^{\theta}(\mathbf{x}_i), \mathbf{x}_i, \mathbf{r}_i)$.

4.1 Background Swapping Loss

The primary loss function, background swapping loss, computes the divergence between empirical measures $\hat{\mu}$ and $\hat{\nu}^{\theta}$ formed from the batches of foreground $\mathcal{X}$ and counterfactual images $\tilde{\mathcal{X}}^{\theta}$.

$$\mathcal{L}_1(\theta) = \hat{D}_{DSW_2^c}(\hat{\mu}, \hat{\nu}^{\theta}) \tag{9}$$

Intuitively, this loss is minimized when the statistics of batch $\mathcal{X}$ match the statistics of batch $\tilde{\mathcal{X}}^{\theta}$ as closely as possible. Recall that the background seafloor images $\mathcal{R}$ were all collected from locations nearby the foreground object images. Assuming the *backgrounds* in the images from batches $\mathcal{X}$ and $\mathcal{R}$ are drawn from the same distribution, the perfect mask will allow the statistics of both foreground and background in $\mathcal{X}$ and $\tilde{\mathcal{X}}$ to match closely. However, this approach clearly allows for the degenerate solution for masking network parameters such that $\forall \mathbf{x} \in \mathcal{I}$, $\mathbf{x} = G^{\theta}(\mathbf{x}, \mathbf{r})$, where the underlying mask $M^{\theta}(\mathbf{x})_j = 0, \forall j \in \{1, \ldots, d\}$. That is, the masking network can minimize the loss by producing all zeros for all inputs. Thus, an additional loss is needed.

4.2 Foreground Subset Loss

The foreground subset loss encourages the counterfactual images to match the background (mask output of one) on all but a subset of pixels, where the ideal subset is those that correspond to the foreground. The estimated foreground subset is chosen to be all pixels in the mask with a value lower than the minimum of 0.5 and the q-th quantile of the mask pixels $\{M^\theta(\mathbf{x})_1, \ldots, M^\theta(\mathbf{x})_d\}$, where q is chosen by validation. This means a fraction of q pixels in each mask are not penalized, while $1-q$ are encouraged to be exactly one. For the i-th instance in the batch, this accomplished by minimizing the expected sum of squared errors (SSE) between $\tilde{\mathbf{x}}_i$ and $\mathbf{r}_i$ over all but that subset. Another alternative is to use a pseudo-mask label as all background and minimize the expected sum of the binary cross-entropy (BCE) for the mask pixels on all but that subset. We detail the the formulation of the SSE subset loss below, but BCE has been shown to work equally well.

$$\mathcal{L}_2(\theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j\in\mathcal{Q}_i}\left(r_{i,j} - \tilde{x}^\theta_{i,j}\right)^2, \tag{10}$$

$$\mathcal{Q}_i = \{j \in \{1,\ldots,d\} : M^\theta(\mathbf{x}_i)_j \geq \tau_i\}, \tag{11}$$

$$\tau_i = \min\{\text{quantile}(\{M^\theta(\mathbf{x}_i)_j\}_{j=1}^d, q),\ 0.5\}. \tag{12}$$

4.3 Background Loss

When the input to the masking network is background only images, we are sure that the mask output should indicate background for all pixels (mask output of one). This provides weak supervision to the masking network to avoid calling everything foreground. However, without the foreground subset loss, the network often learns to tell the difference between images with and without foreground. Thus, for known background images, we can simply minimize a loss between the mask pixels and all ones:

$$\mathcal{L}_3(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{d}\log(M^\theta(\mathbf{r}_i)_j). \tag{13}$$

4.4 Total Loss

The total loss is an unweighted combination of the losses,

$$\mathcal{L}_T(\theta) = \mathcal{L}_1(\theta) + \mathcal{L}_2(\theta) + \mathcal{L}_3(\theta). \tag{14}$$

The hyper-parameters underlying this loss are the choice of the optimization algorithm parameters underlying the distributional sliced divergence, the concentration parameter λ_C, the number of slices to take at each iteration n_w, and the quantile q.

4.5 Mask Network Optimization

The masking network hyper-parameters are the U2-Net architecture hyperparameters [13], the optimization algorithm parameters, the batch size m, and the image size. To simplify the approach, we selected the default U2-Net hyperparameters. To save computation, we only used the fused output layer from the U2-Net. We used the ADAM optimization algorithm with an initial learning rate of 10^{-3}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. Furthermore we used cosine learning rate decay, with the learning rate being decayed each epoch for 10 epochs. The batch size was selected to be 512 and the images were downsampled to 90×90 pixels. This approach requires rather large batch sizes to compute accurate divergences; the smaller image sizes were chosen to allow faster computation and iteration time. However, larger (non-downsampled) image sizes are feasible and will be used in future work.

5 RESULTS

Figure 2 shows example images from a fully trained counterfactual generator G^{θ}. Eleven foreground images were randomly selected (first row) from the test set and input to the generator. The second row shows the generated masks and the third row shows $(1 - M(\mathbf{x}))\mathbf{x}$. The fourth row shows randomly selected counterfactual backgrounds, and the fifth row shows the alpha-blended image generated using foreground image, mask, and counterfactual background.

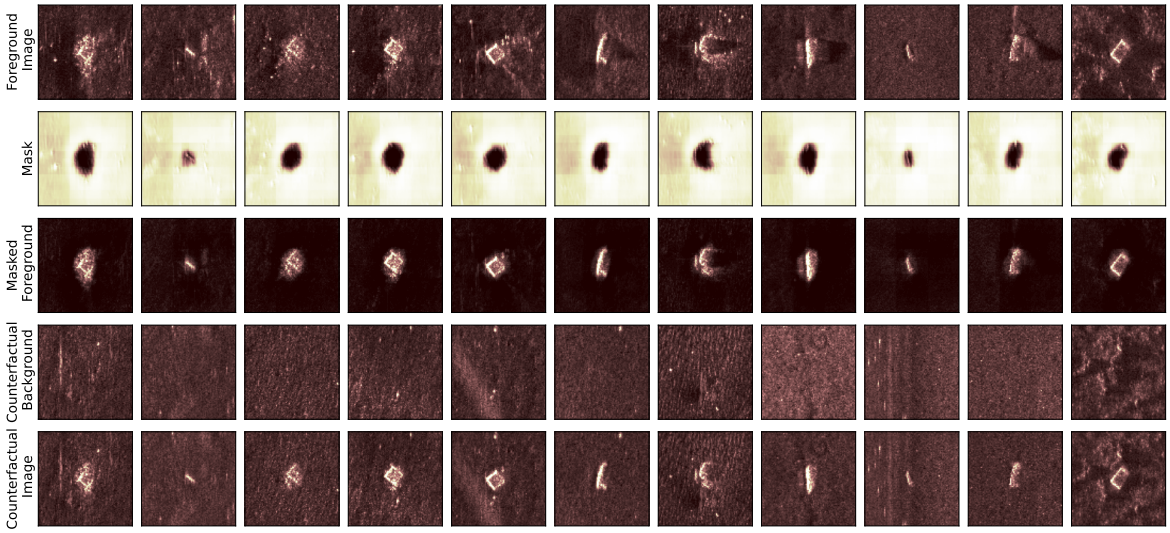

Figure 2: Counterfactual Examples

Figure 3 compares the masks generated by the masking network with baseline masks produced by simply thresholding the pixels. The first row contains the same randomly selected foreground images. The second row shows threshold masks produced by thresholding pixels at 0.15 times the max pixel value. The threshold was selected qualitatively so that most of the foreground was contained in the mask while the amount of seafloor background was minimized. The third and fourth rows compare the results of masking the data using the threshold and the masking network masks respectively. As expected, a threshold mask that contains most of the object also brings over much of the (seafloor) background as well.

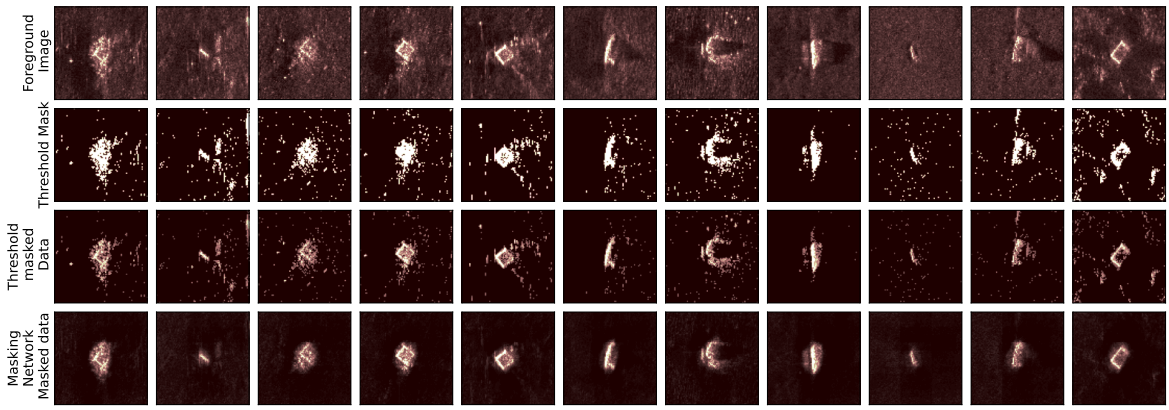

Figure 3: Threshold baseline

6 CONCLUSION

We have proposed a novel weakly supervised approach to generate pixel-wise masks for SAS imagery using distributed sliced Wasserstein distance. We demonstrated this approach on example images and showed that the approach reliably generates tight masks around the objects.

There are a number of directions for future work. We intend to compare the advantages and disadvantages of additional loss functions not discussed in this paper. Further, we intend to label a small subset of the SAS data pixelwise as object and seafloor so that we can better quantify our results. Finally, we will investigate using the counterfactual images to augment existing machine learning datasets.

7 ACKNOWLEDGEMENT

The authors would like to express gratitude to the Office of Naval Research for funding this research. Research at the University of Delaware was sponsored by the Department of the Navy, Office of Naval Research under ONR award number N00014-21-1-2300. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research.

8 REFERENCES

[1] Tal Remez, Jonathan Huang, and Matthew Brown. Learning to segment via cut-and-paste. In *Proceedings of the European conference on computer vision (ECCV)*, pages 37–52, 2018.

[2] Adam Bielski and Paolo Favaro. Emergence of object segmentation in perturbed generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Yu Yang, Hakan Bilen, Qiran Zou, Wing Yin Cheung, and Xiangyang Ji. Learning foreground-background segmentation from improved layered gans. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2524–2533, 2022.

[4] Pedro Savarese, Sunnie SY Kim, Michael Maire, Greg Shakhnarovich, and David McAllester. Information-theoretic segmentation by inpainting error maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4029–4039, 2021.

[5] Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced Wasserstein generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3713–3722, 2019.

[6] Ishan Deshpande, Ziyu Zhang, and Alexander G Schwing. Generative modeling using the sliced Wasserstein distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3483–3491, 2018.

[7] Soheil Kolouri, Gustavo K Rohde, and Heiko Hoffmann. Sliced Wasserstein distance for learning Gaussian mixture models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3427–3436, 2018.

[8] Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander G Schwing. Max-sliced Wasserstein distance and its use for GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10648–10656, 2019.

[9] Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. Generalized sliced Wasserstein distances. In *Advances in Neural Information Processing Systems*, pages 261–272, 2019.

[10] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

[11] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

[12] Khai Nguyen, Nhat Ho, Tung Pham, and Hung Bui. Distributional sliced-Wasserstein and applications to generative modeling. In *International Conference on Learning Representations*, 2021.

[13] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R. Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106:107404, oct 2020.