Revised Supplemental Material

1. Samples

We first illustrate several samples from our sonar image dataset in Figure 1. There are three categories with various observed conditions and scales. Corpse has the poorest imaging quality and lowest resolution because it is hard to capture in the underwater area. The shipwreck and plane wreckage have large volume so that the samples containing these objects are relative high-resolution.

In addition, we show the noised samples which we use in Section 4.6 of the paper in Figure 2. Four kinds of noised samples illustrate different patterns in pixel space.

2. Detection Results

Figure 3 and Figure 4 show the results of detector with several mechanisms. Figure 3 showcases the detection on the original test set while Figure 4 is in the case of noise attack to test set. It is obvious that the original Faster R-CNN is vulnerable to noise attack while NAN brings performance improvement. Equipped with NAN and NB simultaneously, the detector yields high performance as well as noise robustness in both cases.

3. Ablation Study on Different NAN Strategies

As a supplement for Section 4.3, in this part we explore the contribution of different strategies utilizing the adversarial examples. In this section only NAN works. As Table 1 shows, only taking advantage of KL-divergence loss to approximate the distribution of noised feature gains a mAP of 84.4%, making significant progress compared with the baseline. When making the adversarial examples be involved in classification and bounding box regression individually, the detector yields results of 86.1% and 85.7%, respectively. The combination of the two strategies makes the performance reach 86.8% mAP. Furthermore, the integration of the three alternatives improves the performance to 88.3% mAP. The result boosts an improvement of 6.6% compared with the baseline. Only utilizing KL divergence loss to approximate a true distribution obtains a relatively low mAP of 84.4% compared with other strategies. We hypothesize that in this case, NAN plays a role that only generates adversarial examples with fixed parameters, none backpropagation is used in NAN because instead of being fed into the subsequent layers, the adversarial examples are only used to measure the divergence with original examples.

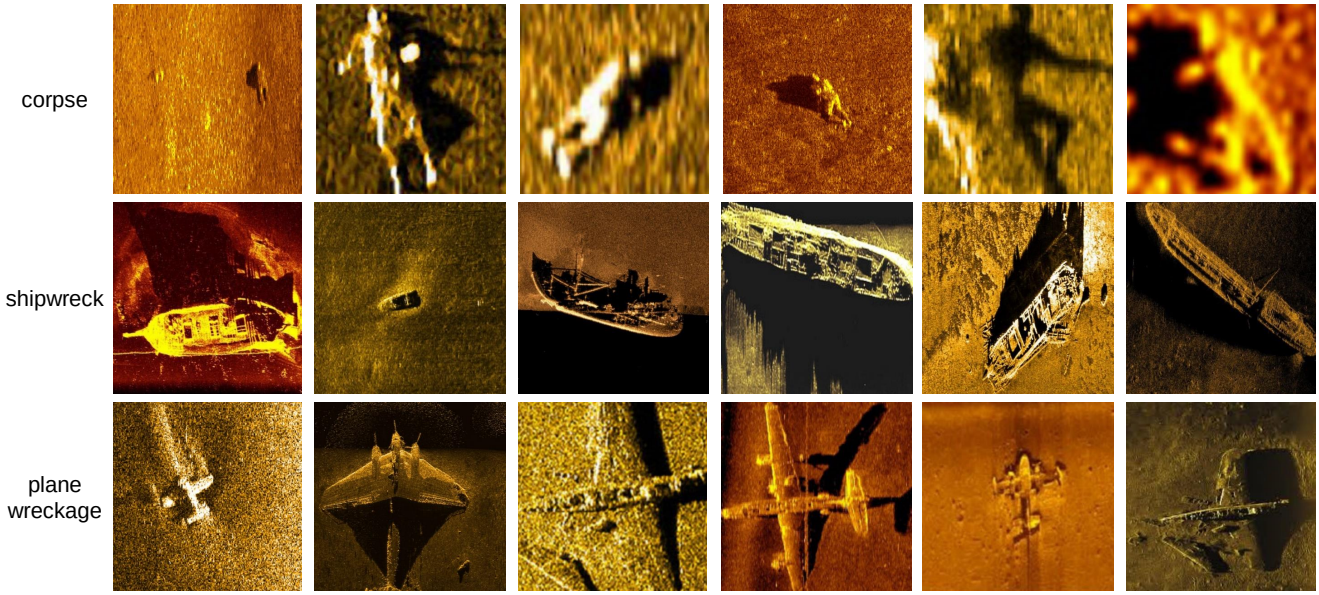


Figure 1. Samples from our collected dataset

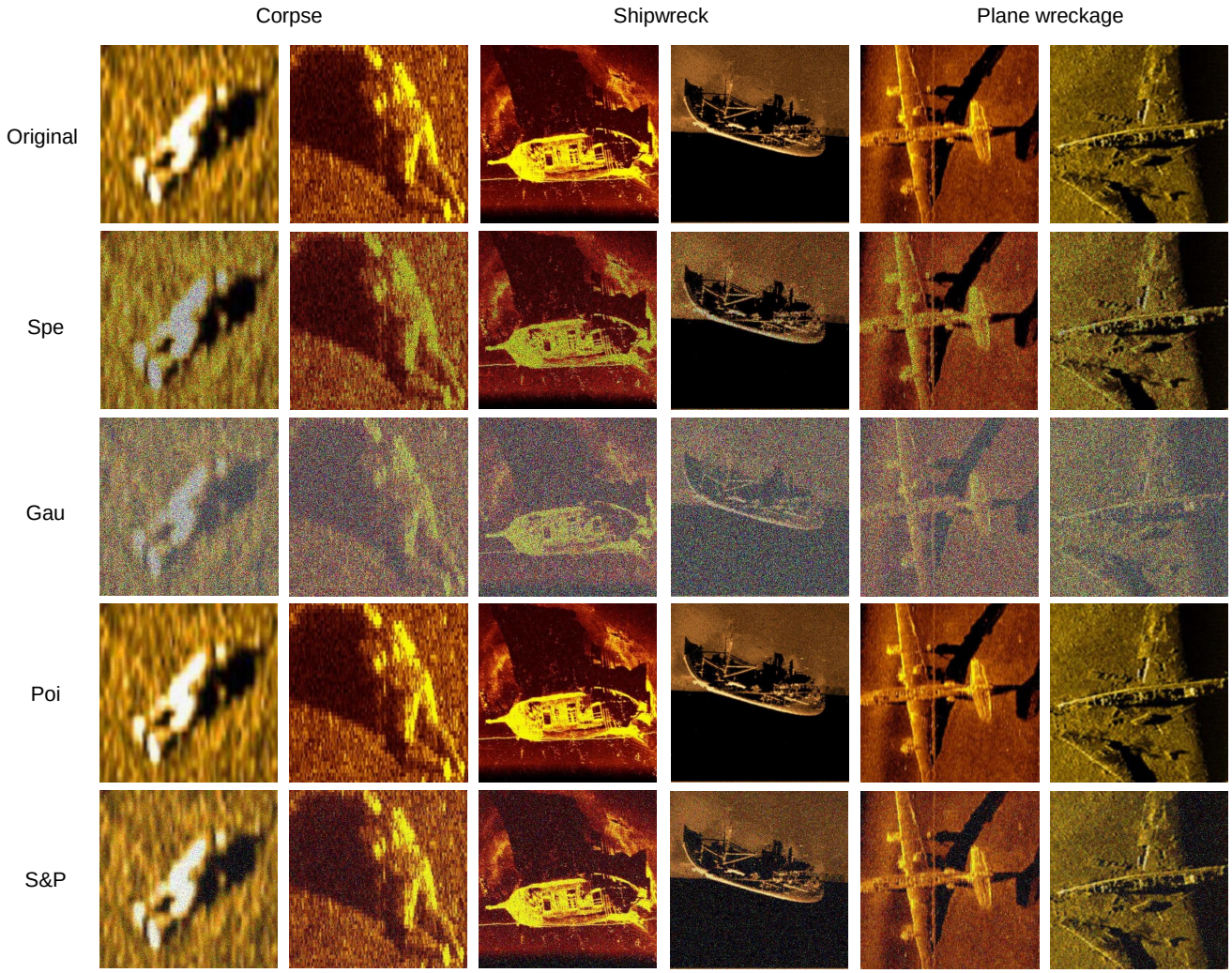


Figure 2. Noised samples used in Section 4.6. **Spe**, **Gau**, **Poi** and **S&P** represent speckle, gaussian, poisson and salt-and-pepper noised samples, respectively.

4. Mechanism of Noise Block

We present detailed mechanism and flowchart of NB in this section. In this paper, we embed NB between the first and second convolutional layers of backbones, which aims to predict Rayleigh noise in high resolution feature space while provide prior knowledge to NAN.

As Figures 5 shows, $x_i^{'}$ is the input feature map of NB. Firstly it is feed into the rightmost branch to generate Rayleigh noise with zero mean and variance ν_i. To get variance ν_i, the input feature $x_i^{'}$ passes through three 3×3 convolutional layers and a RoI Align Pooling layer [1] as well as a fully connected (FC) layer. The RoI Align Pooling layer is to transfer feature maps into a fixed-length input of FC layer. We get a parameter σ_i from FC layer. Unlike NAN, here we do not directly use square value of σ_i as the predicted variance because in the upstream layers of backbones (especially in ResNet-101), the absolute value of σ_i has a high probability to be a large scalar which generates dramatically intense noise and ruins detector. Therefore, we set a threshold τ to restrict the noise into a reasonable range, the variance ν_i is calculated by,

$$\nu_i = \begin{cases} \tau/{\sigma_i}^2, & if\ {\sigma_i}^2 > \tau \\ {\sigma_i}^2, & otherwise, \end{cases} \tag{1}$$

τ is related to the distribution of training samples, we assume that a training set with smaller variety is supposed to be assigned a larger threshold, with the aim to improve the variety in feature domain. In this paper, we set τ=1 when

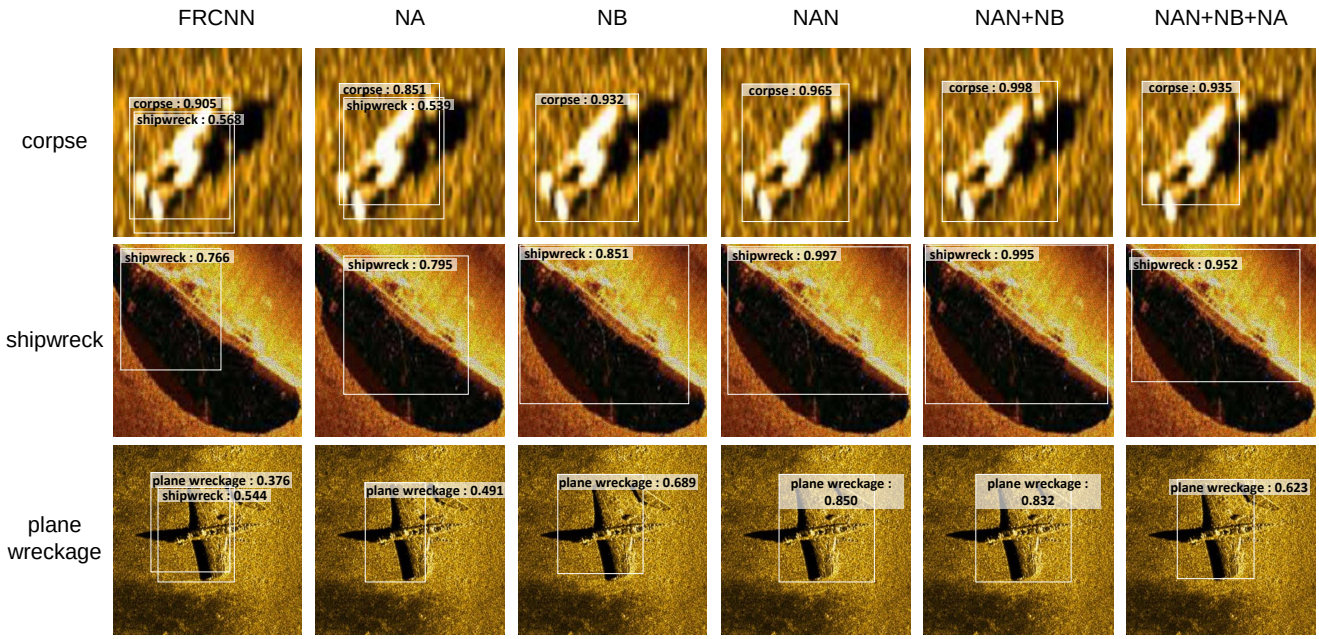


Figure 3. Detection results on original test set

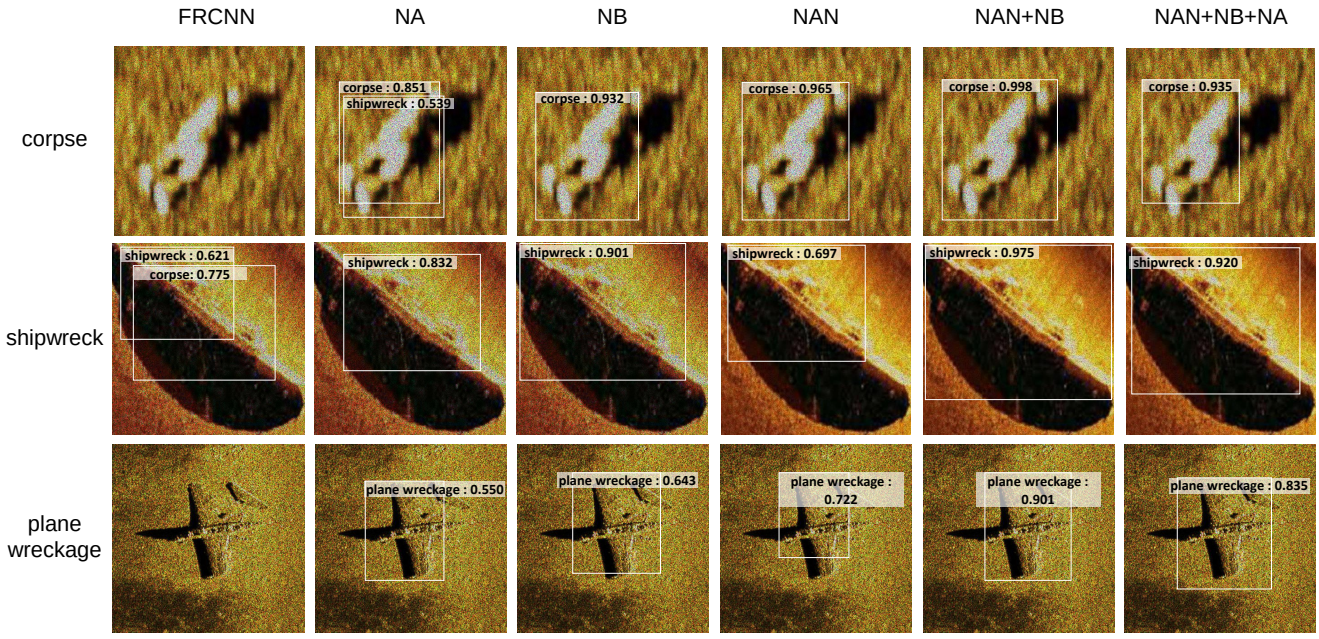


Figure 4. Detection results on speckle noise attacked test set (μ=0, σ^2=0.5). The image without bounding boxes means that the detector yields no box which is correct (overlap $>$ 0.5)

training with our sonar dataset, and set τ=0.1 when training with PASCAL VOC and MS COCO datasets.

Once the variance is predicted, the Rayleigh noise is generated by the Equ. (10) in the paper. Then as the middle branch shows, the Rayleigh noise is add to the original feature map by the noise model Equ. (13) in the paper (γ=1), yielding the adversarial Feature α_i.

To both utilize the original and adversarial features during training stage without changing the shape of the input feature, we concatenate both of them on channel dimension (leftmost branch), feeding it into a 1×1 convolutional layer to get output feature x_i^o. This feature map carries both

Method	mAP	cp	shw	plw
FRCNN [5]	79.8	72.8	90.6	76.0
FRCNN+	81.7	84.1	90.5	70.5
Ours(kl)	84.4	82.9	90.6	79.6
Ours(bbox)	85.7	85.4	90.3	**81.5**
Ours(cls)	86.1	93.1	**90.9**	74.4
Ours(cls+bbox)	86.8	90.1	90.6	79.7
Ours(cls+bbox+kl)	**88.3**	**96.7**	90.8	77.4

Table 1. Detection average precision (%) of different strategies utilizing the adversarial examples (with ResNet-101). The kl means use the KL divergence loss to approximate a true distribution by the adversarial examples. Cls and bbox refer to leveraging the adversarial examples to participate in decision making of classification and bounding box regression, respectively.

Method	ResNet-101	VGG16
H,W	300	600
H_1,W_1	150	150
H_2,W_2	75	75
H_3,W_3	38	38
C	64	64
P	38	38

Table 2. Feature map size, channel and pooling size (with 600 × 600 input size).

the pattern of original feature and the Rayleigh noised feature, the Rayleigh noised pattern provide prior knowledge to NAN.

During test stage, only the two leftmost branches work, which means two identical input features are concatenated and fed into the following layers.

When the input size of image is 600 × 600. The feature map size, channel and pooling size in Figure 5 are showed in Table 2.

5. Random noise augmentation in feature space

To verify that NAN and NB have better performance and robustness than random number generators, in this section we compare the our method with random noise augmentation in feature space.

We conduct two experiments of feature space noise augmentation. In the first one, we keep the structure of NAN and NB, only replacing the predicted noise with random noise. In this case, the NAN and NB are still active, but instead of predicting noise variance by its learnable weights, they generate variance in a random way. We mark this model as Model A.

In the second experiment, we remove both NAN and NB, only training baseline with random noise introduced in feature space, which is to exclude the effect of the structure of

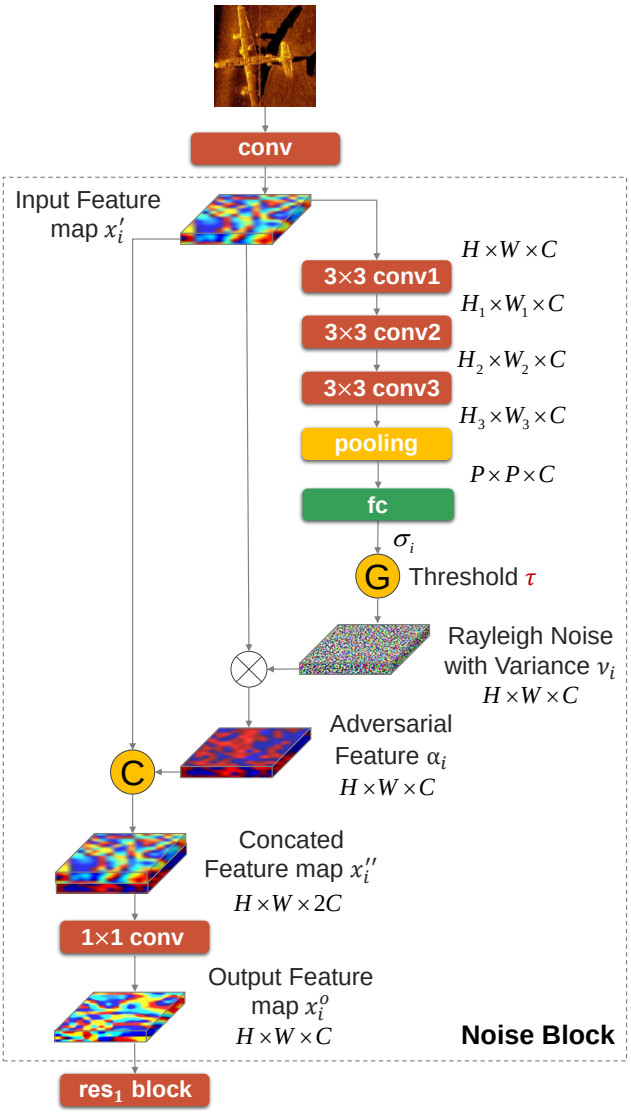


Figure 5. Flowchart of Noise Block.

the two sideway networks. We mark this model as Model B.

In each experiment, we use the ResNet-101 as backbone, the noises are introduced to the same layers as our approach. In addition, the randomly generated noises follow Rayleigh distribution with zero mean, and the variances randomly varies from 0 to v_1. All the random noises are added to the features with noise model Equ. (13) in the paper (γ=1).

Before comparing the performance, we analyze the distribution of variances generated in each model. We train our model, Model A and model Model B for 50k iterations, respectively. In k-th iteration, we record a pair of variances $\left\{\nu_k, \overline{\sigma^2}_k\right\}$. ν_k is generated where NB locates; $\overline{\sigma^2}_k$ is generated where NAN locates (note that in each iteration, the

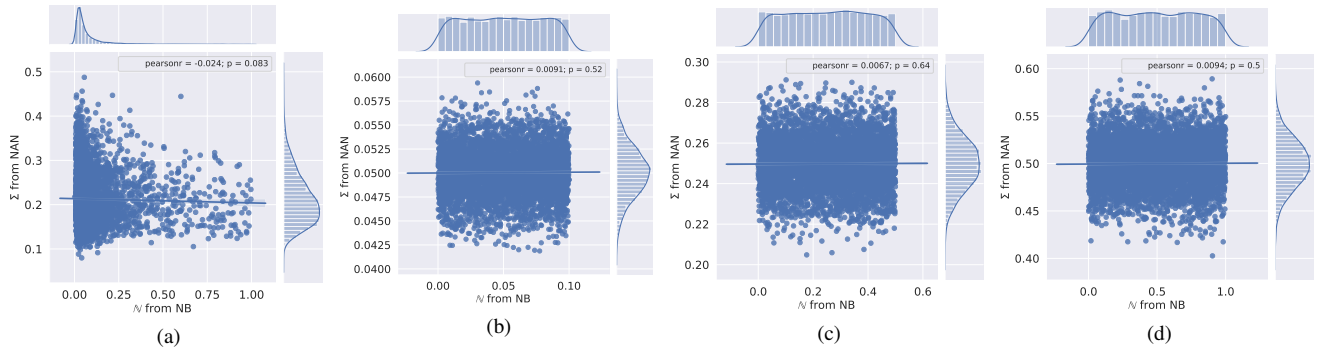


Figure 6. Marginal distribution of $\mathbb{N}$ and Σ as well as their joint distribution from each model. (a) Distribution on our model; (b) Distribution on model A, v_1=0.1; (c) Distribution on model A, v_1=0.5; (d) Distribution on model A, v_1=1.

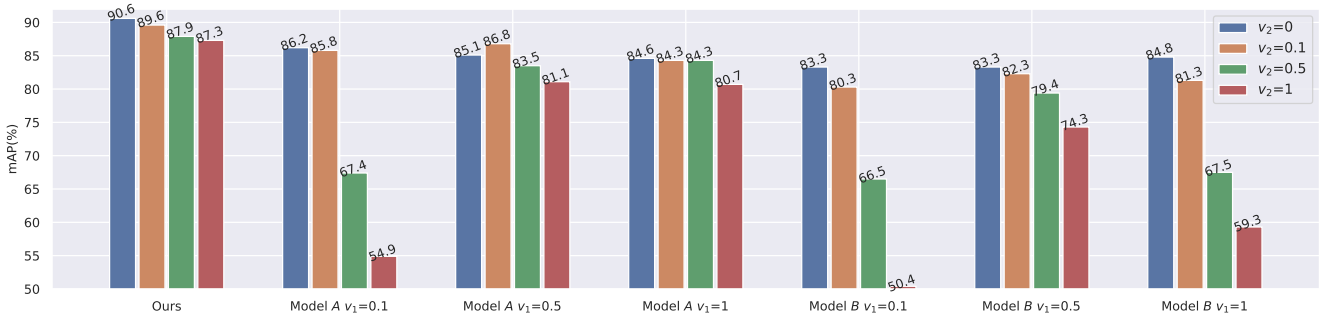


Figure 7. Results on our model, Model A and Model B under different magnitudes of speckle noise attack. The variance of noise attack on test set randomly varies from 0 to v_2. we set three cases of v_2: 0.1, 0.5 and 1.

output from RoI Pooling is a mini-batch which contains n feature maps, thus there are n variances σ_i^2 generated. We use the arithmetic mean $\overline{\sigma^2}$ of the n variances to represent the variance of this mini-batch).

Totally, we record $\mathbb{N}=\{\nu_1, \nu_2, ..., \nu_k\}$ and $\Sigma=\left\{\overline{\sigma^2}_1, \overline{\sigma^2}_2, ..., \overline{\sigma^2}_k\right\}$. Figure 6 shows the both marginal distribution of $\mathbb{N}$ and Σ as well as their joint distribution from each model. It is obvious that in our method, there is a qualitatively negative correlation between $\mathbb{N}$ and Σ generated from NB and NAN (Figure 6 (a)). Instead, there is no remarkable correlation between $\mathbb{N}$ and Σ in Model A (Figure 6 (b)(c)(d)) and Model B (since the scatter plots of Model A and Model B are similar, we only illustrate plots of Model A). We also report the Pearson Correlation Coefficient (PCC) [2] on each plot. The negative PCC means there is a negative linear correlation between $\mathbb{N}$ and Σ (Figure 6 (a)). The PCC close to 0 means no linear correlation between $\mathbb{N}$ and Σ (Figure 6 (b)(c)(d)). In addition, we illustrate p-value [6] on each plot, smaller p-value means higher statistical significance of the results. From the marginal distribution of $\mathbb{N}$ and Σ as well as their joint distribution, we can prove that the noise predicted by our model (Figure 6 (a)) has different pattern from one generated from a random number generator (Figure 6 (b)(c)(d)).

Figure 7 illustrate the results on our model, Model A and Model B under different magnitudes of speckle noise attack. It is remarkable that our model (Ours) yields competitive results and noise robustness. On the contrary, Model A and Model B shows relatively weaker performance. Model A achieves better results when the introduced random variances vary from 0 to 0.5 or 1. However, with slighter noise augmentation (v_1=0.1) in feature space, the robustness of Model A dramatic drops (54.9% mAP under noise attack with v_2=1). Model B yields poorer performance and robustness than Model A, especially under intense noise attack, *e.g.* in the case of v_1=1, the performance and noise robustness of Model B is far behind Model A when v_2=0.5 or 1.

It is worth noting that although Model A introduces noise in a random way, compared with Model B, it still shows moderate robustness with v_1=0.5 or 1. It demonstrates the effect of NAN and NB, even introducing noise variances in a random way, they still guarantee the robustness of detector. All in all, our method is far more than a random number generator on both performance and noise robustness.

6. Results on PASCAL VOC 2012 and MS COCO

Backbone	ResNet-101		VGG16	
test set	O	Gau	O	Gau
FRCNN [5]	73.8	-	70.4	-
Ours (Gau)	**75.6**	73.9	**71.5**	68.8

Table 3. PASCAL VOC 2012 test mAP (%). **O** means test on original `test` set; **Gau** means test on gaussian noised (zero mean, variance is 0.1) `test` set; **Ours (Gau)** means our approach with gaussian noised adversarial examples.

We evaluate our method on additional optical datasets PASCAL VOC 2012 and MS COCO [3] to verify its correctness. For PASCAL VOC 2012, We use VOC07 `trainval+test` and VOC12 `trainval` ("07++12") for training, and test on VOC12 `test` set. Following the stage-wise training strategy in Section 3.2 in the paper, we set SGD for 20K and 180K training on each stage. The initial learning rate is 0.001 and decreases to 0.0001 after 8K and 60K iterations, respectively. For MS COCO, we use `trainval` for training and test on the `test-dev` set. We train the model for 50K and 320K iterations on each stage, with a starting learning rate 0.01 for first 20K and 80K iterations, divided by 10 for each 50K iterations (second stage). During training stage, the batch size of image is 4 in each iteration. We introduce gaussian noise adversarial examples in both NAN and NB. For other implementation details, we follow the Section 4.2 in the paper.

Table 3 shows the results on VOC12 `test` set. Compared with the base network Faster R-CNN [5], our method achieves better performance, 75.6% on ResNet-101 and 71.5% on VGG16, which is the similar improvement as on VOC07. We also verify the gaussian noise robustness, introducing gaussian noise (zero mean, variance is 0.1) to the `test` set. Under noise attack, our model yields dropped results which are still competitive (71.5% on ResNet-101 and 68.8% on VGG16).

Results on MS COCO are summarized on Table 4. On `test-dev` set, our method improves the results to 35.8% and 22.3% with the two backbones, respectively. The model based on ResNet-101 shows better performance on both median and large objects while drops on the small ones, which implicitly shows that our the noise adversarial method may have side effects on small objects. In addition, we also report the results under gaussian noise attack. As expected, in this case the results drop with a certain amount of degree.

The results on both PASCAL VOC 2012 and MS COCO further demonstrate the correctness of our approach, which means that our method not only works on sonar images but also on optical images.

7. Effect of the Exponential Parameter

We specifically discuss the exponential parameter γ of subsection 3.2. The parameter γ is related to the dependence on the original examples χ. Figures 10 shows the effects of γ on the test performance on our dataset. For the detector with VGG16, the mAP has a slow downward trend with the increase of γ. The strategy of gradient clipping with a gradient threshold τ=10 is applied in VGG16's training approach to avoid gradient explosion. It gains a largest mAP of 81.3% at the initial value of γ=0, which is the case that the perturbation noise is induced by an additive noise model.

The detector with ResNet-101 shows a substantial higher performance when $\gamma \leq 3$, reaching a highest mAP of 88.3% with γ=1. However, the model collapses when $\gamma \geq 4$ with a gradient explosion. It is mainly because that without gradient clipping [4], the gradient accumulates with the exponential growth of the original examples χ, which ruins the model during several iterations.

Both of the performance of two models decrease from γ=1, it is mainly because that the larger exponential parameter γ means a heavier dependency on the original feature γ, which induces larger gradient failing to be reduced and degrade the effect of perturbation noise.

Figure 7 (a) and Figure 7 (b) show the result with speckle noise added to the test set on ResNet-101 and VGG16, respectively. In Figure 7 (a), the detector equipped with NAN which is tuned by γ from 0 to 3 yields competitive results compared to the baseline. The mAP of NAN decreases with the increase of γ. However, all the cases in NAN outperform the original detector. The higher mAPs explain that the noise robustness detector is strengthened by the introduction of noise perturbation from NAN. Figure 7 (b) illustrates similar cases with lower mAPs. It is obvious that VGG16 is prone to be attacked by the noise with large intensity (*e.g.* $\sigma^2 = 0.5$). With VGG16, NAN provides higher robustness with a relatively low γ (*e.g.*, $\gamma \leq 1$).

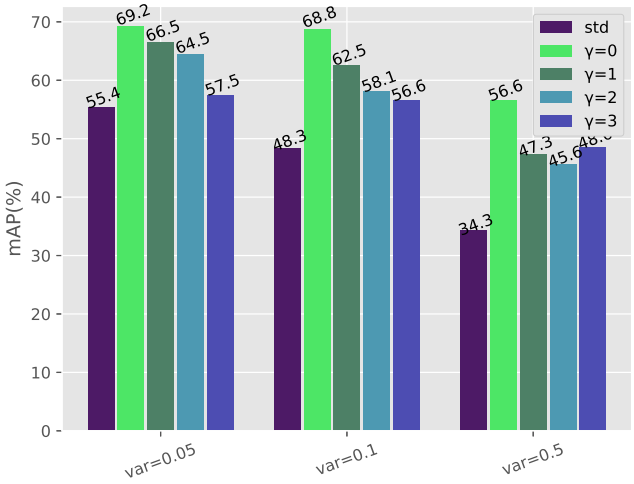
Figure 7 (a) and Figure 7 (b) display the same case with Figure 7 (a) and Figure 7 (b), but instead of NAN only, they leverage the combination of NAN and NB. Both of them show that the detector equipped with NAN and NB simultaneously achieves strong noise robustness as well as high performance with various parameter γ, especially in the case of $\gamma = 1$.

8. Error analyses

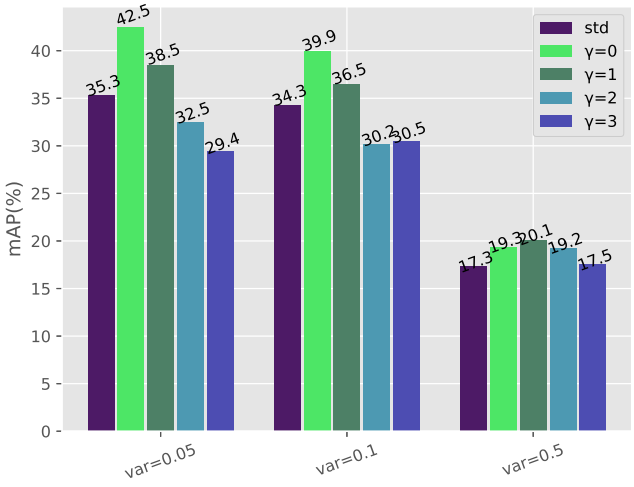
We specifically display the error analyses in terms of class in Section 4.5. As Figure 12 shows, with the speckle noise attack, corpse and plane wreckage can be easily confused with other classes by the original detector. Only the shipwreck has less false positives, which is mainly caused by the high resolution of original shipwreck sonar images.

Method	noise attack?	backbone	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
FRCNN [5]	no	ResNet-101	34.9	55.7	37.4	15.6	38.7	50.9
FRCNN [5]	no	VGG16	21.9	42.7	-	-	-	-
Ours (Gau)	no	ResNet-101	35.8	55.9	38.6	14.7	41.1	51.6
Ours (Gau)	no	VGG16	22.3	42.8	22.8	5.9	25.2	35.8
Ours (Gau)	yes	ResNet-101	32.3	53.6	37.1	12.9	33.7	50.3
Ours (Gau)	yes	VGG16	19.8	41.7	20.6	3.3	24.0	32.1

Table 4. MS COCO `test-dev` mAP (%). **Ours (Gau)** means our approach with gaussian noised adversarial examples.

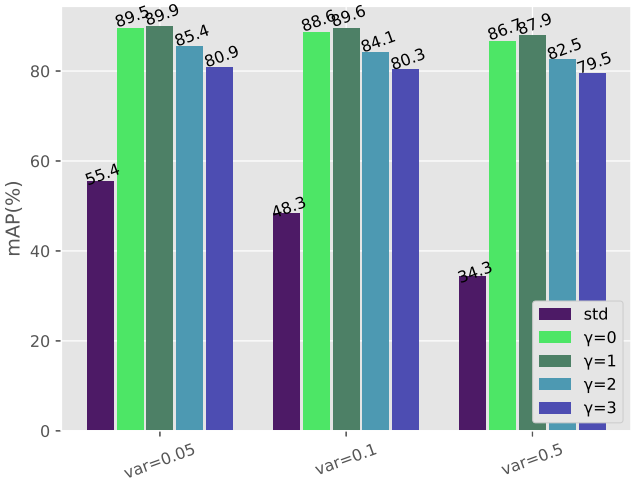


(a) Result on ResNet-101

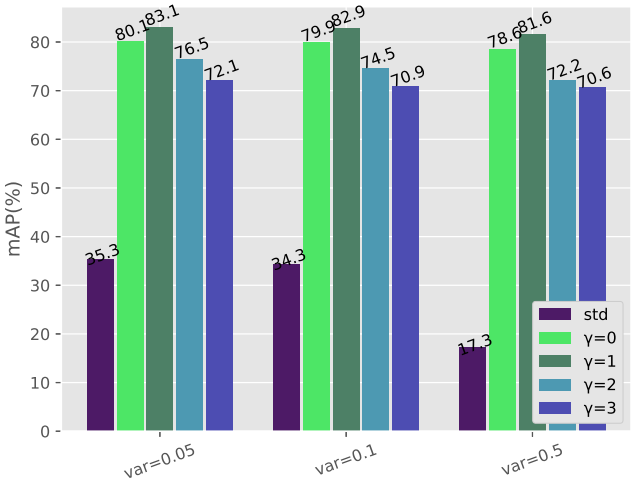


(b) Result on VGG16

Figure 8. Changes of mAP with different noise attack (with NAN). The var refers to variance σ^2 of speckle noise with zero mean. Std refers to the standard pipeline (baseline) while γ is the exponential parameter.



(a) Result on ResNet-101



(b) Result on VGG16

Figure 9. Changes of mAP with different noise attack (with NAN and NB). The var refers to variance σ^2 of speckle noise with zero mean. Std refers to the standard pipeline (baseline) while γ is the exponential parameter.

The combination of NAN and NB improves the performance by eliminating most of the Sim error in corpse as well as reducing the location error in three classes.

9. Feature maps

During test stage, we add speckle noise to the test sample, feeding them into the detectors. We visualize the feature maps from several backbone layers of ResNet-101 with

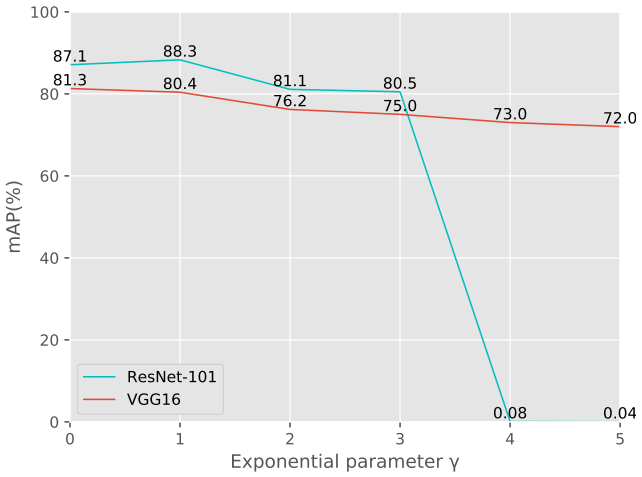


Figure 10. Changes of mAP with different exponential parameter γ.

both original Faster R-CNN and our model (with the combination of NAN and NA). In Figure 11, it is obvious that the speckle noise deteriorates the original detector from the upstream layers such as Res_1, especially for corpse objects. Without robustness of speckle noise, the original detector cannot distinguish the object from background under noise attack, which impedes the generation of the activate region from downstream layers such as Res_3. The informative features degrades layer by layer.

However, our model can easily differentiate the noise and objects. From shallow layers such as Res_1, our model mitigates the effect of speckle noise so that the deep layers suffice to predict the region precisely, especially in corpse and plane wreckage. The activated informative features are crucial for the following regression and classification tasks.

References

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

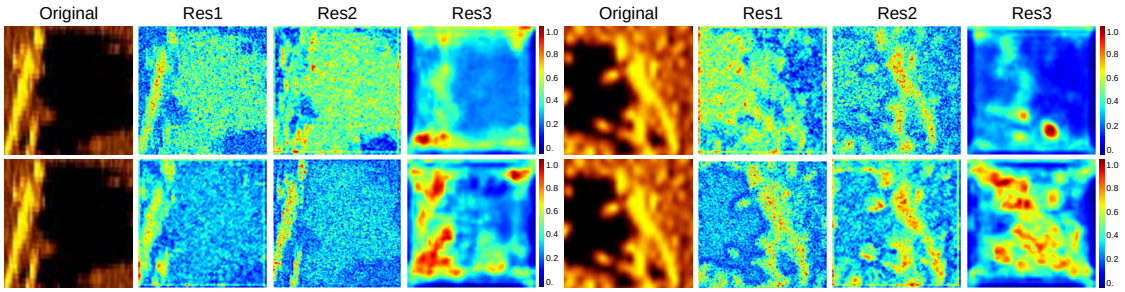
[2] J. Lee Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

[3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

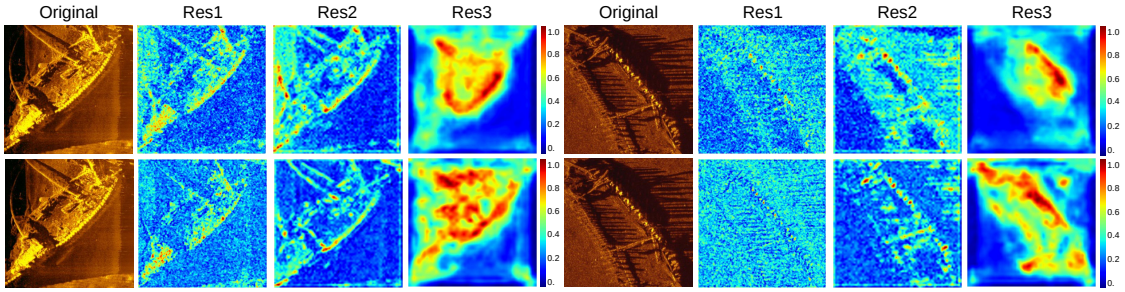
[4] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

[5] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

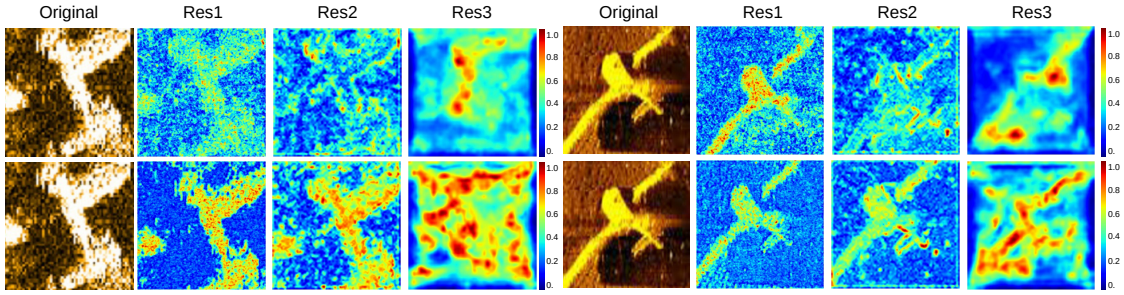
[6] R. L. Wasserstein, N. A. Lazar, et al. The asas statement on p-values: context, process, and purpose. *The American Statistician*, 70(2):129–133, 2016.



(a) Feature maps of corpse



(b) Feature maps of shipwreck



(c) Feature maps of plane wreckage

Figure 11. Feature maps of several layers (specifically in ResNet-101, they are output of Res_1, Res_2, Res_3 blocks). The feature maps are extracted from original Faster R-CNN (first row) and Faster R-CNN equipped with NAN and NB (second row), respectively. Every single feature map here is the result of the channel-wise mean of its corresponding layer.

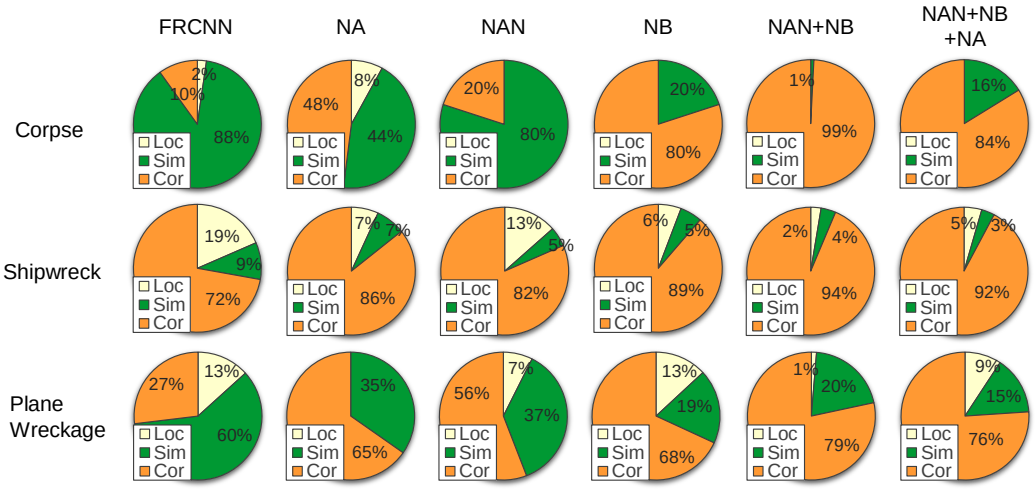


Figure 12. Error analyses distibuted on three classes. Test set is attacked by speckle noise (μ=0, σ^2=0.5). Distribution of top-ranked detections include Cor (correct), Loc (misaligned localization) and Sim (confusion with a wrong class).