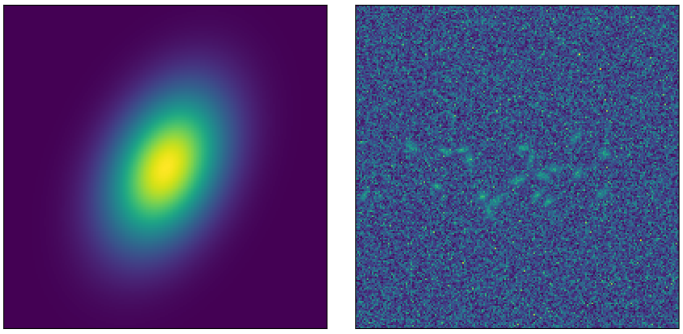
Supplementary Materials

CONTENTS

I. SIMULATE THE EXTENDED TARGETS IN THE RAYLEIGH-DISTRIBUTED CLUTTER

In the paper, Onet model is compared with CFAR for extended target segmentation in the Rayleigh-distributed clutter. We simulate Rayleigh-distributed clutter in frames and embed multiple extended targets (



(a) target template (b) simulated frame

Fig. 1: Embed multiple 2d-Gaussian-distributed extended targets in Rayleigh-distributed clutter. (a) One particular target template in normalized form (intensities range in [0,1]). (b) Twenty targets are embedded in the simulated clutter with PSNR=2.

(see Fig.1) with varied peak-signal-to-ratio (PSNR). Target template adopts 2d elliptical Gaussian function $g(x, y; \boldsymbol{x}_n)$ to imitate the energy diffusion of the ship's echoes in marine radar signal as discussed in [Zhou et al., 2022; Jiang et al., 2017]. Here $\boldsymbol{x}_n = (x_n, y_n)$ defines the center position of the n-th target. Let w and h represent the target's width and height, the 2d positions inside the target are limited in $x_n - w/2 \leq x \leq x_n + w/2$ and $y_n - h/2 \leq y \leq y_n + h/2$. Their intensities (range in (0,1]) have the form:

$$g(x, y; \boldsymbol{x}_n) = a_n \exp\left(-\alpha (x - x_n)^2 + 2\beta (x - x_n)(y - y_n) + \gamma (y - y_n)^2\right), \tag{1}$$

where

$$\begin{aligned} \alpha &= \cos^2\theta/2\sigma_x^2 + \sin^2\theta/2\sigma_y^2, \\ \beta &= \sin 2\theta/4\sigma_y^2 - \sin 2\theta/4\sigma_x^2, \\ \gamma &= \sin^2\theta/2\sigma_x^2 + \cos^2\theta/2\sigma_y^2. \end{aligned} \tag{2}$$

σ_x and σ_y denote the standard derivation on the two axes of the elliptical Gaussian. θ means the clockwise rotation angle of the ellipse. a_n denotes the peak amplitude and locates in the center of the 2d target. Assume that clutter has constant power σ_c and the average power of the peak amplitude $\sigma_p = E\left(a_n^2\right)$ over frames. $E(\cdot)$ represents the expectation of a random variable. The PSNR is defined by

$$\text{PSNR} = 10 \log_{10} \frac{\sigma_p}{\sigma_c}. \tag{3}$$

Let the target be the Swerling type 0, which is free of fluctuating and constant over frames [Swerling, 1960], then a_n is equal to $\sqrt{\sigma_p}$. Therefore given a PSNR, the coefficient of the target template is computed by

$$a_n = \sqrt{\sigma_c 10^{\text{PSNR}/10}}. \tag{4}$$

To merge the target with the clutter, the pixels of the target template with weaker intensities than those of the clutter are covered by the background. So the intensity of the pixel inside the n-th target region is given by:

$$z_n(x, y) = \begin{cases} a_n g(x, y) + c(x, y), & \text{if } a_n g(x, y) > c(x, y) \\ c(x, y), & \text{otherwise} \end{cases} \tag{5}$$

Here $c(x, y)$ denotes a sample of the Rayleigh-distributed background. Based on the (4) and (5), we simulated extended targets with 11 PSNRs in the range $[0, 10]$. For each PSNR, we kept the scale parameter

of Rayleigh distribution as 1 which made $\sigma_c = 2$ over 150 frames. In each frame, 20 targets scattered in random positions and random widths (95% range in [8, 14] pixels and σ_x is around $w/4$) and heights (95% range in [14, 22] and σ_y is around $h/4$). Therefore in the simulated dataset, it got 1650 frames and 33000 labeled targets from the extremely low PSNR=0 to the high PSNR=10.

II. Ensemble two Onet-TW to segment the weak targets

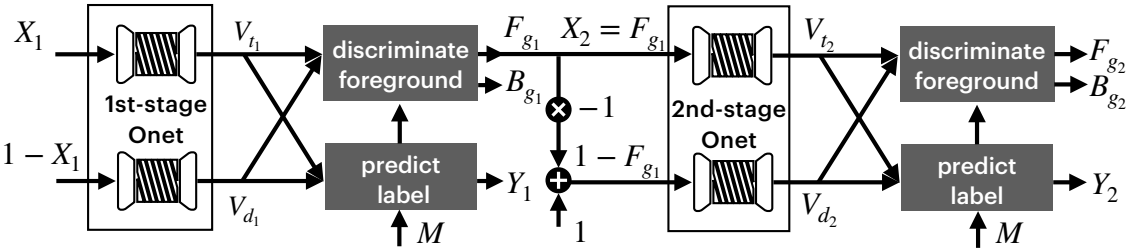


Fig. 2: The architecture of two-stage Onet ensemble

Onet-TW's characteristic of denoising in low PSNR frame and good performance in high PSNR inspires us to ensemble two Onets to address the target segmentation in two stages. Fig. 2 illustrates the two-stage architecture (Onet2). First-stage Onet-TW is trained on the dataset only with low PSNR (0, 1, and 2), while the second-stage Onet-TW is trained on the high PSNR (≥ 5) dataset. Here we utilize the first-stage Onet-TW to denoise the input X_1 and then send forward the foreground feature map F_{g_1} with improved SNR to the next Onet-TW as input X_2. The second Onet-TW can greatly depress the false alarms as it does in the last sub-figure of Fig. 2 of the paper where PSNR=10. It is noteworthy that the output V_t and V_d are sent to the foreground discriminating part. When the predicted label Y has greater accuracy in matching with the ground truth mask M than its reversion $1 - Y$, V_d represents the foreground heatmap F_g. Otherwise, V_t does. This is because V_t and V_d are concatenated in the 0 and 1 channels respectively. After softmax and argmax of the concatenated tensor, locations in Y where $V_d > V_t$ will be marked as 1 (target pixels).

The visual effect of the two-stage Onet ensemble is demonstrated in Fig. 3. Compared to the clutter frame of X_1, the first-stage Onet predicts the foreground F_{g_1} correctly with improved SNR. Fed with this F_{g_1}, the second-stage segmentation Y_2 shows cleaner background than Y_1. It sharply decreases the false alarm rate without missing the peak region of the targets.

Fig.4 further compares the two-stage Onet-TW ensemble with CFAR. it shows that Onet2 has a cleaner background and more target labels than CFAR when they both work at $P_{fa} = 0.03$. Table I lists the

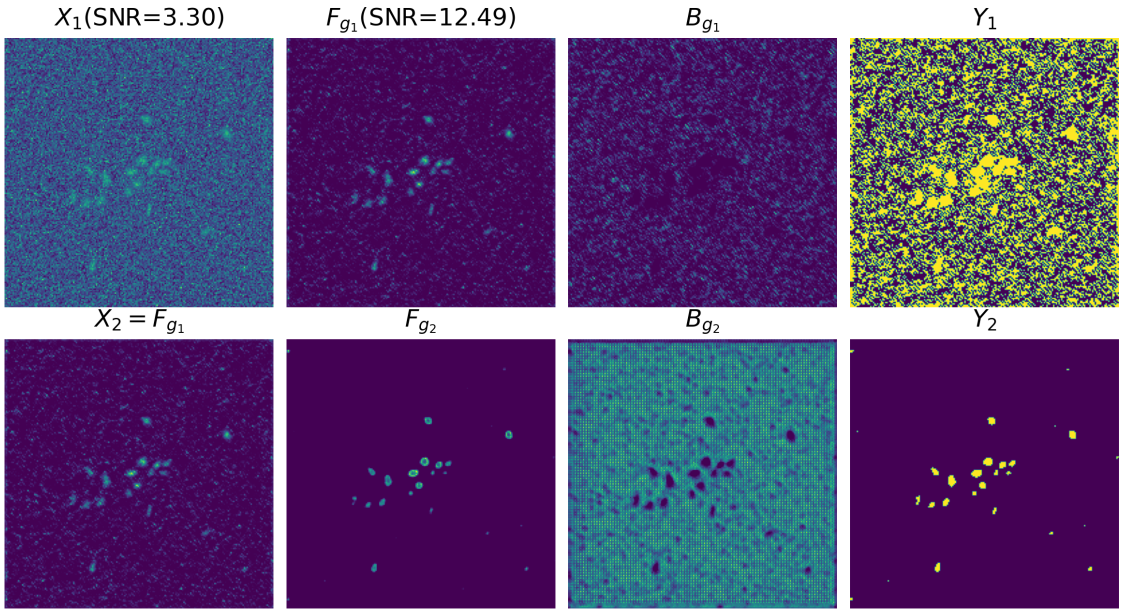


Fig. 3: Demonstration of extended target segmentation via two-stage Onet. Input frame of the first stage Onet is denoted as X_1 with SNR=3.3, which is composed of Rayleigh clutter and multiple 2d-Gaussian-distributed targets with pre-setting PSNR=2. In the first row, F_{g_1} and B_{g_1} denote the score map of the foreground and background, respectively. Y_1 is the predicted target label in the first stage. In the second row, the second-stage Onet's input $X_2 = F_{g_1}$ is with improved SNR. The outputs of the 2nd stage are named F_{g_2}, B_{g_2}, and Y_2 respectively.

detailed results for both methods. Note that accuracy metrics are calculated for both true negatives and true positives. Since the background region accounts for the majority, when the false alarm rate is set to a low value (e.g., $P_{fa} \approx 0.001$ or 0.0001), pixels are likely to be marked as background by CFAR, i.e., the segmented negative will be close to 100%. Therefore, the overall accuracy is biased toward the accuracy of background segmentation. The overall accuracy metrics of both methods are high. However, the low false alarm rate setting in CFAR weakens the detection rate of targets, which is the main metric for verifying target segmentation. For the same false alarm rate, the proposed Onet2 has a better detection rate than CFAR by a large margin. In the Fig.4 it is shown that the central part of the target is correctly marked by Onet2, but the boundary is corrupted in the low PSNR frames. The reason is that the strong background clutter absorbs some weak pixels in the target boundary region. However, the ground truth mask is measured directly in the target template without considering the clutter strength.

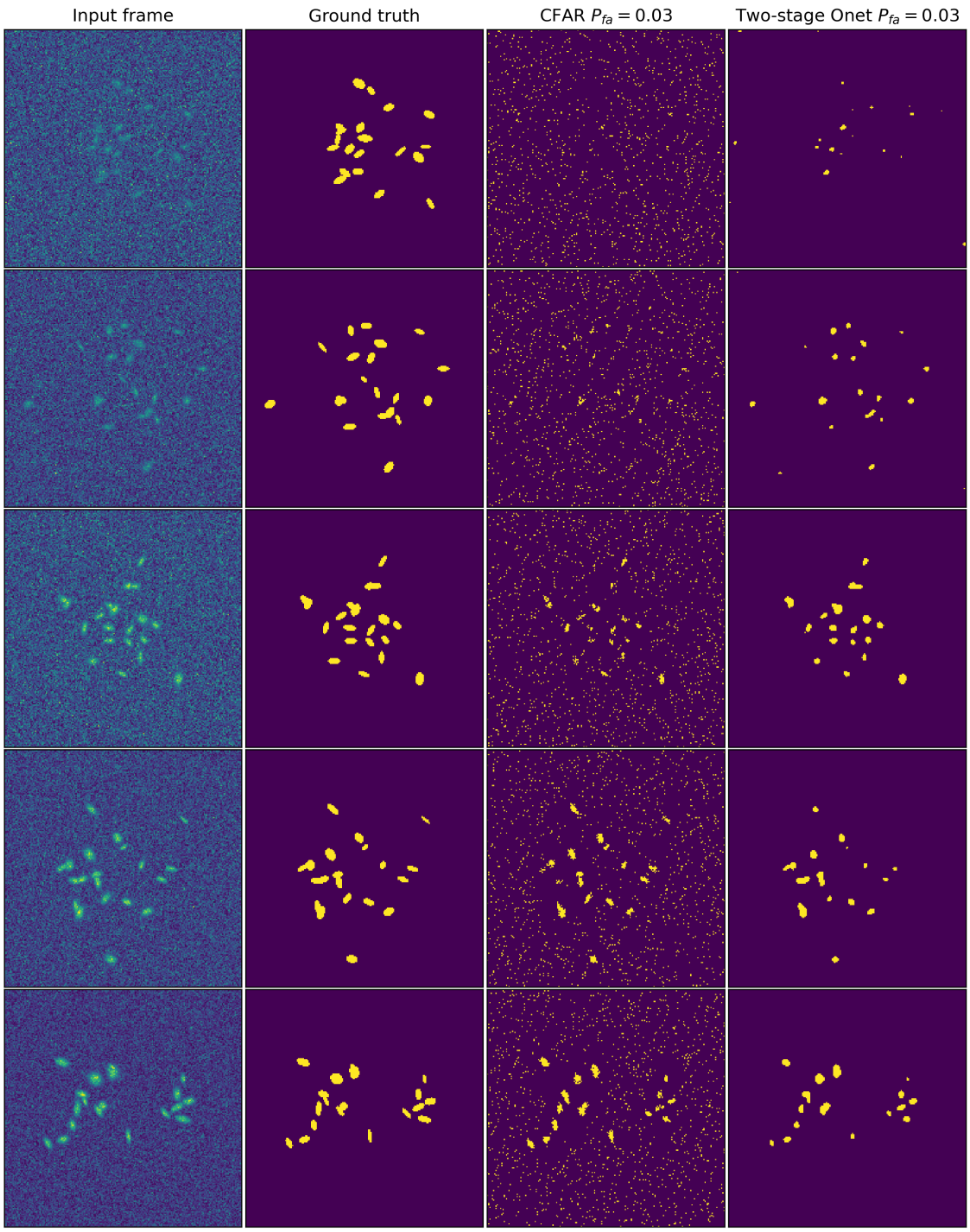


Fig. 4: Comparison of extended target segmentation in Rayleigh-distributed clutter between CFAR and Onet2. The first column shows the input frames with PSNR setting of 1, 3, 5, 7, 9. The second column gives the true labels of the target. For fair comparison, CFAR is tuned to has the same false alarm rate $P_{fa} = 0.03$ as the Onet2.

TABLE I: Comparison of CFAR and two-stage Onet ensemble (Onet2) in the simulated clutter for different PSNRs

PSNR	Model	OA ↑	mIoU ↑	$\mathbf{P_d}$ ↑	$\mathbf{P_{fa}}$ ↓
0	CFAR	0.9778	0.4890	0.0003	0.0013
	Onet2	0.9780	0.5032	0.0306	0.0017
1	CFAR	0.9779	0.4890	0.0002	0.0012
	Onet2	0.9813	0.5765	0.1856	0.0017
2	CFAR	0.9778	0.4890	0.0002	0.0011
	Onet2	0.9870	0.7020	0.4399	0.0012
3	CFAR	0.9787	0.4894	0.0000	0.0001
	Onet2	0.9889	0.7339	0.4830	0.0002
5	CFAR	0.9793	0.4898	0.0002	0.0002
	Onet2	0.9924	0.8120	0.6376	0.0002
7	CFAR	0.9791	0.4904	0.0017	0.0002
	Onet2	0.9921	0.8056	0.6223	0.0001
9	CFAR	0.9796	0.4916	0.0036	0.0001
	Onet2	0.9910	0.7765	0.5652	0.0001

III. Data augmentation and pre-processing in ZY3 experiment

In order to close the gap between the unsupervised Onet and the pretrained UDA [Guo et al., 2022] , we tried data cleaning and similar data augmentation in UDA to boost the performance. Specifically, we adapted multiple types of data augmentation on the training samples, which includes image flipping, rotating, perspective transforming, grid distortion and random brightness contrast (we name it train_augmentation). Inspired by the winner blog of cloud cover detection competition for Sentinel-2 imagery[1], we split the training and testing dataset into multiple sets with different distributions and monitor the Onet's performance on different sets. In the experiment, we divided the testing sets into 3 categories by observing the RGB-format images and ground truth labels: (1) normal cloud image without snow/ice, (2) cloud image with snow/ice and (3) thin cloud image. And we noticed that even for the experts marked thin cloud mask, the standard is not unified. Some of them with very shallow intensities are marked out in certain images. However, some thin clouds with heavier intensities are ignored in other images. Therefore, in the training set without ground truth labels, we conservatively divided the training dataset into: (1) cloud image without snow/ice (containing the personal visually confirmed thin cloud), (2) cloud image with snow/ice, (3) terrain image without clouds. We finds that excluding the 3rd type terrain image without clouds in train set, onet can be trained more smoothly and achieves 94% accuracy on 1st type testing set (cloud image without snow/ice), 80% accuracy on 2nd-type cloud image with snow/ice and 85% accuracy on the 3rd-type thin cloud image in the testing set.

To further enable the ability of Onet to distinguish the snow/ice and clouds, we transplanted the

[1] https://drivendata.co/blog/cloud-cover-winners/

segmented clouds from normal cloud images to the snow/ice terrain only images. Fine tuning the Onet on the synthetic cloud datasets with labels in supervised way improves the performance on the cloud images with snow/ice in test set from 80% to 90%. However, the performance on thin-cloud images are dropped, the overall accuracy are around 0.8984.

Viewing the thin cloud image, we found that the remotely sensed thin cloud looks like fog captured by the ordinary camera. Hence, reversely applying the haze removal algorithm with dark channel [He et al., 2011], we can enhance the thin cloud's intensity by adding its reflected energy from atmospheric light following the equation:

$$I(x) = J(x)t(x) + A(1 - t(x)), \tag{6}$$

where I denote the observed intensity in spatial position x, J is the scene radiance with dark channel energy, A is the global atmospheric light, and t is the medium transmission coefficients. In remote sensing images, the energy of $J(x)t(x)$ is generated by the dark terrain and $A(1-t(x))$ is mainly from the white clouds. Following the algorithm of dark channel computing [He et al., 2011], J and t can be estimated for every spatial points. Given an observed image I, the light reflected by the cloud (A) can be computed by (6). Then multiply a coefficient k ($k > 1$) to A, the power of the thin cloud would be enhanced (we call it cloud-enhancing) by $I + kA$. While for the snow scene, the energy from the sparse white snow embedded in darker terrain can be weakened by filtering the image with block-wise dark channel and computing the terrain radiance J (we call it snow-removal pre-processing). The performance of Onet with the accumulated processing and other backbones are summarized in following Table II.

TABLE II: Performance of cloud segmentation in ZY3 cloud thumbnails.

Method	**OA** ↑	**mIoU** ↑
Deeplabv2*[Chen et al., 2018]	0.8399	0.6730
Onet-CNN1.0 (previous version)	0.8602	0.6902
Onet-CNN2.1(train_augmentation)	0.8812	0.7190
Onet-CNN2.2(train_set splitting and augmentation)	0.8984	0.7386
Onet-CNN2.3(train_augmentation, cloud_enhancing and snow_removal)	**0.9210**	0.7823
Onet-HRNet	0.9223	0.7885
Onet-ConvNeXt	**0.9250**	0.8010
Onet-TransUnet	0.9243	0.7981
Onet-SwinUnet	0.9247	0.8011
UDA*[Guo et al., 2022]	0.9127	**0.8216**
Onet-CNN2.2*(upper bound of the backbone: supervised training on the labelled test set)	0.9376	0.8467

The asterisk '*' indicates the method uses supervised training or pre-training.

IV. Nucleus segmentation in Microscope images

The 2018 Data Science Bowl [Caicedo et al., 2019] was an open competition on Kaggle with the aim to find the best algorithm of nuclei segmentation in microscopy images. It contained a diverse set of light microscopy images across different tissues, cell lines, imaging conditions, staining protocols and instruments. The competition provided a limited training set of images (670) with annotated masks for the nuclei. While the second test stage consists of 15 diverse image sets (more than 3000 images) from different biological experiments which were not seen in the training set. It is a typical and challenging platform to test the binary semantic segmentation.

Since the real masks of the second stage test is not public, participants have to upload the specified mask files to get the final testing score from the Kaggle server (https://www.kaggle.com/c/data-science-bowl-2018). In Table III, we list the scores of 4 methods. The Infoseg method scores the least. This is because the types of nuclei are diverse. It is hard to model the global features in two categories. A single U-Net model is trained on the training set via supervised learning and evaluated on the second-stage testing set. Onet scores better than the supervised U-Net. The main reason is that the annotated training set is too small to capture all the variations of nuclei. The supervised single U-Net can not generate reliable segmentation in the second-stage test. While Onet training does not need the annotated masks, it is trained directly on the second stage testing set. It is reported that the best-performing team 'Topcoders' ensembles eight CNN architectures (Six of them adopt U-Net structure) via supervised training and additional augmented training-set [Caicedo et al., 2019].

In Fig. 5 and 6 we show the 'positive' prediction of foreground nuclei, background and the mask. The quotation mark of 'positive' means it is positive in subjective visual effect (the real masks are not published). Obviously negative results are given in Fig. 7.

TABLE III: Nucleus segmentation scores

Method	**Score**	**Rank**(742 teams)
Infoseg	0.02437	693
Supervised U-Net	0.21124	654
Onet	0.24247	639
Topcoders	0.63164	1

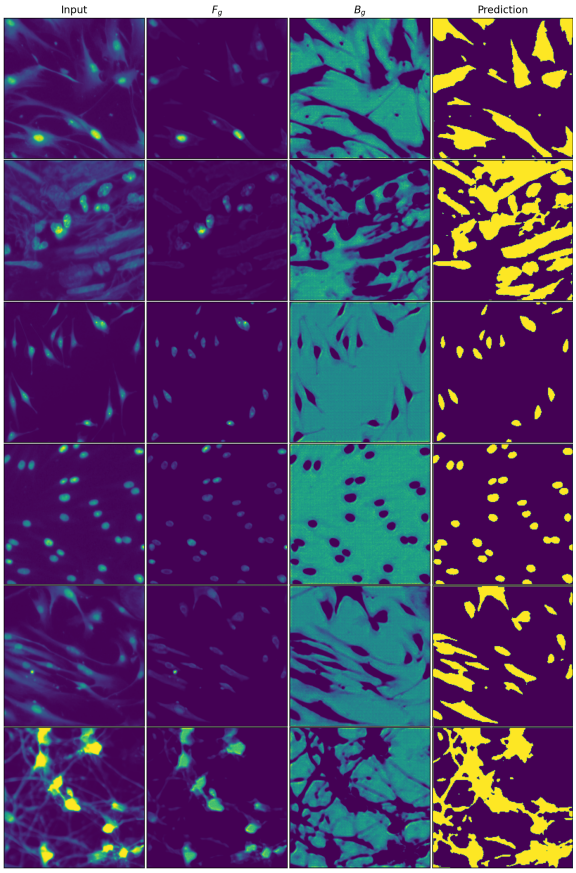


Fig. 5: Group 1 of the 'positive' nucleus segmentation. Input frames are transformed to gray channel. F_g (foreground) denotes the prediction map of the nuclei. B_g means the clutter background. Predicted mask is generated by argmax on F_g and B_g.

V. Ablation Study On Selection Of Training Dataset For Rayleigh-Distributed Clutter.

In Table IV we train the Onet on different training sets with varied PSNRs and test the performance on the datasets of low PSNRs (0-5), high PSNRs (5-10), and all PSNRs (0-10). The training sets are divided into three parts which divide all PSNRs with intervals 0, 2, and 5 respectively (two horizontal lines divide these three parts in Table IV). When training the model on low PNSRs (≤ 5), Onet obtains very high P_d ($\geq 92\%$) on all types of test sets, regardless of the choice of PSNR interval. It reflects Onet's great advantage in target segmentation, especially in the low SNR scenario. It is worth noting that in training with increasing PSNRs, the model's OA increases at the cost of a decrease in P_d for test at low PSNRs (0-5). This means a model trained on high PSNRs is likely to treat weak targets as clutter. While testing the model on the high and all PSNRs, the model trained on low PSNRs still scores high in P_d. For three

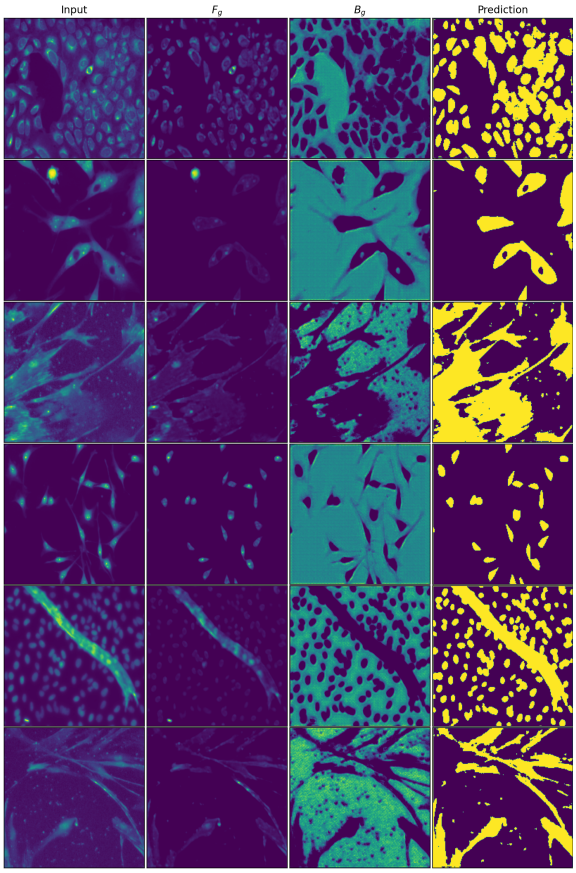


Fig. 6: Group 2 of the 'positive' nucleus segmentation. It is worth noting that the nuclei have very different shapes from image to image. Onet successfully filters them out of the background. The fine boundaries of Fg mean that Onet will be a good assistant for some advanced applications to label microscope images automatically.

intervals (0, 2, and 5), the training sets which include PSNR=2 obtain the maximum number of the best metrics. Since all tests keep P_{fa} in the order of 10^{-1}, we choose the model trained on PSNRs in [0,2] (with the least P_{fa} and $P_d \geq 95\%$ in three test sets.) as the basic model for the first-stage Onet. The first-stage Onet denoises the clutter and improved the SNR for the targets. In the second-stage, we choose the Onet trained on PSNRs in [5-10] where targets are clean and the background is more smoother.

VI. Ablation Study on High-Dimensional Image-Level Feature Versus Shallow Dense Feature

In Infoseg, it enhances more on the image-level global features which are represented by high-dimensional vectors. In our proposal, we utilize the shallow dense global features from the decoding end of U-Net. In this ablation, we test the performance of Onet with two different choices of global features. Onet-Infoseg

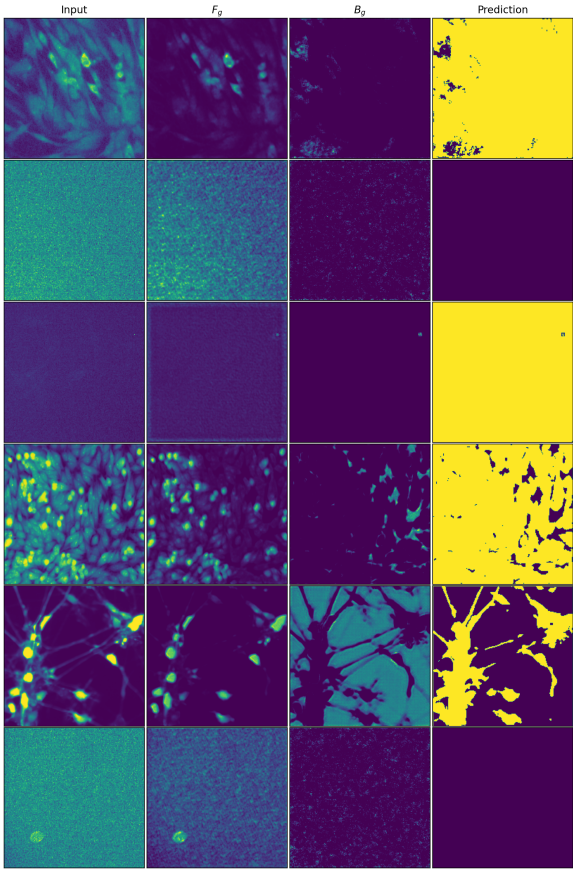


Fig. 7: The negative result of nucleus segmentation. Uniformly distributed noises caused strong activation on the foreground or background maps. In rows 1, 2, 3, 4, and 6, they lead to extensive single labeling (e.g., in row 2, the predicted labels are all 0). In row 5, we were unable to determine whether the translucent tissue belonged to the nucleus.

chooses the Infoseg as the backbone, which represents the local and global feature in high-dimensional vector (1024 dimensions). By contrast, Onet utilizes the shallow local feature L and global feature H (64 dimensions), which are extracted from the initial encoding part and from the end of the decoding path of U-Net, respectively.

In Table V, we compare performance of the Onet-Infoseg and Onet in twins mode in the low PSNR ($\leq$ 2) scenario and in weight-sharing (WS) mode for other datasets. In the low PSNR ($\leq$ 2) scenario, Onet-Infoseg seems to be better in OA, mIoU and P_{fa}. However, in fact the Onet-Infoseg fails to learn the feature of the real targets when they are immersed in strong clutter. In the first two rows of Fig. 8, we observe that Onet-Infoseg's prediction and foreground score can not tell the difference between strong clutter and weak target. By contrast, Onet's foreground score denoises the input effectively. When PSNR raising, Onet-WS scores higher than Onet-Infoseg in all metric except the P_d in the datasets PSNR 5-10

TABLE IV: Ablation study on the training dataset

Training-set PSNRs	Test on low PSNRs (0-5)				Test on high PSNRs (5-10)				Test on all PSNRs 0-10				Best counts
	OA ↑	**mIoU** ↑	$\mathbf{P_d}$ ↑	$\mathbf{P_{fa}}$ ↓	**OA** ↑	**mIoU** ↑	$\mathbf{P_d}$ ↑	$\mathbf{P_{fa}}$ ↓	**OA** ↑	**mIoU** ↑	$\mathbf{P_d}$ ↑	$\mathbf{P_{fa}}$ ↓	
0	0.5531	0.2934	0.9609	0.4555	0.8310	0.4709	0.9996	0.1726	0.7027	0.3816	0.9785	0.3031	0
1	0.5586	0.2965	**0.9630**	0.4499	0.8335	0.4718	0.9997	0.1700	0.7040	0.3816	**0.9796**	0.3019	2
2	0.5752	0.3056	0.9567	0.4329	**0.8343**	**0.4726**	0.9997	**0.1692**	**0.7129**	**0.3870**	0.9762	**0.2927**	6
3	0.5624	0.2984	0.9544	0.4459	0.8146	0.4560	0.9995	0.1893	0.6968	0.3770	0.9752	0.3091	0
4	0.5660	0.3003	0.9485	0.4421	0.8050	0.4492	0.9991	0.1991	0.6926	0.3743	0.9708	0.3133	0
5	0.5338	0.2826	0.9578	0.4752	0.7753	0.4278	0.9994	0.2294	0.6629	0.3565	0.9763	0.3437	0
6	0.5498	0.2753	0.0211	0.4389	0.7043	0.3821	0.9999	0.3019	0.5863	0.3124	0.9756	0.4219	0
7	0.6333	0.3169	0.0092	0.3534	0.6196	0.3317	**1.0000**	0.3884	0.5324	0.2744	0.4939	0.4668	1
8	0.6947	0.3475	0.0044	0.2906	0.5452	0.2877	0.8780	0.4618	0.5696	0.2853	0.0291	0.4189	0
9	0.7459	0.3730	0.0022	0.2384	0.5381	0.2713	0.1463	0.4536	0.6315	0.3158	0.0013	0.3551	0
10	**0.7841**	**0.3921**	0.0013	**0.1993**	0.6019	0.3010	0.0000	0.3854	0.6845	0.3423	0.0007	0.3010	3
0-2	0.5757	0.3060	**0.9623**	0.4325	**0.8421**	**0.4786**	0.9998	**0.1612**	**0.7173**	**0.3899**	0.9793	**0.2883**	7
1-3	0.5672	0.3009	0.9379	0.4406	0.7873	0.4361	0.9982	0.2172	0.6836	0.3686	0.9648	0.3224	0
2-4	0.5743	0.3048	0.9349	0.4333	0.7914	0.4390	0.9979	0.2130	0.6905	0.3727	0.9634	0.3153	0
3-5	0.5680	0.3012	0.9368	0.4399	0.7844	0.4340	0.9981	0.2201	0.6842	0.3689	0.9641	0.3217	0
4-6	0.5429	0.2875	0.9472	0.4656	0.7630	0.4196	0.9989	0.2419	0.6602	0.3549	0.9706	0.3464	0
5-7	0.5225	0.2624	0.0764	0.4681	0.7060	0.3830	0.9997	0.3002	0.6005	0.3205	**0.9828**	0.4076	1
6-8	0.5906	0.2956	0.0161	0.3972	0.6354	0.3409	**1.0000**	0.3723	0.5409	0.2835	0.7567	0.4636	1
7-9	0.6505	0.3255	0.0079	0.3359	0.5652	0.3008	**1.0000**	0.4439	0.5373	0.2704	0.1103	0.4537	1
8-10	**0.7038**	**0.3520**	0.0038	**0.2814**	0.5280	0.2702	0.3902	0.4691	0.5976	0.2988	0.0021	0.3898	3
0-5	**0.5900**	**0.3134**	0.9237	0.4170	**0.7986**	**0.4444**	0.9975	**0.2056**	**0.7018**	**0.3794**	0.9570	**0.3036**	8
1-6	0.5766	0.3060	0.9292	0.4308	0.7866	0.4356	0.9981	0.2179	0.6885	0.3714	0.9609	0.3173	0
2-7	0.5591	0.2964	**0.9380**	0.4489	0.7717	0.4256	0.9987	0.2330	0.6734	0.3625	0.9657	0.3328	1
3-8	0.5196	0.2744	0.9324	0.4891	0.7371	0.4025	0.9995	0.2685	0.6356	0.3405	0.9749	0.3716	0
4-9	0.5178	0.2601	0.0791	0.4729	0.7026	0.3809	0.9998	0.3037	0.6010	0.3208	**0.9819**	0.4070	1
5-10	0.5780	0.2894	0.0181	**0.4102**	0.6411	0.3442	**1.0000**	0.3664	0.5456	0.2876	0.8470	0.4608	2

For each test, the best metrics are marked in bold blue for the training set with same PSNR interval. (i.e. the best in the region separated by the solid lines for PSNR interval= 1, 2, and 5 in training set.)

and rain clutter. The reason is that, Onet-Infoseg gets less spatial granularity. It tends to predict bigger region on the small-size targets. Therefore, the detection rate is increasing at the sacrifice of higher false alarm rate P_{fa}.

TABLE V: Performance comparison between Onet-Infoseg and Onet in different modes

Dataset: Simulated clutter with PSNR 0-2

Model	**OA** ↑	**mIoU** ↑	$\mathbf{P_d}$ ↑	$\mathbf{P_{fa}}$ ↓
Onet-Infoseg	0.6895	0.3650	0.8705	0.3140
Onet-TW	0.5490	0.2871	0.9018	0.4570

Dataset: Simulated clutter with PSNR 5-10

Model	**OA** ↑	**mIoU** ↑	$\mathbf{P_d}$ ↑	$\mathbf{P_{fa}}$ ↓
Onet-Infoseg	0.9746	0.6690	0.8477	0.0233
Onet-WS	0.9867	0.7389	0.6198	0.0072

Dataset: Rain clutter

Model	**OA** ↑	**mIoU** ↑	$\mathbf{P_d}$ ↑	$\mathbf{P_{fa}}$ ↓
Onet-Infoseg	0.9975	0.7875	0.6835	0.0009
Onet-WS	0.9976	0.7978	0.6388	0.0005

Dataset: ZY3

Model	**OA** ↑	**mIoU** ↑	$\mathbf{P_d}$ ↑	$\mathbf{P_{fa}}$ ↓
Onet-Infoseg	0.8958	0.7226	0.7362	0.0779
Onet-WS	0.9208	0.7842	0.7852	0.0635

TABLE VI: Complexity comparison between Onet-Infoseg and Onet in twins mode.

Model	Parameters	FLOPs
Onet-Infoseg	423.6M	16.9G
Onet-TW	62.1M	83.6G

In Table VI, we list the parameters, FLOPs of the architecture with different global features. More layers of downscale and upscale convolutions in U-Net makes the Onet-TW consumes more inference time than Onet-infoseg. While the high-dimensional global features and the alignment with local dense features in Infoseg costs Onet-Infoseg more graphical memories (much more parameters are needed). Onet-TW with dense global features is better than Onet-Infoseg in overall segmentation performance.

In Fig. 8 when compare the foreground and background score maps of these two methods, Onet-Unet highlighted the meaningful target region in the foreground score and Onet-Infoseg pays more attention on the repetitive texture patterns. We analyze that Infoseg enhances more on the image-level global features. If the small targets are immersed in strong clutter (PSNR ≤ 2), the global feature of Infoseg learns the whole texture information of the image but ignores the foreground targets. When the PSNR increases (≥ 5), we notice that Onet-Infoseg becomes less sensitive to the spiky noise and improves the accuracy of segmentation. By contrast, Onet-Unet gets a clearer foreground maps and maintains better granularity on the target regions.

VII. Comparison with the pre-trained segmentation models

SAM1 used a transformer model to establish a relationship between the representation of image patches (pre-trained ViT [Dosovitskiy et al., 2021] by MAE [He et al., 2022]) and the annotated masks through a cross-attention mechanism. The initial masks in the training images were manually annotated. The Meta's research team followed the idea of the "scaling law" [Kaplan et al., 2020], using a large-parameter ViT model combined with a massive annotated dataset (11 million images and 10 billion masks) and trained the network using a supervised learning paradigm with massive GPUs. During the process of improving annotation quality and speed, SAM1 incorporated a user-interactive segmentation strategy. By embedding user input, such as click locations, selection boxes, and refined masks, as prompts into the encoded image output, SAM1 associates these inputs with the final segmentation mask. This strategy expands the network's segmentation capabilities by aligning with users' attentions. SAM2 builds upon SAM1 by introducing object segmentation in video. Its core strategy remains the same: first, manually

TABLE VII: Performance comparison with SAM2 in different model size. The rows with gray background color are unreliable for its zero P_d or P_{far}. The best scores are marked in blue, while the second are marked in green. Pre-trained SAM2-small works best in simulated datasets. Guided-SAM2 improves SAM2's performance in the last two datasets.

Dataset: Simulated clutter with PSNR 0-2

Model	OA ↑	mIoU ↑	P_d ↑	P_{fa} ↓
Onet	0.9548	0.5010	0.1369	0.0314
SAM2-tiny	0.9834	0.4917	0	0
Guided-SAM2-tiny	0.9834	0.492	0.0006	0
SAM2-small	**0.9863**	**0.6300**	**0.3237**	0.0026
Guided-SAM2 _small	**0.9855**	**0.6239**	**0.3327**	0.0035
SAM2-base	0.9826	0.4914	0.0003	**0.0009**
Guided-SAM2-base	0.9824	0.4954	0.0135	**0.0013**
SAM2-large	0.9834	0.4921	0.0007	0
Guided-SAM2-large	0.9834	0.4920	0.0006	0

Dataset: Simulated clutter with PSNR 5-10

Model	OA ↑	mIoU ↑	P_d ↑	P_{fa} ↓
Onet	0.9873	**0.7168**	**0.6075**	0.0063
SAM2-tiny	0.9834	0.4918	0.0002	0
Guided-SAM2-tiny	**0.9875**	0.6606	0.4741	0.0038
SAM2-small	**0.9924**	**0.8088**	**0.7738**	0.0039
Guided-SAM2-small	0.9867	0.6592	0.4410	0.0041
SAM2-base	0.9832	0.4944	0.0106	**0.0004**
Guided-SAM2-base	0.9848	0.5939	0.3258	0.0041
SAM2-large	0.9857	0.6180	0.3204	0.0031
Guided-SAM2-large	0.9835	0.5204	0.0804	**0.0013**

Dataset: Rain clutter

Model	OA ↑	mIoU ↑	P_d ↑	P_{fa} ↓
Onet	0.9968	0.7881	0.6442	**0.0008**
SAM2-tiny	0.9940	0.4971	0.0001	0
Guided-SAM2-tiny	0.9968	**0.8149**	**0.8817**	0.0026
SAM2-small	0.9855	0.6087	0.6447	0.0126
Guided-SAM2-small	**0.9975**	**0.8148**	0.8099	**0.0016**
SAM2-base	0.9940	0.4970	0	0
Guided-SAM2-base	**0.9973**	0.8099	0.8087	0.0018
SAM2-large	0.9948	0.6539	0.4275	**0.0016**
Guided-SAM2-large	0.9966	0.7998	**0.8781**	0.0029

Dataset: ZY3

Model	OA ↑	mIoU ↑	P_d ↑	P_{fa} ↓
Onet	**0.9239**	**0.8292**	**0.8547**	**0.0552**
SAM2-tiny	0.6862	0.3649	0.2500	0.2709
Guided-SAM2-tiny	0.7805	0.4566	0.3658	0.1967
SAM2-small	0.7743	0.4402	0.3043	0.1721
Guided-SAM2-small	0.7833	0.4818	**0.4473**	0.2175
SAM2-base	0.7326	0.4316	0.4117	0.2529
Guided-SAM2-base	**0.7931**	**0.5033**	0.3629	**0.1327**
SAM2-large	0.7087	0.3882	0.3256	0.2625
Guided-SAM2-large	0.7801	0.4888	0.4059	0.1650

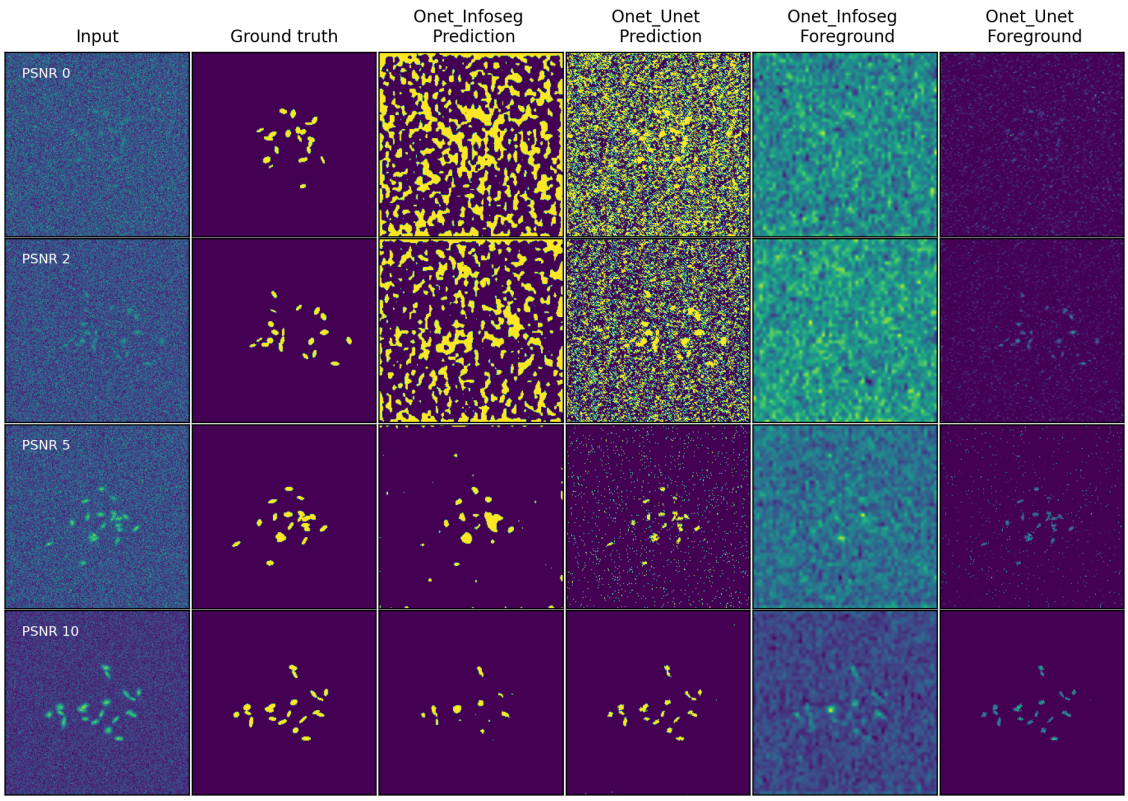
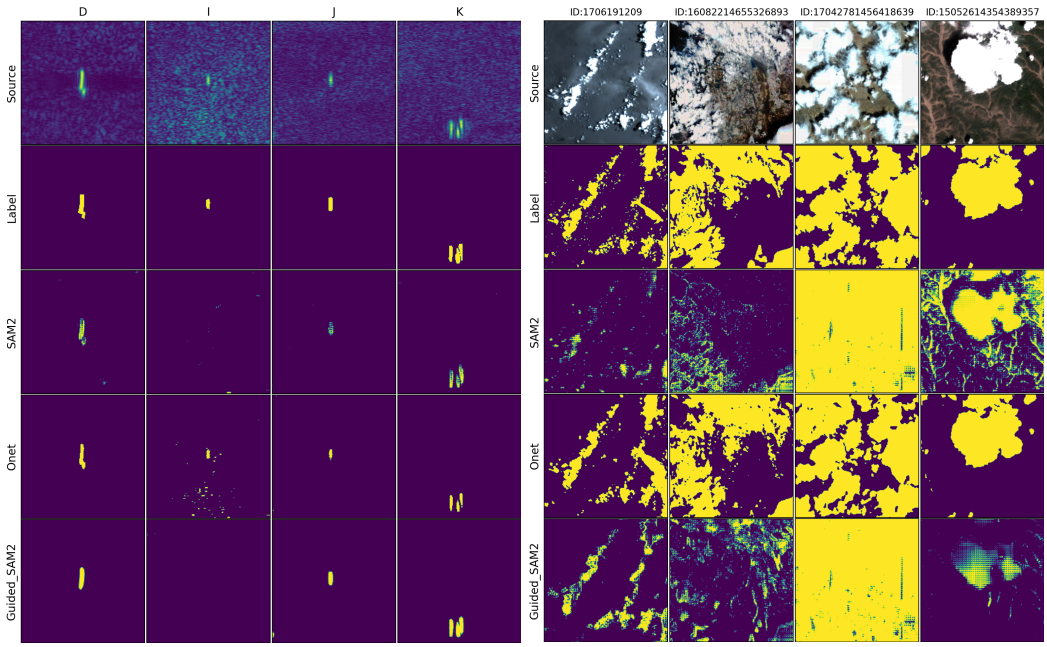


Fig. 8: Comparison of foreground prediction in Rayleigh-distributed clutter with PSNR values of 0, 2, 5, and 10 for Onet-Infoseg and Onet-Unet.

annotating the target masks in 6 million videos, then training the model to segment masks automatically, and using human interaction to further refine the masks. After one round of training, 11 billion masks were annotated using this self-generated data plus user evaluation. To improve segmentation consistency across video frames, SAM2 introduced a memory bank to store features of the segmented targets over several consecutive frames, guiding object segmentation in subsequent frames through cross-attention.

U2Seg integrates CutLER [Wang et al., 2023], a method used for instance segmentation, and STEGO [Hamilton et al., 2022], a method for unsupervised semantic segmentation, to generate pseudo-labels for "things" (instances) and "stuff" (semantic background). It is the first to achieve unsupervised panoptic segmentation by using unsupervised clustering. Its outputs include both semantic background and individual identification of foreground objects, which is why the authors name it as "unsupervised universal segmentation" (U2Seg). It claims that this multi-task learning model allows U2Seg to achieve state-of-the-art (SOTA) performance on certain instance and semantic segmentation datasets.

Since SAM2 far exceeds that of SAM1, in our comparative experiments, we compare the performance of



(a) Target segmentation in real rain-clutter echoes. (b) Cloud segmentation in remote sensing images.

Fig. 9: Comparison of image segmentation for SAM2, Guided-SAM2, and Onet. Here, SAM2 in 3rd row adopts the grid prompts with equally 20-pixel distance. In Guided-SAM2, the prompting points are the centroid points of the connected regions in Onet's foreground mask. The point prompt guided by Onet helps to improve the accuracy of SAM2 in average in these two datasets.

Onet with SAM2 and U2Seg on simulated sea clutter, real rain clutter for marine radar object segmentation, and ZY3 remote sensing images for cloud segmentation. In these experiments, we set SAM2's memory bank to be 1 frame for image segmentation and choose the 'small' size model (compared to the tiny, base and large models, the small model has the best performance in average. Please see Table VII). Avoiding human intervention, we employ SAM2's grid-point prompt method with a 20-pixel distance (considering our need to detect small targets and clouds). Among the output masks (whole, parts, and subparts) from SAM2, we select the mask with the best overall accuracy as the foreground mask. For U2Seg we utilize the pre-trained model of panoptic segmentation model with default 800 clusters.

In rain-clutter scenario (please see Fig. 9a), SAM2's segmentation of foreground mask became sparse, and it failed to cluster all the pixels within the target area effectively, leaving a portion of foreground pixels classified as background. It implies that SAM2's prediction is disturbed by the rain clutter. To further improve SAM2's segmentation performance, we exploited Onet's prediction masks to locate the connected regions and extracted the center point coordinates as SAM2's positional prompts. This resulted in Guided-SAM2 in the last row of the Fig. 9a. We notice that even with prompt points placed on the

real target, Guided-SAM2 is unable to overcome the influence of clutter for target 'I' (target in the 2nd column).

In cloud detection experiments (please see Fig. 9b), SAM2's performance was not outstanding. One possible cause is that clouds in remote sensing images can appear in large clusters, small scattered patches, or thin transparent forms which are unseen in SAM2's training datasets. After incorporating Onet's predictions, Guided-SAM2's attention was more focused on the clouds, but it is also affected by the land background with uniform intensity in the 3rd column.

As shown in Table VIII, the pre-trained SAM2 performs better than Onet in the simulated clutter across all PSNR levels, demonstrating its superior ability to segment small objects in Rayleigh-distributed clutter. In this case, applying Guided-SAM2 actually degrades the SAM2's prediction. If we use the Onet's foreground score maps F_g to enhance the source images X (i.e. $X = X + F_g$, please see Fig.10), SAM2 will get better performance on the augmented images (SAM2-Aug) than on the source images (SAM2-Source). In contrast, pre-trained SAM2 performs worse in real radar clutter and remote sensing images. In these scenarios, combining the predictions of the Onet with SAM2's prompt, Guided-SAM2 improves SAM2's performance.

When we directly applied U2Seg to remote sensing images for cloud detection and target segmentation in marine radar, its performance was not as outstanding as on natural image datasets. In the ZY3 dataset, U2Seg tends to merge high-saturation backgrounds and clouds into a semantic background, often neglecting large independent clouds or small scattered clouds (please see Fig. 11). In the marine radar target segmentation experiment, U2Seg fails to separate the target from the cluttered background, indicating that it is not sensitive to small targets in semantic segmentation tasks (please see Fig.12).

We observe that U2Seg's semantic segmentation model is based on DINO (pre-trained ViT features [Caron et al., 2021] on the ImageNet dataset [Deng et al., 2009]), which serves as the foundation for feature clustering by mask cut. Since ImageNet mainly consists of natural images and photos, the features learned by DINO are biased to the visible light images which are sensed by the normal camera, making it less sensitive to the varied shapes and intensities of the clouds in remote sensing images and clutter textures in radar's microwave echoes. In Fig.13 it shows attention bias in the DINO's attention map of the cloud image and weak bipartition property in the radar clutter echoes. We notice that the pre-trained DINO backbone pays contrary attention to the big clouds and isolated small clouds. It makes the U2Seg

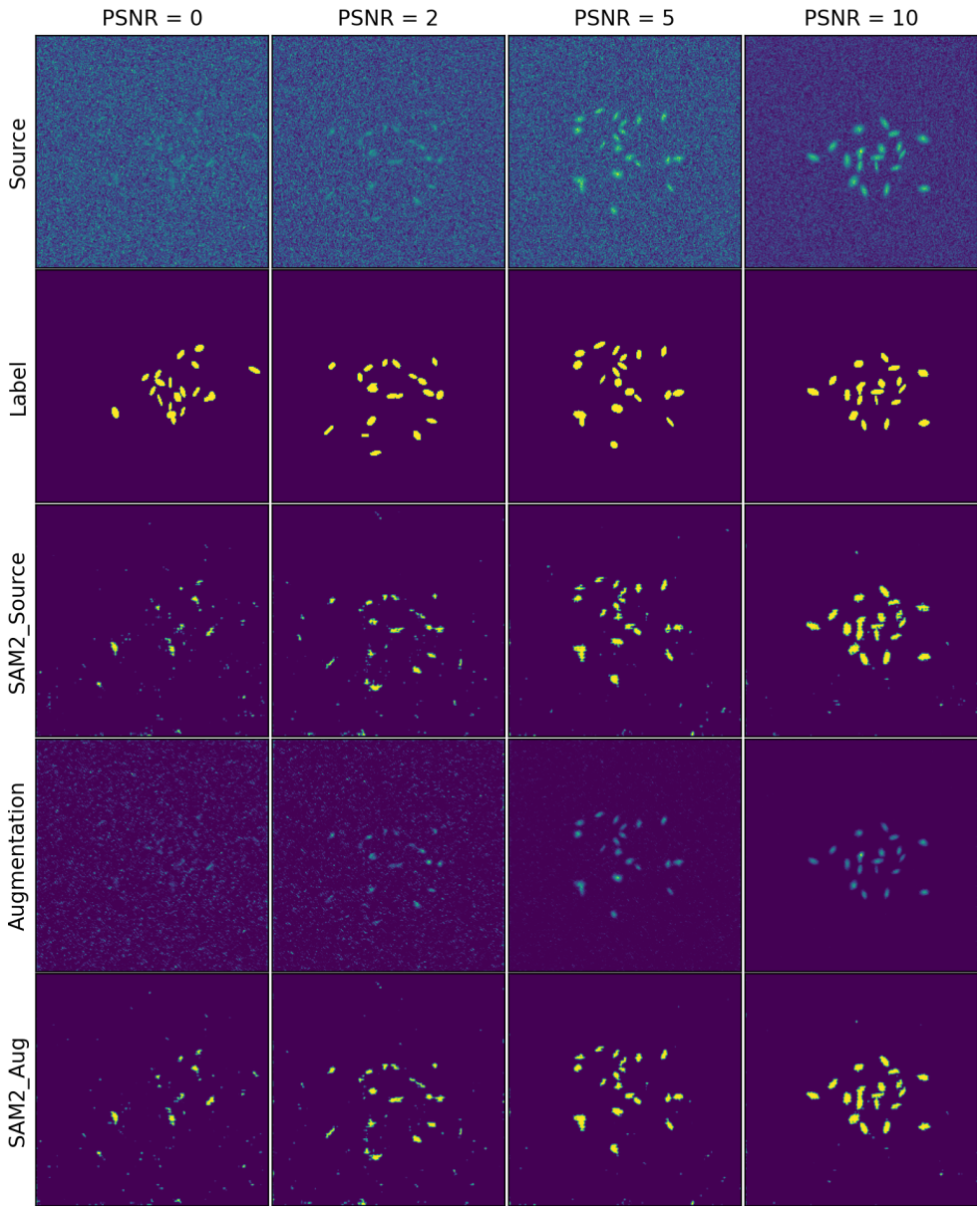


Fig. 10: Target segmentation in Rayleigh-distributed clutter using SAM2. In the fourth row, the source images are augmented with Onet's foreground score maps. The last row, SAM2-Aug shows the predictions of SAM2 on these augmented images. Compared to SAM2-Source, SAM2-Aug identifies more target pixels.

harder to cluster the clouds with different sizes into the same label. In the rain echoes, the target and clutter generate similar features in the pre-trained DINO model. U2Seg's bipartition scheme on their feature affinity matrix does not work well in this scenario. Most clutter is misclassified as the target, or vice versa. Therefore, retraining or fine-tuning the ViT backbone and rearranging the cluster model for the CutLER are necessary for using U2Seg in these datasets.

TABLE VIII: Performance comparison for SAM2, Guided-SAM2, U2Seg and Onet

Dataset: Simulated clutter with PSNR 0-2

Model	OA ↑	mIoU ↑	$\mathbf{P_d}$ ↑	$\mathbf{P_{fa}}$ ↓
SAM2-Source	**0.9863**	**0.6300**	**0.3237**	**0.0026**
SAM2-Aug	**0.9871**	**0.6518**	**0.3760**	**0.0026**
Guided-SAM2	0.9855	0.6239	0.3327	**0.0035**
U2Seg	0.9589	0.4795	0.0000	0.0249
Onet	0.9548	0.5010	0.1369	0.0314

Dataset: Simulated clutter with PSNR 5-10

Model	OA ↑	mIoU ↑	$\mathbf{P_d}$ ↑	$\mathbf{P_{fa}}$ ↓
SAM2-Source	**0.9924**	**0.8088**	**0.7738**	**0.0039**
SAM2-Aug	**0.9939**	**0.8383**	**0.8014**	**0.0029**
Guided-SAM2	0.9867	0.6592	0.4410	0.0041
U2Seg	0.7878	0.3939	0.0001	0.1989
Onet	0.9872	0.7232	0.6291	0.0067

Dataset: Rain clutter

Model	OA ↑	mIoU ↑	$\mathbf{P_d}$ ↑	$\mathbf{P_{fa}}$ ↓
SAM2	0.9855	0.6087	**0.6447**	0.0126
Guided-SAM2	**0.9975**	**0.8148**	**0.8099**	**0.0016**
U2Seg	0.9789	0.4895	0.0000	0.0152
Onet	**0.9969**	**0.7916**	0.6270	**0.0005**

Dataset: ZY3

Model	OA ↑	mIoU ↑	$\mathbf{P_d}$ ↑	$\mathbf{P_{fa}}$ ↓
SAM2	0.7743	0.4402	0.3043	0.1721
Guided-SAM2	**0.7833**	**0.4818**	**0.4473**	**0.2175**
U2Seg	0.6578	0.3920	0.3430	0.2488
Onet	**0.9239**	**0.8364**	**0.8641**	**0.0539**

The rows with gray background color are unreliable for its zero P_d

It is also reported that the ViT backbone trained on 400 million internet images shows significantly better classification performance on natural images than in specialized fields such as remote sensing [Radford et al., 2021] . This problem persists in SAM and U2Seg. Both of them are mostly trained on the natural images. When comparing SAM2 and U2Seg in Table VIII, we found that SAM2 performed better, likely because SAM2 was trained on a broader dataset and has larger model parameters. The inclusion of a medical microscopy cell imaging dataset [Mariscal et al., 2021] in SAM2's training sets, for example, greatly aids in segmenting extended targets in the simulated clutter. Based on our experience, cells under a microscope often have elliptical shapes of varying sizes, and their imaging also resembles uniformly diffused textures, making SAM2 particularly useful for detecting the simulated targets in the Rayleigh-distributed clutter.

To effectively apply these foundational vision models in specialized fields, the large corresponding

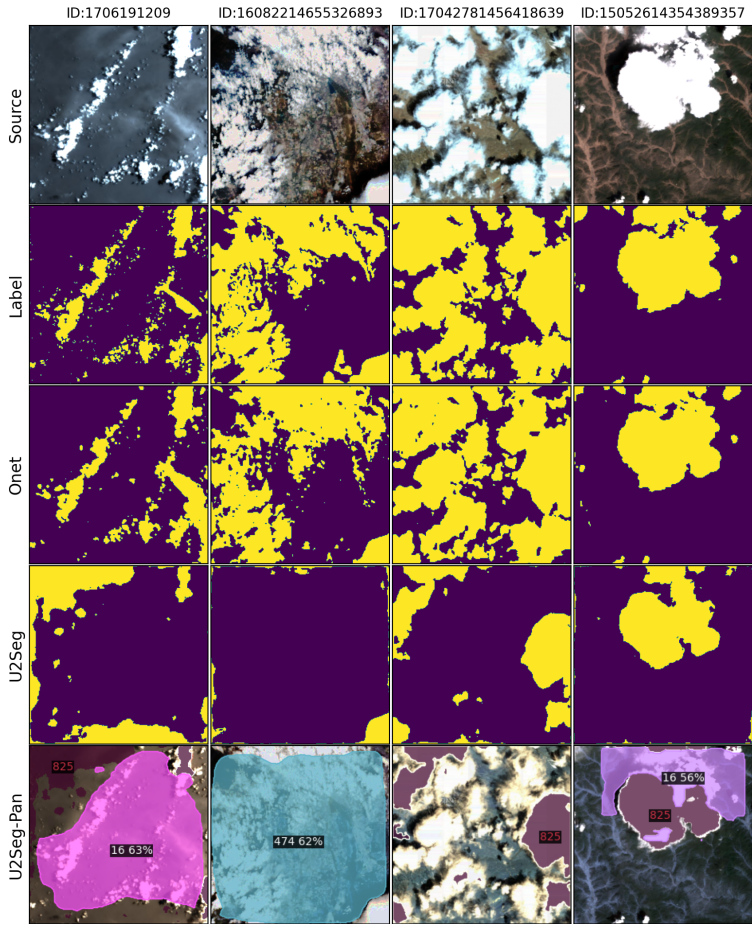
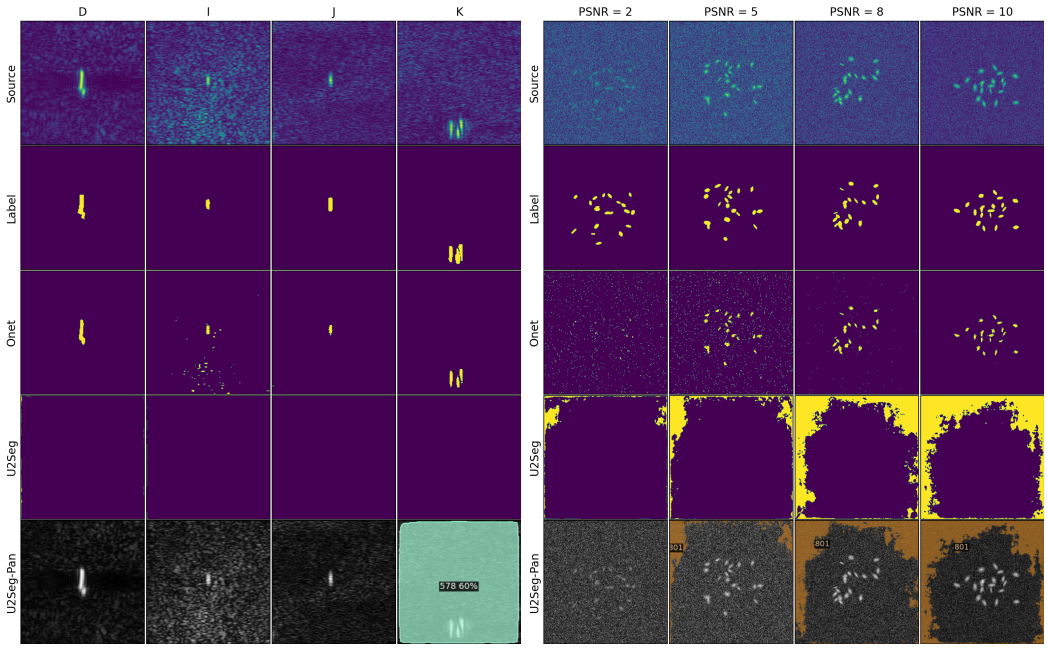


Fig. 11: Comparison of cloud segmentation in ZY3 between U2Seg and Onet. U2Seg-Pan outputs panoptic segmentation with instance IDs. In the last sub-figure, ID 825 appears to correctly identify a cloud region. However, most of the clouds in the first three images are incorrectly merged or missed.

datasets are needed, along with modifications and ablation studies tailored to the network [Wang et al., 2024]. Given the limited computational resources and time constraints, we were unable to complete such extensive data collection (there is no open marine radar image dataset with more than thousands samples so far) and conduct the vision foundation model re-training. What we can currently confirm is that small models like Onet, with their simple structure and no need for manual intervention, can serve as auxiliary tools for the large models in image fields where signal strength distinguishes targets. Onet can guide the large models to generate masks during the initial training stages and serve as an evaluation metric during testing.



(a) Target segmentation in real rain-clutter echoes. (b) Targe segmentation in simulated clutters.

Fig. 12: Comparison of target segmentation between U2Seg and Onet in radar echoes. U2Seg is less sensitive to small targets in clutter and fails to segment them effectively.

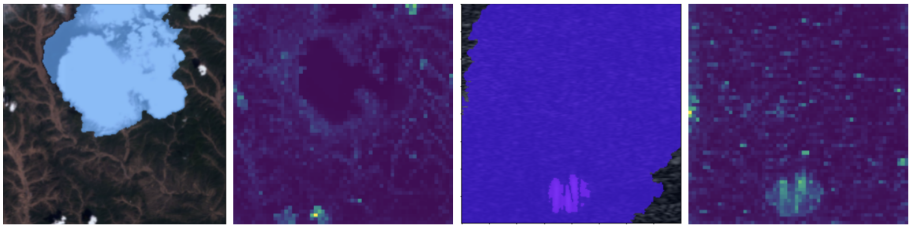
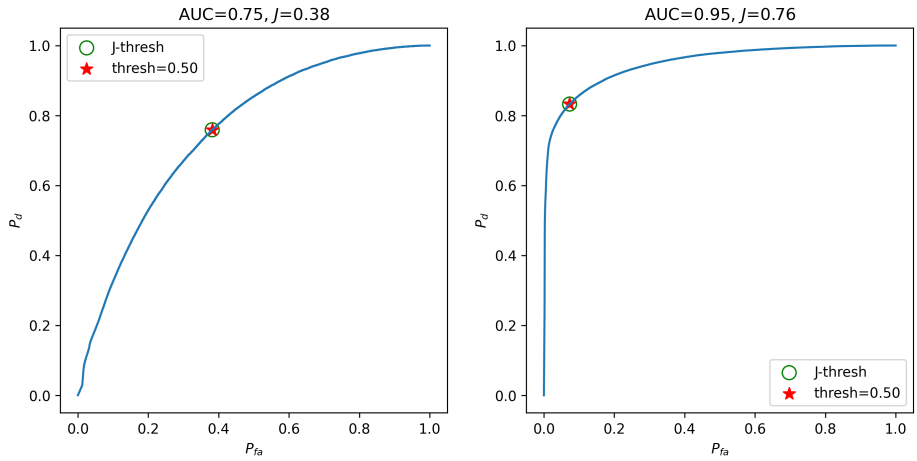


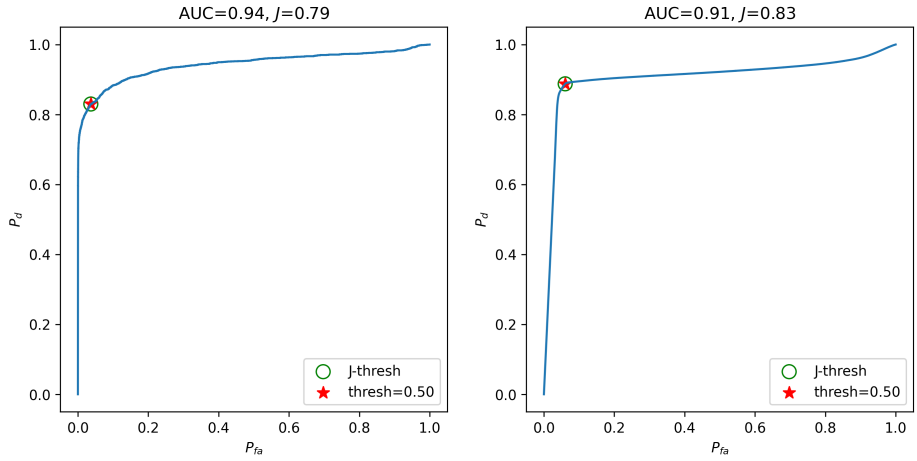
Fig. 13: (a) Cloud mask of U2Seg only covers the big cloud. (b) Cloud's attention map from the ViT's last block in DINO. The big cloud region (dark in center) and isolated small clouds (bright at bottom) get contrary attentions.(c) Target mask of U2Seg is extended by the rain clutter (d) Target's attention is interrupted by the clutter.

VIII. Discussion on the threshold selection for foreground segmentation

In the paper, we used a relatively simple and less obvious threshold selection method. Specifically, we predicted the logits for both foreground and background, V_t and V_d, for inputs X and $1-X$, respectively. $V = [V_d, V_t]$ was converted into probability tensors $S = [S_d, S_t]$ via a softmax function, and then turned into mask labels Y using the argmax function. In this case, if $S_t > S_d$, then the corresponding element in Y would be 1. Considering the constraint $S_t + S_d = 1.0$ of the softmax function, the detection threshold for all elements is essentially determined by whether S_t exceeds 0.5. Here, we employ the ROC curve



(a) PSNR0-2 (b) PSNR5-10



(c) Rain Clutter (d) ZY3

Fig. 14: ROC for target detection in simulated clutter with varied PSNR, rain clutter and ZY3-cloud datasets. The green circles mark the points with best J statistic and the red stars mark the points with fixed threshold 0.5. The threshold that produces the best J statistic coincides with the threshold of 0.5.

to discuss why we use this threshold. The ROC curve plots the true positive rate P_d against the false positive rate P_{fa} at different decision thresholds. That is to say, given a foreground probability map S_t, for each threshold γ in range (0,1) the predicted foreground mask is the $S_t > \gamma$. Then with the ground truth label and predicted masks, we can compute the P_d and P_{fa} for all the thresholds. Here, optimal threshold is found by searching for the maximum J statistic [Youden, 1950] as:

$$\begin{aligned} J &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} - 1 \\ &= P_d - P_{fa}. \end{aligned} \tag{7}$$

Here, TP, TN, FP, and FN are the true positive, true negative, false positive, and false negative, respectively. In equation (7), the thresholds with high P_d and relatively low P_{fa} will generate the maximum J.

For each ROC curve, we computed the threshold that yielded the maximum J score, marking the corresponding P_d and P_{fa} with a circle marker on the ROC curve, and also marking the star point corresponding to a fixed threshold of 0.5 (please see Figure 14). We observed that these two markers overlapped. At this point, the optimal threshold (1e-5 in our test) and the 0.5 threshold are both corresponding to the maximum J statistic. Why do the two markers for different thresholds coincide? We took the histogram of the foreground probabilities in rain clutter for explanation. It showed that the Onet's estimated probabilities for the foreground were concentrated around 0 and 0.9 (please see Fig. 15), indicating the model had high confidence in its predictions. Setting the foreground threshold anywhere between 0.1 and 0.8 resulted in the best J statistic. Therefore, selecting a threshold of 0.5 is reasonable.

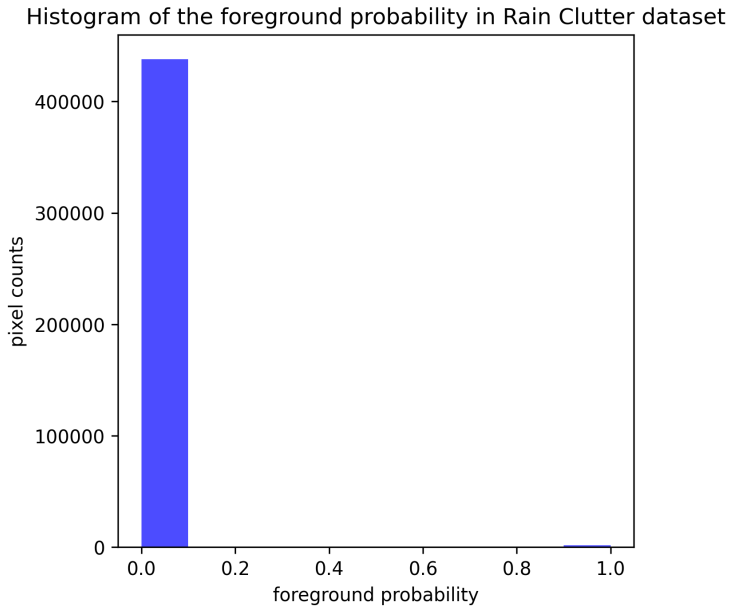


Fig. 15: Histogram of Onet's foreground prediction S_t on the rain clutter dataset. Since the clutter occupies the majority of the pixels, over 98% of the predicted probabilities are near zero. In contrast, the minority of pixels, corresponding to the target regions, have values around 0.9.

REFERENCES

Juan C. Caicedo et al. Nucleus segmentation across imaging experiments: the 2018 data science bowl. *Nature Methods*, 16(12):1247–1253, 2019. 8

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 9630–9640, 2021. 17

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. 7

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 17

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 13

Jianhua Guo, Jingyu Yang, Huanjing Yue, Xin Liu, and Kun Li. Unsupervised domain-invariant feature learning for cloud detection of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2022. doi: 10.1109/TGRS.2021.3120001. 6, 7

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *International Conference on Learning Representations*, 2022. 15

Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353, 2011. 7

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15979–15988, 2022. 13

Haichao Jiang, Wei Yi, Thia Kirubarajan, Lingjiang Kong, and Xiaobo Yang. Multiframe radar detection of fluctuating targets using phase information. *IEEE Transactions on Aerospace and Electronic Systems*, 53(2):736 – 749, 2017. 2

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint*

arXiv:2001.08361, 2020. 13

E. G. Mariscal, Hasini Jayatilaka, Özgün Çiçek, Thomas Brox, Denis Wirtz, and Arrate Muñoz-Barrutia. Search for temporal cell segmentation robustness in phase-contrast microscopy videos. *arXiv:2112.08817*, 2021. 19

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, volume 139, pages 8748–8763, 2021. 19

P. Swerling. Probability of detection for fluctuating targets. *IRE Transactions on Information Theory*, 6 (2):269–308, 1960. doi: 10.1109/TIT.1960.1057561. 2

Lin Wang, Xiufen Ye, Liqiang Zhu, Weijie Wu, Jianguo Zhang, Huiming Xing, and Chao Hu. When SAM Meets Sonar Images. *IEEE Geoscience and Remote Sensing Letters*, 21:3387712, January 2024. 20

Xudong Wang, Rohit Girdhar, Stella X. Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3124–3134, 2023. 15

W.J. Youden. Index for rating diagnostic tests. *Cancer*, 3:32–35, 1950. 22

Yi Zhou, Hang Su, Shuai Tian, Xiaoming Liu, and Jidong Suo. Multiple-kernelized-correlation-filter-based track-before-detect algorithm for tracking weak and extended target in marine radar systems. *IEEE Transactions on Aerospace and Electronic Systems*, 58(4):3411–3426, 2022. 2