SAVeD: Learning to Denoise Low-SNR Video for Improved Downstream Performance

Supplementary Material

We present additional experimental results as ablations Sec. (A), additional implementation details (Sec. B), and additional visualizations (Sec. C).

Benefits and risks of this technology. Improving classification, tracking, and counting in sonar and ultrasound videos is useful across medical, ecological, and other fields. Counting fish with sonar allows for a non-invasive way to measure population size, which can then be used for conservation and ecological efforts, for understanding effects of climate change, and for monitoring human fishing behavior for economical reasons. Improving classification in ultrasound videos, too, paves a path for more automated diagnosis. Risks, though, are inherent in both tracking applications and applications of sensitive data. Care must be taken when using these models, so that they are not used blindly without human intervention to make decisions.

A. Additional Experimental Results

A.1. Additional CFC22 Ablation Results

As in the main paper, we evaluate CFC22 on the detection val/test splits, and show results using mAP_{50} across the dataset splits. We look at the effect of bottleneck size in the hourglass network, traditional augmentations, input resolution size, and reconstruction targets on how the trained denoiser affects downstream detection performance. We also look at the effect of downstream task performance when using the reconstruction target alone ($S_{t,T}$) compared with using the learned reconstruction ($\hat{S}_{t,T}$).

Bottlenecks size. For all experiments on CFC22, we use a default input size of 1024hx512w, reconstruction target as PFDwT1, mean-squared error (MSE) loss, and we train the denoiser for 20 epochs. Here the hourglass network remains 2 layers, with the number of input channels as 512, but the number of channels in the middle layer changes. We notice that for training, larger (less-restrictive) bottlenecks yield higher performance. For val and test, though, bottleneck sizes over 64 improve performance, but the differences between 128 and 512 is worse for val and negligible for test. Results can be seen in Tab. 5a.

Resolution size. We vary the input resolution size to train the denoiser and notice higher performance for train and test when higher resolutions are used, seen in Tab. 5b. We hypothesized that higher resolution size would make the denoiser more stable for downstream detections because higher resolution sizes would mean that removing entire fish (*i.e*. small fish) would be less probable. It is interesting to note that the highest resolution size 2048x1024 for val led to lower detection performance than that of resolution size 1024x512. We note, though, that higher resolutions lead to smaller batch sizes and longer training time.

Traditional Augmentations. We apply salt-and-pepper noise, gaussian-blur, motion-blur, brightness, and erasing from the kornia.Augmentations library. We apply these augmentations when training the denoiser. We do *not* apply these augmentations when training downstream tasks. We found that no traditional augmentations to train the denoiser, though, improve downstream performance. Results can be seen in Tab. 5c.

Reconstruction Targets. We experimented with a handful of reconstruction targets:

Frame difference—such as absolute difference ($S_{|d|} = |I_t - I_{t+T}|$) or raw difference ($S_d = I_t - I_{t+T}$)—has been used in other self-supervised works as a spatiotemporal reconstruction target [47]. This works well in video where the movement in the background is less than the foreground movement. For our experiments, we use absolute difference as frame difference.

Raw frame (I_t) predicts the input (identity) frame alone.

Background subtraction (bs) We approximate the background frame, $\bar{I}_v$, as the mean aggregate of video over time. This is based on the approximation that objects of interest are sparse in terms of space and time. The mean frame is subtracted from every frame in the video ($S_d = (I_v)_t - \bar{I}_v$).

Positive Frame Difference with current frame (PFDwTN). We discuss this in more detail in the main paper, Sec. 3.2. We experimented with T=2 (PFDwT2) and T=1 (PFDwT1), ultimately selecting T=1.

Standard Deviation across all frames (σ) is taken across all of the frames loaded in a window of continuous frames, $\sigma(I_{t-N} : I_{t+N})$ where 2N+1 is the size of the window. We experimented with N=1 and N=2.

*Sum frames minus N*background* ($\Sigma - N\bar{I}_v$) sums all of the frames in a window size N and takes the positive difference $N * \bar{I}_v$ where $\bar{I}_v$ is the mean frame of all frames in a video: $\max(0, (\sum_t^T I_t) - N\bar{I}_v)$. We experimented with window sizes N=3 and N=5.

Visualizations of all of these can be seen in Fig. 10

A.2. POCUS Per-Class Performance

SAVeD performs well across all classes (COVID, Pneumonia, and Regular) in the POCUS dataset (Fig. 11). For Pneumonia, precision levels across all methods were lower than

Signal Modification	AE	mAP_{50}		
		Train	Val	Test
Signal Modification w/o Denoising Network				
Raw (I_t)	✗	79.6	69.6	54.2
σ	✗	79.8	69.4	72.5
$\Sigma - 5\bar{I}$	✗	78.3	67.6	71.7
PFDwT1	✗	80.2	66.9	68.2
PFDwT2	✗	81.2	68.1	63.0
Signal Modification w/ Denoising Network				
Raw (I_t)	✓	81.5	68.4	73.4
σ	✓	82.2	70.0	73.5
$\Sigma - 5\bar{I}$	✓	79.8	68.1	71.7
PFDwT1	✓	**83.5**	**70.6**	**77.6**
PFDwT2	✓	82.2	68.5	71.4

Table 4. **Effect of Different Motion Enhancements with and without SAVeD's Autoencoder Network (AE) on CFC22.** All detectors that leverage the AE have superior performance to those that use only the motion-enhanced target on the test set. The modified signal is used as the reconstruction target for the denoising autencoder when it is present, and is the input signal for the downstream task when the autoencoder is not used. All results are on CNNs with skip connections with resolution 1024 and bottleneck 512.

for other classes. Pneumonia false negatives are more often categorized as Regular than they are Covid across all denoising methods.

B. Implementation Details

B.1. SAVeD Architecture Details

Our method uses a series of convolution blocks with skip connections as an encoder Φ, a bottleneck (hourglass network) Θ, and a reconstruction decoder Ψ. Architectural details about each of these are shown in Tab. 6. For more implementation details, the code is publicly available here.

B.2. SAVeD Hyperparameter Comparisons

The hyperparameters for our method are in Tab. 8. All DAE models are trained until the training loss converges on 2 NVIDIA RTX 4090 GPUs.

B.3. CFC22 Detector Details

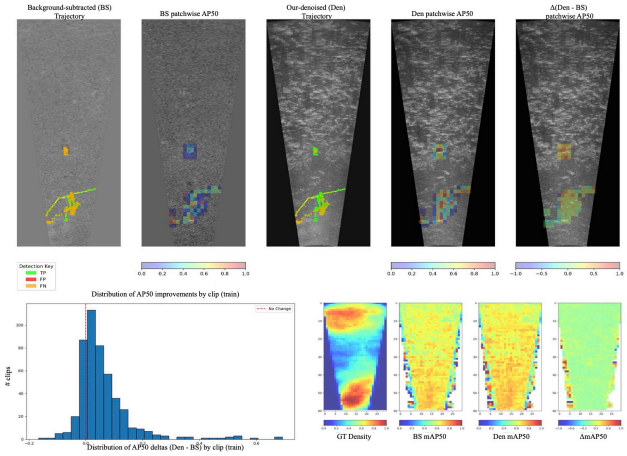
We fine-tune a YoloV5-small model pretrained on COCO using the default training settings from Ultralytics over 5 epochs with a batch size of 16. As in Kay et al. [24], we resize all inputs to have 896 pixels as their longest side; the learning rate is 0.0025. We select the best model checkpoint based on validation mAP_{50}. We train on two NVIDIA RTX A6000 GPUs. We recognize that the number of epochs (5) differs from the number of epochs in the original paper (150), and that is intentional. The reasoning is two-fold: 1.) CFC22++ Val and Test Performance after 5 epochs are $< 1\%$ lower than Val and Test Performance after 150 epochs, therefore our denoised improvement beats the CFC22++ method also after CFC22++ is trained for 150 epochs while the detector model based on SAVeD frames is trained for 5 epochs; 2.) We wanted to show that a very simple detector could be used as a result of passing in denoised frames.

B.4. CFC22 Tracker Details

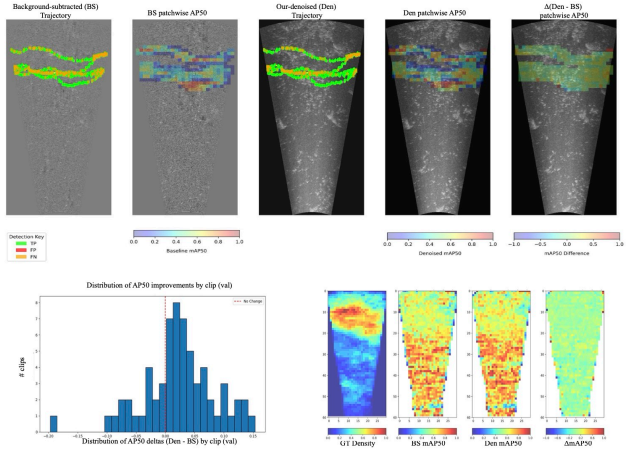
We use a pretrained ByteTrack tracker with hyperparameters selected as the optimal hyperparameters for tracking performance on the validation set. Max age, the time until a missing or occluded object is assigned a new id, is 20; Min hits, the minimum number of frames with a track for the track to be considered valid, is 11; IOU threshold, the iou required for an object to be considered the same in the subsequent frame, is 0.01.

C. Visualizations

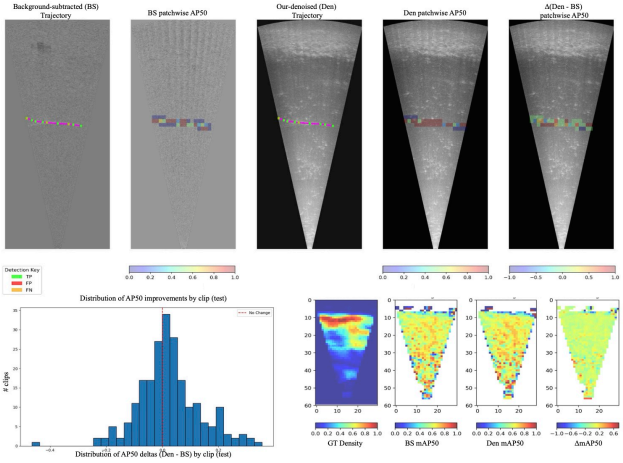
Additional visualizations of the denoising performance on fish in sonar (CFC22[24]) can be seen in Fig. 15).



(a) One clip from the CFC22-train river. You can can see the trajectory and patchwise detection performance improves after denoising. Overall, the biggest denoising gains appear to be at the edges of the cone, where fish are known to be small (entering/exiting) but moving.



(b) One clip from the CFC22-val river. The denoising gain is smaller and therefore more difficult to see here.



(c) One clip from the CFC22-test river.

Figure 9. **Denoising-improved detection leads to better tracks**. On the single-clip trajectory plots, orange dots indicate false negatives, green dots indicate true positives, red indicates false positives.

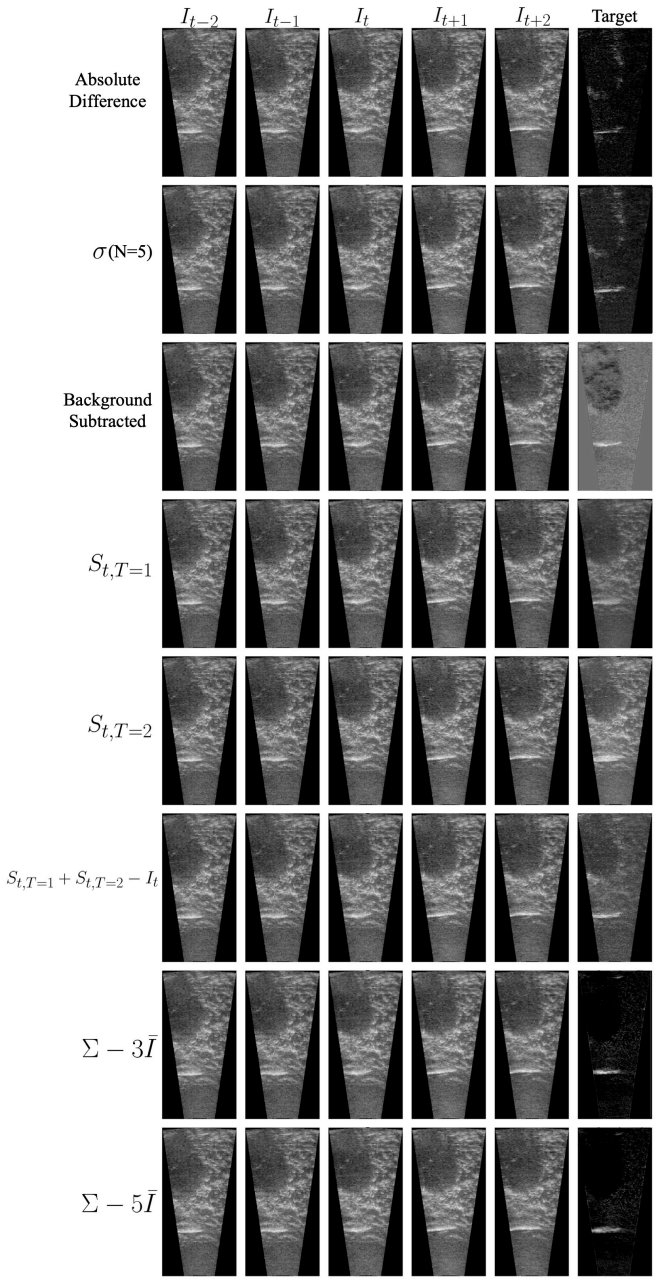


Figure 10. **Reconstruction Targets**. The window T=5 set of frames is shown with each reconstruction target we experimented with on CFC22. While $\Sigma - N\bar{I}$ frames appear strong in this example, we found that empirically they struggled to capture fish that did not move significantly between frames.

Bottleneck	Train mAP_{50}	Val mAP_{50}	Test1 mAP_{50}
64	79.1	68.6	71.6
128	80.0	69.2	**72.6**
512	**81.6**	**69.4**	**72.6**

(a) **Bottleneck size.** A larger bottleneck outperforms overly-constricted networks. All results are from CNNs with no skip connections and non-residual blocks.

Resolution	Train mAP_{50}	Val mAP_{50}	Test1 mAP_{50}
512	**81.3**	**69.2**	71.9
1024	79.1	68.6	71.6
2048	80.1	68.1	**72.1**

(b) **Resolution size.** There is no clear optimal - in terms of train and val, the smallest resolution size is the best; however, in terms of test, the largest resolution size is optimal. Note that higher resolutions also lead to longer training times.

Augmentations	Train mAP_{50}	Val mAP_{50}	Test1 mAP_{50}
$\text{saltpepper}_{0.25}$	81.2	68.5	72.2
$\text{saltpepper}_{0.5}$	**83.7**	69.7	75.1
$\text{saltpepper}_{0.75}$	81.4	69.2	72.8
$\text{gaussianblur}_{0.25}$	82.1	69.9	74.8
$\text{gaussianblur}_{0.5}$	81.3	68.9	75.0
$\text{gaussianblur}_{0.75}$	83.5	68.4	75.6
$\text{motionblur}_{0.25}$	83.5	68.3	76.5
$\text{motionblur}_{0.5}$	81.2	68.2	74.7
$\text{motionblur}_{0.75}$	**83.7**	69.6	73.9
$\text{brightness}_{0.25}$	**83.7**	69.8	74.7
$\text{brightness}_{0.5}$	82.2	69.0	73.9
$\text{brightness}_{0.75}$	83.6	69.7	76.8
$\text{erase}_{0.25}$	82.0	68.7	68.0
$\text{erase}_{0.5}$	81.1	68.7	75.6
$\text{erase}_{0.75}$	77.4	59.3	62.4
No augmentations	83.5	**70.6**	**77.6**

(c) **Augmentations.** Augmentations appear to degrade performace. All augmentation experiments are named as $augmentation_{probability}$. All networks are CNNs with skip connections with resolution 1024 and bottleneck 512.

Target	Train mAP_{50}	Val mAP_{50}	Test1 mAP_{50}
Raw*	81.5	68.4	73.4
Absolute Difference $\lvert I_t - I_{t+1} \rvert$	81.6	69.2	73.5
Sigma(N=5)	78.8	69.2	72.8
$\hat{S}_{t,T=1}$*	82.7	70.0	74.0
$\hat{S}_{t,T=2}$*	82.8	**70.6**	73.0
$\hat{S}_{t,T=2} + \hat{S}_{t,T=1} - I_t$*	**83.7**	69.2	**74.6**
$\Sigma - 3\bar{I}$	80.3	68.3	69.0
$\Sigma - 5\bar{I}$	80.7	68.7	72.0

(d) **Reconstruction targets.** Reconstruction targets including both the original frame and the next or previous frames do better than reconstruction targets incorporating information from just one. Reconstruction targets with the current frame in have *. All results are on CNNs with resolution 1024 and bottleneck 512 with no skip connection.

Architectures	Train mAP_{50}	Val mAP_{50}	Test1 mAP_{50}
Autoencoder	82.6	68.9	67.8
CNN-fine	82.7	69.1	74.0
CNN-SKIP	**83.5**	**70.6**	**77.6**
CNN-residual	**83.5**	69.2	73.1
CNN-resnet-block	79.8	70.0	73.6
UNet-downscaled	82.1	69.1	75.8
UNet	81.2	70.0	73.9
UNet3D	79.0	67.0	66.9

(e) **Denoising backbone architecture.** All experiments have our target from equation 2 ($\hat{S}_{t,T=1}$) as their target. Networks are ordered from smallest (in terms of parameters and TFLOPs) to largest – it is interesting to note that as model size increases, performance does not necessarily increase. We see the top performer is the CNN-SKIP architecture.

Table 5. **Additional denoise-detection ablations on CFC22.** All values are generated via the detection stage of our pipeline. All reconstruction targets are sized 1024 x 512 unless otherwise stated. We report the mAP_{50} of the *combined* background-subtracted and target reconstruction frame unless otherwise noted. Default settings are marked in gray.

Encoder		
Type	Input shape	Output shape
Conv_block	(1,1024,512)	(16, 1024, 512)
Pooling	(16, 1024, 512)	(16, 512, 256)
Skip	(16, 1024, 512)	(16, 512, 256)
Conv_block	(16, 512, 256)	(32, 512, 256)
Pooling	(32, 512, 256)	(32, 256, 128)
Skip	(32, 512, 256)	(32, 256, 128)
Conv_block	(32, 256, 128)	(64, 156, 128)
Pooling	(64, 156, 128)	(64, 128, 64)
Skip	(64, 156, 128)	(64, 128, 64)
Conv_block	(64, 128, 64)	(128, 128, 64)
Pooling	(128, 128, 64)	(128, 64, 32)
Skip	(128, 128, 64)	(128, 64, 32)
Conv_block	(128, 64, 32)	(256, 64, 32)
Pooling	(256, 64, 32)	(256, 32, 16)
Skip	(256, 64, 32)	(256, 32, 16)
Conv_block	(256, 32, 16)	(512, 32, 16)
Pooling	(512, 32, 16)	(512, 16, 8)
Skip	(512, 32, 16)	(512, 16, 8)

Decoder		
Type	Input shape	Output shape
Upsample_block	(512, 16, 8)	(256, 32, 16)
Skip_connect	(256, 32, 16)	(768, 32, 16)
Conv_block	(768, 32, 16)	(512, 32, 16)
Upsample_block	(512, 32, 16)	(256, 64, 32)
Skip_connect	(256, 64, 32)	(512, 64, 32)
Conv_block	(512, 64, 32)	(256, 64, 32)
Upsample_block	(256, 64, 32)	(128, 128, 64)
Skip_connect	(128, 128, 64)	(256, 128, 64)
Conv_block	(256, 128, 64)	(128, 128, 64)
Upsample_block	(128, 128, 64)	(64, 256, 128)
Skip_connect	(64, 256, 128)	(128, 256, 128)
Conv_block	(128, 256, 128)	(64, 256, 128)
Upsample_block	(64, 256, 128)	(32, 512, 256)
Skip_connect	(32, 512, 256)	(64, 512, 256)
Conv_block	(64, 512, 256)	(32, 512, 256)
Upsample_block	(32, 512, 256)	(1, 1025, 512)

Table 6. **Architecture details of the encoder, bottleneck, and decoder of SAVeD.** "Conv_block" is a basic convolutional block composed of 3x3 convolution with padding side of 1 and ReLU activation. "Skip" is a skip connection (stored to be input into the decoder) composed by maxpooling and then running a 1x1 convolution. "Upsample_block" is a 2D ConvTranspose with a 2x2 kernel and a stride of 2 and a ReLU activation. "Skip_connect" is the concatenation of the output from Upsample_block+Conv_block and the "Skip" corresponding to the same layer saved by the encoder. Note that this architecture is on input size of 1024x512.

Encoder		
Type	Input shape	Output shape
Conv_block	(3, 1024, 512)	(16, 1024, 512)
Pooling	(16, 1024, 512)	(16, 512, 256)
Conv_block	(16, 512, 256)	(32, 512, 256)
Pooling	(32, 512, 256)	(32, 256, 128)
Conv_block	(32, 256, 128)	(64, 156, 128)
Pooling	(64, 156, 128)	(64, 128, 64)
Conv_block	(64, 128, 64)	(128, 128, 64)
Pooling	(128, 128, 64)	(128, 64, 32)
Conv_block	(128, 64, 32)	(256, 64, 32)
Pooling	(256, 64, 32)	(256, 32, 16)
Conv_block	(256, 32, 16)	(512, 32, 16)
Pooling	(512, 32, 16)	(512, 16, 8)
Decoder		
Type	Input shape	Output shape
Bilinear_upsample_block	(512, 16, 8)	(512, 32, 16)
Conv_block	(512, 32, 16)	(256, 32, 16)
Bilinear_upsample_block	(256, 32, 16)	(256, 64, 32)
Conv_block	(256, 64, 32)	(128, 64, 32)
Bilinear_upsample_block	(128, 64, 32)	(128, 128, 64)
Conv_block	(128, 128, 64)	(64, 128, 64)
Bilinear_upsample_block	(64, 128, 64)	(64, 256, 128)
Conv_block	(64, 256, 128)	(32, 256, 128)
Bilinear_upsample_block	(32, 256, 128)	(32, 512, 256)
Conv_block	(32, 512, 256)	(16, 512, 256)
Bilinear_upsample_block	(16, 512, 256)	(16, 1024, 512)
Conv_block	(16, 1024, 512)	(1, 1024, 512)

Table 7. **Architecture details of the vanilla autoencoder.** "Conv_block" is a basic convolutional block composed of 3x3 convolution with padding side of 1 and ReLU activation. "Bilinear_upsample_block" is a Bilinear Upsample kernel with a scale factor of 2 and align corners set to True. Note that this architecture is on input size of 1024x512.

Dataset	Resolution	Target	Epochs	Batch size	Learning Rate	Optimizer	Scheduler
CFC22	(1024,512)	$S_{t,T=1}$	20	16	0.0005	AdamW	Plateau f=0.1 pat=2
POCUS	(1024,512)	$S_{t,T=1}$	120	8	0.0005	AdamW	Step ss=2, $\gamma = 0.05$
BUV	(1024,1024)	inverse$(S_{t,T=1})$	40	8	0.0005	AdamW	Step ss=2, $\gamma = 0.05$
Fluo	(1024,1024)	I_t	1000	8	0.0005	AdamW	Step ss=2, $\gamma = 0.05$

Table 8. **SAVeD Hyperparameters.** Note "inverse$(S_{t,T=1})$"$= \min(0, I_t - I_{t-T}) + I_t + \min(0, I_t - I_{t+T})$. f=Factor, pat=Patience, ss=Step size.

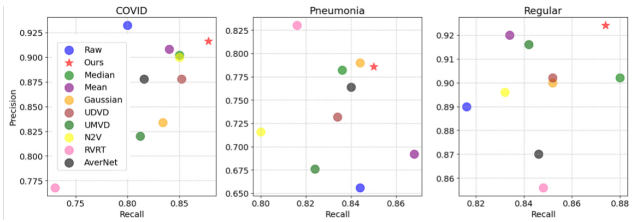


Figure 11. SAVeD (starred) has high precision and high recall across all POCUS classes.

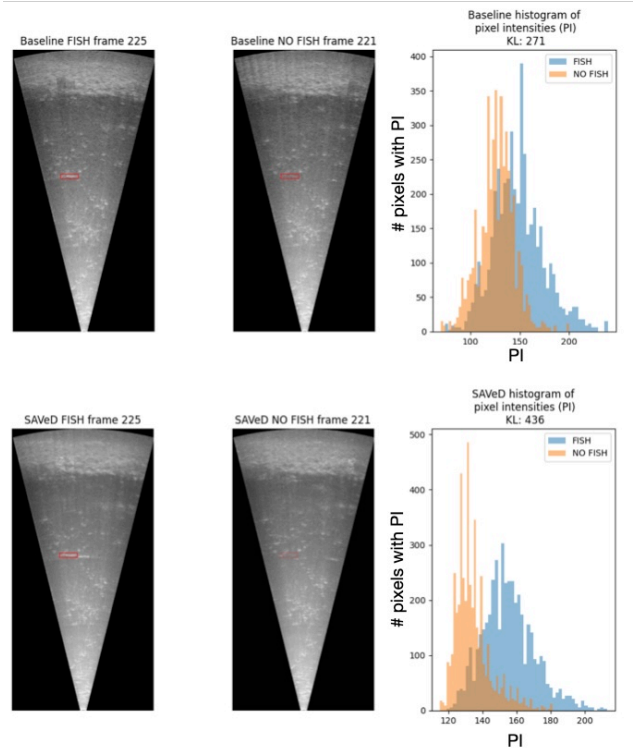


Figure 12. **Visualization of** FBD. Both images on the left are noisy images. The image on the far left has a fish located in the red bounding box. The image in the middle is a frame from the same video clip but with no fish in the red box. The histogram compares the pixel intensity values of the pixels within the bounding boxes. We can see these distributions, while overlapping, are distinct.

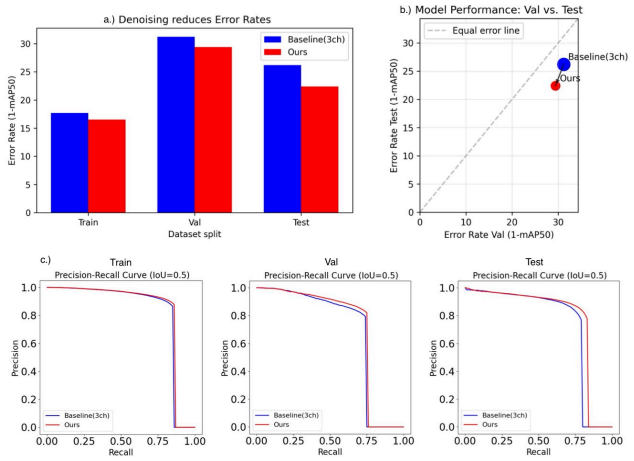


Figure 13. **Denoising lowers detections error-rates by improving precision and recall** (a) shows baseline detection error (1-mAP_{50}) compared to our detection error after our denoising preprocessing step. For all splits train, val, and test, denoising results in lower error. (b) compares error rates from the validation set (x-axis) to error rates from the test set (y-axis) to see how denoising impacts each split. There is a 5.8% reduction in error in the val set and a 14.5% reduction in error on the test set. (c) Shows inverted Precision-Recall plots for each CFC22 dataset split – precision and recall both improve for all splits.

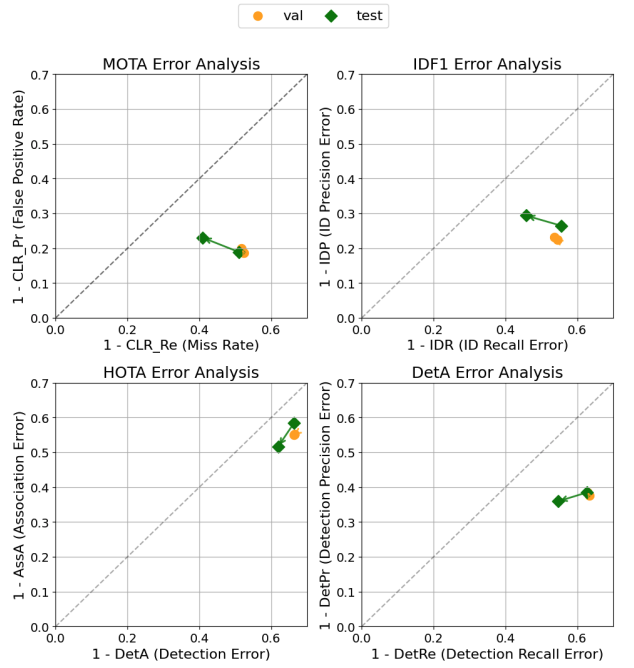


Figure 14. **Breakdown of track performance improvements for CFC22 val and test**. We can see test improves far more than val, as is standard for the CFC22 dataset.

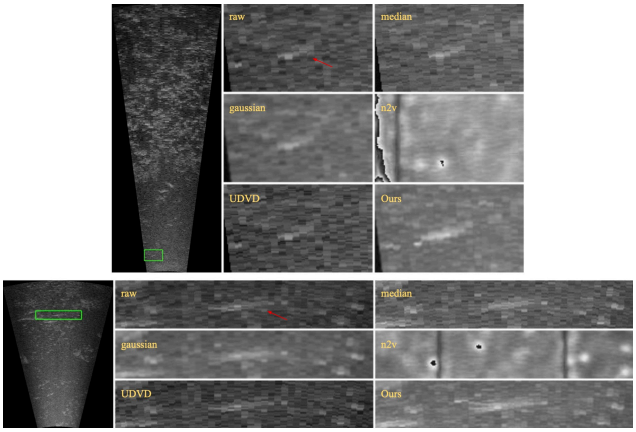


Figure 15. **Additional visualizations of denoising methods on CFC22**

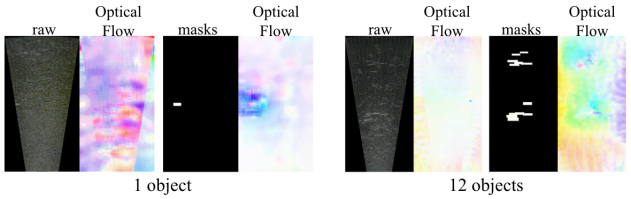


Figure 16. **RAFT [50] on CFC22 imagery and bounding box masks.** On the left, we can see the optical flow signal does not find the fish. When looking at the motion from the bounding-box mask of the fish (making the background movement stationary), the optical flow signal area is far greater than the actual area of the fish. On the right are frames (with fish and corresponding bounding-box masks), when there are 12 fish in the frame at once. Again, optical flow's signal is weak with the fish movement compared to the background. With the mask movement, optical flow signals cluster in groups of masked fish, but individuals are difficult to distinguish.

References

[1] Mary Aiyetigbo, Alexander Korte, Ethan Anderson, Reda Chalhoub, Peter Kalivas, Feng Luo, and Nianyi Li. Unsupervised microscopy video denoising. In *IEEE/CVF Computer Vision and Pattern Recognition Workshop(CVPRW)*, 2024. 2, 6, 7

[2] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *Proceedings of the 36th International Conference on Machine Learning*, pages 524–533. PMLR, 2019. 2

[3] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, page 899–907, Red Hook, NY, USA, 2013. Curran Associates Inc. 8

[4] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. 5, 7

[5] J Born, N Wiedemann, M Cossio, C Buhre, G Brändle, K Leidermann, and A Aujayeb. L2 accelerating covid-19 differential diagnosis with explainable ultrasound image analysis: an ai tool. *Thorax*, 76(Suppl 1):A230–A231, 2021. 5, 7

[6] Jannis Born, Nina Wiedemann, Manuel Cossio, Charlotte Buhre, Gabriel Brändle, Konstantin Leidermann, Avinash Aujayeb, Michael Moor, Bastian Rieck, and Karsten Borgwardt. Accelerating detection of lung pathologies with explainable ultrasound image analysis. *Applied Sciences*, 11 (2):672, 2021. 5, 7

[7] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020. 3

[8] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, Dequan Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019. 3

[9] Benjamin J. Choi, Griffin Milsap, Clara A. Scholl, Francesco Tenore, and Mattson Ogg. Targeted adversarial denoising autoencoders (tada) for neural time series filtration. *arXiv preprint arXiv:2501.04967*, 2025. 2

[10] Shaveta Dargan, Munish Kumar, Maruthi Rohit Ayyagari, and Gulshan Kumar. A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 27(4):1071–1092, 2020. 1

[11] Mariette Dupuy, Marie Chavent, and Rémi Dubois. mdae: modified denoising autoencoder for missing data imputation. In *arXiv preprint arXiv:2411.12847*, 2024. 2

[12] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 5, 7

[13] Eyrun Eyjolfsdottir, Steve Branson, Xavier P. Burgos-Artizzu, Eric D. Hoopfer, Jason Schor, David J. Anderson, and Pietro Perona. Detecting social actions of fruit flies. In *European Conference on Computer Vision (ECCV)*, 2014. 4

[14] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32:1231–1237, 2013. 3

[15] Martin A. Giese and Tomaso Poggio. Cognitive neuroscience: neural mechanisms for the recognition of biological movements. *Nature Reviews Neuroscience*, 4(3):179–192, 2003. 1

[16] Rafael C Gonzalez and Richard E Woods. *Digital Image Processing*. Prentice-Hall, Inc., Upper Saddle River, 3rd edn. edition, 2006. 6, 7

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, 2022. 8

[18] Gregory Holste, Evangelos K. Oikonomou, Bobak J. Mortazavi, Zhangyang Wang, and Rohan Khera. Efficient deep learning-based automated diagnosis from echocardiography with contrastive self-supervised learning. *Communications Medicine*, 4:133, 2024. 1

[19] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14781–14790, 2021. 2, 4

[20] Yifan Huang, Weixiang Li, and Fei Yuan. Speckle noise reduction in sonar image based on adaptive redundant dictionary. *Journal of Marine Science and Engineering*, 8(10): 761, 2020. 2

[21] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Advances in Neural Information Processing Systems*, 2018. 3

[22] Yeong Il Jang, Keuntek Lee, Gu Yong Park, Seyun Kim, and Nam Ik Cho. Self-supervised image denoising with downsampled invariance loss and conditional blind-spot network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12196–12205, 2023. 2, 4

[23] Bo Jiang, Jinxing Li, Yao Lu, Qing Cai, Huaibo Song, and Guangming Lu. Efficient image denoising using deep learning: A brief survey. *Information Fusion*, 92:1–18, 2025. 2

[24] Justin Kay, Peter Kulits, Suzanne Stathatos, Siqi Deng, Erik Young, Sara Beery, Grant Van Horn, and Pietro Perona. The caltech fish counting dataset: A benchmark for multiple-object tracking and counting. In *European Conference on Computer Vision (ECCV)*, 2022. 4, 5, 6, 7, 2

[25] Daniel Khalil, Christina Liu, Pietro Perona, Jennifer J Sun, and Markus Marks. Learning keypoints for multi-agent behavior analysis using self-supervision. *arXiv preprint arXiv:2409.09455*, 2024. 1

[26] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void: Learning denoising from single noisy images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2129–2137, 2019. 2, 4, 6, 7

[27] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 1951. 4

[28] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *Advances in Neural Information Processing Systems*, 2019. 2, 4

[29] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *International Conference on Learning Representations*, 2022. 8

[30] Yawei Li, Yulun Zhang, Radu Timofte, Luc Van Gool, Zhijun Tu, Kunpeng Du, Hailing Wang, Hanting Chen, Wei Li, Xiaofei Wang, Jie Hu, Yunhe Wang, Xiangyu Kong, Jinlong Wu, Dafeng Zhang, Jianxing Zhang, Shuai Liu, Furui Bai, Chaoyu Feng, Hao Wang, Yuqian Zhang, Guangqi Shao, Xiaotao Wang, Lei Lei, Rongjian Xu, Zhilu Zhang, Yunjin Chen, Dongwei Ren, Wangmeng Zuo, Qi Wu, Mingyan Han, Shen Cheng, Haipeng Li, Ting Jiang, Chengzhi Jiang, Xinpeng Li, Jinting Luo, Wenjie Lin, Lei Yu, Haoqiang Fan, Shuaicheng Liu, Aditya Arora, Syed Waqas Zamir, Javier Vazquez-Corral, Konstantinos G. Derpanis, Michael S. Brown, Hao Li, Zhihao Zhao, Jinshan Pan, and Jiangxin Dong. Ntire 2023 challenge on image denoising: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1905–1921, 2023. 2

[31] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc Van Gool. Recurrent video restoration transformer with guided deformable attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 7

[32] Zhi Lin, Junhao Lin, Lei Zhu, Huazhu Fu, Jing Qin, and Liansheng Wang. A new dataset and a baseline model for breast lesion detection in ultrasound videos. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, pages 614–623, Cham, 2022. Springer Nature Switzerland. 5, 7

[33] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. *CVPR*, pages 6536—-6545, 2018. 1

[34] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129(2): 548–578, 2021. 5

[35] Adrià Marcos Morales, Matan Leibovich, Sreyas Mohan, Joshua Lawrence Vincent, Piyush Haluai, Mai Tan, Peter Crozier, and Carlos Fernandez-Granda. Evaluating unsupervised denoising requires unsupervised metrics. In *Proceedings of the 40th International Conference on Machine Learning*, pages 23937–23957. PMLR, 2023. 2

[36] Markus Marks, Manuel Knott, Neehar Kondapaneni, Elijah Cole, Thijs Defraeye, Fernando Perez-Cruz, and Pietro Perona. A closer look at benchmarking self-supervised pre-training with image classification. *arXiv preprint arXiv:2407.12210*, 2024. 1

[37] Oleg V. Michailovich and Allen Tannenbaum. Despeckling of medical ultrasound images. *IEEE Trans Ultrason Ferroelectr Freq Control*, 1:64–78, 2013. 2

[38] Fengpu Pan, Jiangtao Wen, and Yuxing Han. Snapshot compressed imaging based single-measurement computer vision for videos. *arXiv preprint arXiv:2501.15122*, 2025. 2

[39] Adrià Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Ross Hemsley, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Patraucean, Florent Altché, Michal Valko, Jean-Bastien Grill, Aäron van den Oord, and Andrew Zisserman. Broaden your views for self-supervised video learning. *ICCV*, pages 1255–1265, 2021. 1

[40] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016. 5

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 2

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, 2015. 7, 8

[43] Serim Ryou and Pietro Perona. Weakly supervised keypoint discovery. *arXiv preprint arXiv:2109.13423*, 2021. 3

[44] Dev Yashpal Sheth, Sreyas Mohan, Joshua Vincent, Ramon Manzorro, Peter A. Crozier, Mitesh M. Khapra, Eero P. Simoncelli, and Carlos Fernandez-Granda. Unsupervised deep video denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 4, 5, 6, 7

[45] Yuge Shi, Imant Daunhawer, Julia E. Vogt, Philip Torr, and Amartya Sanyal. How robust are pre-trained models to distribution shift? In *ICML 2022: Workshop on Spurious Correlations, Invariance, and Stability*, 2022. 1

[46] Abhishek Sinha and Shreya Singh. Zero-shot active learning using self supervised learning. *arXiv preprint arXiv:2401.01690*, 2024. 8

[47] Jennifer J Sun, Serim Ryou, Roni Goldshmid, Brandon Weissbourd, John Dabiri, David J Anderson, Ann Kennedy, Yisong Yue, and Pietro Perona. Self-supervised keypoint discovery in behavioral videos. *CVPR*, 2022. 1, 3

[48] Kalaivani Sundararajan and Damon L. Woodard. Deep learning for biometrics: A survey. *ACM Computing Survey*, 51(3): 1–34, 2018. 1

[49] Matias Tassano, Julie Delon, and Thomas Veit. Dvdnet: A fast network for deep video denoising. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 1805–1809. IEEE, 2020. 2

[50] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402—-419, Berlin, Heidelberg, 2020. 9

[51] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022. 8

[52] Vladimír Ulman, Martin Maška, Klas E. G. Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miloš Radojevic, Ihor Smal, Karl Rohr, Joakim Jaldén, Helen M. Blau, Oleh Dzyubachyk, Boudewijn Lelieveldt, Pengdong Xiao, Yuexiang Li, Siu-Yeung Cho, Alexandre C. Dufour, Jean-Christophe Olivo-Marin, Constantino Carlos Reyes-Aldasoro, Jose A. Solis-Lemus, Robert Bensch, Thomas Brox, Johannes Stegmaier, Ralf Mikut, Steffen Wolf, Fred A. Hamprecht, Tiago Esteves, Pedro Quelhas, Ömer Demirel, Lars Malmström, Florian Jug, Pavel Tomančák, Erik Meijering, Arrate Muñoz-Barrutia, Michal Kozubek, and Carlos Ortiz-de Solorzano. An objective comparison of cell-tracking algorithms. *Nature Methods*, 14:1141–1152, 2017. 5

[53] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 4

[54] Zichun Wang, Ying Fu, Ji Liu, and Yulun Zhang. LG-BPN: Local and global blind-patch network for self-supervised real-world denoising. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 6, 7

[55] Alistair Weld, Giovanni Faoro, Luke Dixon, Sophie Camp, Arianna Menciassi, and Stamatia Giannarou. Standardisation of convex ultrasound data through geometric analysis and augmentation. *arXiv preprint arXiv:2502.09482*, 2025. 3

[56] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8): 1106–1125, 2019. 2

[57] Yanyang Yan, Qingbo Wu, Bo Xu, Jingang Zhang, and Wenqi Ren. Vdflow: Joint learning for optical flow and video deblurring. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, page 878–879, 2020. 2

[58] Qiulong Yang and Kunde Yang. Seasonal comparison of underwater ambient noise observed in the deep area of the south china sea. *Applied Acoustics*, 172:107672, 2021. 2

[59] Zhiwei Yang, Jing Liu, Zhaoyang Wu, Peng Wu, and Xiaotao Liu. Video event restoration based on keyframes for video anomaly detection. *CVPR*, 2023. 1

[60] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 2

[61] Haiyu Zhao, Lei Tian, Xinyan Xiao, Peng Hu, Yuanbiao Gou, and Xi Peng. Avernet: All-in-one video restoration for time-varying unknown degradations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2, 7