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

Suzanne Stathatos Michael Hobley Pietro Perona[=] Markus Marks[=]
Caltech
{sstathat, mahobley, perona, marks}@caltech.edu

Abstract

Low signal-to-noise ratio (SNR) videos—such as those from underwater sonar, ultrasound, and microscopy—pose significant challenges for computer vision models, particularly when paired clean imagery for denoising is unavailable. We present Spatiotemporal Augmentations and denoising in Video for Downstream Tasks (SAVeD), a novel self-supervised method that denoises low-SNR sensor videos using only raw noisy data. By leveraging distinctions between foreground and background motion and exaggerating objects with stronger motion signal, SAVeD enhances foreground object visibility and reduces background and camera noise without requiring clean video. SAVeD has a set of architectural optimizations that lead to faster throughput, training, and inference than existing deep learning methods. We also introduce a new denoising metric, FBD, which indicates foreground-background divergence for detection datasets without requiring clean imagery. Our approach achieves state-of-the-art results for classification, detection, tracking, and counting tasks and it does so with fewer training resource requirements than existing deep-learning-based denoising methods. Project page here, Code: https://github.com/suzanne-stathatos/SAVeD.

1. Introduction

Motion may be the only way to identify objects in video with low signal-to-noise-ratio (SNR), camouflage, or complex textures that may hinder frame-by-frame object detection. The human visual system is excellent at capturing observable motion [15], a capability which has not yet been reproduced by modern computer vision models. Learning to exploit motion cues will improve models' abilities to detect and track objects of interest in noisy video.

In several scientific [18] and medical applications [10, 48], clean (noise-free) imagery are not available to train image or video denoisers. One attractive element of self-supervised models is their ability to find the signal within the noisy imagery itself. Self-supervision can, too, be more robust and generalizable to various noise type [36, 45].

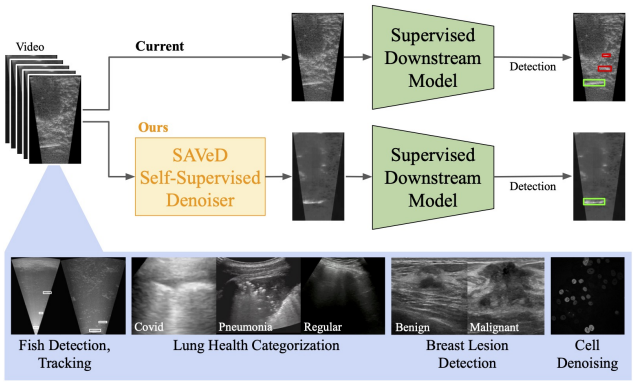


Figure 1. **SpatioTemporal Denoising improves classification, detection, tracking, and counting in video.** We denoise sonar and ultrasound videos of fish in a river, lung scans, breast lesion scans, and cell microscopy to improve downstream classification, detection, tracking, and counting tasks. We propose a self-supervised method to enhance the foreground signal of video frames without manual annotations or clean imagery. Our method works on grayscale videos with: non-stationary backgrounds, low signal-to-noise-ratios, and a variable number of objects in a video.

This work aims to enhance motion signals in low-signal-to-noise (SNR) data with non-stationary backgrounds, such as underwater, ultrasound, and sonar videos, to improve downstream supervised classification, detection, and tracking. We address sensor, background, and noise challenges with SAVeD, a self-supervised learning method to boost signal in noisy video. We exploit object motion to boost the SNR across frames. Inspired by work on self-supervised reconstruction [25, 47], self-supervised video understanding [39], and anomaly detection [33, 59], we use an encoder to encode appearance frames, a temporal bottleneck, and a decoder network to reconstruct the denoised frame.

Our main contributions are:

- We propose SAVeD, a novel, state-of-the-art, resource-

[=]Equal advising contribution

efficient self-supervised method to improve the signal in low-SNR video with variable numbers of agents and non-stationary background; details are in Sec. 3.
- We introduce a novel denoising metric, FBD, that does not rely on clean imagery; it instead relies on background-foreground appearance distribution differences, in Sec. 3.
- We propose a rich benchmark for low-SNR video denoising consisting of a diverse collection of low-SNR video domains (sonar video of fish, ultrasound video of lungs and breast tissue, and microscopy video) and a diverse collection of downstream visual tasks (classification, detection, tracking, and counting), in Sec. 4.

2. Related Work

Several works in the image and video denoising space require paired clean/noisy imagery for training, evaluation, or both, limiting their applicability in real-world denoising scenarios [23, 30, 35, 60]. We, notably, do not, and we focus this section on existing self-supervised and unsupervised approaches.

Self-supervised and unsupervised image (frame-by-frame) denoising. Several approaches use variants of blind-spot networks or pixel-wise masking to denoise imagery, including frame-by-frame video. Noise2Self [2] and Noise2Void (N2V) [26] train on noisy images without requiring clean targets or paired data. N2V trains a blind-spot network to predict masked pixels' intensities from neighboring pixels. Others [2, 22, 28] refrain from masking pixels via structural blind-spot networks with half-plane receptive-field U-Nets [41]. Jang et al. [22] use a conditional blind-spot network and a loss that regularizes denoised images without masking input pixels. Neighbor2Neighbor [19] proposes a self-supervised loss between two sub-sampled images. In general, noise in real-world imagery, including acoustic imagery, has unknown or non-stationary statistics that are spatially correlated, violating assumptions of pixel-wise independence.

Video Denoising with Flow Estimation Some video denoising methods leverage videos' spatiotemporal structure by using optical flow for motion compensation [49, 56]. DVDnet [49] uses calculated flow-estimates to manually warp frames, align their contents, and process them collectively with a CNN. VDFlow [57] jointly learns video deblurring and optical flow. These approaches target video where one component is moving (*e.g.* an object or the camera). However, when objects' motion affects the background's motion, or when objects are small, we find that leveraging optical flow enlarges objects' motion and makes it difficult to distinguish multiple closely-located objects. An example is in Fig. 16 in the Supp Mat.

Video Restoration and Denoising with Spatial and Temporal Consistency UDVD [44] uses a patch-wise noise-to-noise training strategy to predict clean frames by estimating masked pixels from adjacent neighborhoods of noisy frames. It relies on temporal redundancy across frames and does *not* rely on clean imagery, making it appealing for in-the-wild data. However, UDVD assumes gaussian noise, while underwaters sonar and ultrasound videos have pink and speckle noise [20, 37, 58]. UDVD's patchwise nature may cause the model to overwhelmingly learn background when most of the video is spatially and temporally dominated by background. UMVD [1] extends UDVD to focus on microscopy data. It uses frame-level augmentations and a reconstruction loss that predicts each frame from its noisy temporal neighbors. It assumes that the signal is consistent across frames while the noise is not, and it assumes smooth object motion. However, like UDVD, it is also patch-wise.

AverNet [61], another self-supervised video restoration model, similarly relies on temporal consistency to fix time-varying unknown degradations. It uses two modules: 1.) prompt-guided alignment to line up video frames at the pixel level and 2.) prompt-conditioned enhancement, which restores each frame by adapting to the image's degredation. The method was tested and performed well on videos with varying levels of gaussian, poisson, and speckle noise, gaussian and resizing blur, and jpeg and video compression.

LG-BPN [54], another patch-wise denoising method, uses two main ideas: local guidance and internal consistency. Local guidance preserves image structures like edges or textures that are consistent even with noise. Internal consistency assumes nearby patches in space and time share similar structure. However, LG-BPN struggles with long-range temporal dependency and highly-structured noise, such as sensor artifacts. RVRT [31] is a transformer-based model for denoising, deblurring, and super-resolution. It processes videos frame-by-frame, but models long-term dependencies (remembering what it learned from frames that are further apart) using spatiotemporal attention. It, however, requires very large memory even for inference.

Denoising autoencoders (DAEs) were originally introduced to learn robust representations. During training, DAEs intentionally add noise to input data and learn to reconstruct the original uncorrupted signal. mDAE [11], a method for missing data imputation (replacing missing or unavailable data), improves performance on several datasets. DAEs have also been applied to video tasks. CompDAE [38] explicitly models noise from snapshot compressive imaging measurements in low-light conditions to improve edge detection and depth estimation. TADA [9] uses an adversarial denoising autoencoder to remove EMG noise from EEG time series data. Our work similarly extends the application of DAEs to spatio-temporal grayscale video denoising; we uniquely combine temporal frames to enhance signal quality while simultaneously addressing the increased noise introduced by this process.

Sonar and Ultrasound Sonar and ultrasound present

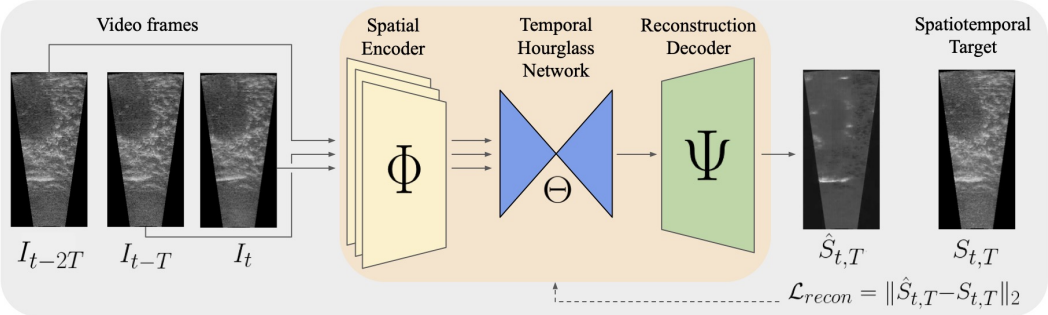


Figure 2. **SAVeD, our approach for self-supervised denoising using spatiotemporal difference and identity reconstruction**. I_t, I_{t-T}, and I_{t-2T} are video frames at times t (current frame), t-T, and t-2T. These frames are input to an appearance encoder Φ. The resulting feature representations are input to a spatiotemporal bottleneck Θ that compresses the 3 appearance features into a single spatiotemporal feature representation. Our model then predicts the reconstruction target, defined in Eq. (2) in Sec. 3.2, using the reconstruction decoder Ψ. The loss, defined in Eq. (3), is calculated and backpropagated through all networks. The architecture is discussed in more detail in Sec. 3.3.

challenges to the computer vision community. Key characteristics include pink noise, brighter pixel intensities for non-cavity objects compared to the background, and limited distinctiveness from appearance features. Weld et al. [55] address ultrasound sensor's variability via geometric analysis and augmentation. Unlike many computer vision datasets based on signal from light intensity, ultrasound, sonar, lidar, and radar rely on wave echoes. The "camera" emits waves that reflect off objects and return to the sensor; distance is measured by the echo's return time. Sonar and ultrasound use sound waves (mainly in liquids), while lidar uses laser pulses and is common in air and land environments [7, 8, 14]. We focus on one sonar dataset and two ultrasound datasets described in more detail in Sec. 4.1.

3. Method

The goal of SAVeD (Fig. 2) is to enhance objects' signal by isolating and emphasizing their motion in video with a non-stationary, fluid background. Inspired by prior work [21, 43, 47], we use an encoder-decoder framework. Our contribution is twofold: (1) a reconstruction target based on spatiotemporal differences across neighboring frames, and (2) a denoising metric that does not require clean ground truth. The method assumes that background spatiotemporal statistics differ from those of the foreground.

3.1. Self-supervised denoising

In our benchmarks' low-SNR videos, signals are distributed across time; as such, we want to condense information from multiple timesteps into a single frame to exaggerate the signal. We do this through the reconstruction target. For simplicity, we choose to reconstruct the spatiotemporal combination of 3 frames, expressed, in Eq. (2), from three input frames, I_t, I_{t-T}, and I_{t-2T}. We explore a vanilla autoencoder, UNets with and without skip connections and residual layers, and 3D convolutions (in Tab. 3), and find a simple encoder-bottleneck-decoder framework optimal. We also explore $N > 3$ input frames, $T > 1$, the target without the DAE network as input to downstream tasks, and more in our ablations in the Supplemental Tab. 4, 5, and Fig. 10.

We use an encoder-decoder architecture, seen in Fig. 2, with a spatial encoder, Φ, a temporal hourglass network, Θ, and a reconstruction decoder, Ψ. During training, spatial encoders, Φ, take I_t, I_{t-T}, and I_{t-2T} as input to generate spatial feature embeddings, which are then used by the hourglass network, Θ, to generate a spatiotemporal feature embedding; this embedding passes to Ψ to reconstruct the learning objective $\hat{S}_{t,T}$.

$$\hat{S}_{t,T} = \Psi(\Theta(\text{concat}(\Phi(I_t), \Phi(I_{t-T}), \Phi(I_{t-2T})))) \quad (1)$$

3.2. Reconstruction target and loss

Target. We use the *directionally-positive frame difference with the current frame (PFDwTN)* as our reconstruction target. This combines the current frame with the positive motion from the previous and next frames. Directionally-positive motion of the next frame is defined: $\max(0, I_{t+T} - I_t)$; of the previous frame, it is defined: $\max(0, I_{t-T} - I_t)$. In both cases, motion is relative to the current frame I_t. Note that the previous frame I_{t-T} goes into the network, while the future frame I_{t+T} does *not*. It is used only when calculating the ground-truth target for the loss.

To handle frames where the background movement does not differ significantly from the foreground objects' motion (*i.e.*, stationary objects), we include the original frame, I_t, in the reconstruction target. The overall target is:

$$S_{t,T} = \max(0, I_{t-T} - I_t) + I_t + \max(0, I_{t+T} - I_t) \quad (2)$$

Other motion-augmenting targets that we tested are defined/visualized in Sec. A.1 and Fig. 10 in the Supp. Mat.

Loss. We apply mean-squared-error loss for reconstructing

the current frame with augmented motion signatures:

$$\mathcal{L}_{recon} = \|\hat{S}_{t,T} - S_{t,T}\|_2 \quad (3)$$

3.3. Noise Removal Network

Appearance Encoder Φ. We implement a 6-layer CNN. Each layer consists of a convolutional block (Conv2D + ReLU) followed by max pooling, progressively increasing the number of feature channels while reducing the spatial dimension, $\mathbb{R}^{(H,W,1)} \rightarrow \mathbb{R}^{(\frac{H}{32},\frac{W}{32},512)}$. We also save skip connections (sequential max pools and 1x1 convolutions) to be used by the hourglass network and decoder. This design uses a fraction of the parameters and FLOPs from an off-the-shelf UNet, to let the network capture multi-scale features efficiently. See Tab. 1 for efficiency comparisons.

Temporal Hourglass Network Θ is an hourglass network with a bottleneck consisting of two 3x3 convolutional layers with 512 channels, each followed by ReLU activation. We also have skip connections as feature combiners at each level of the network, designed to merge information from the provided appearance features' skip connections.

Reconstruction Decoder ψ has 6 upsampling stages, each consisting of a ConvTranspose followed by convolutions and ReLU activations. At each layer, the skip connections from the corresponding encoder level are concatenated with the upsampled features. The decoder reduces the number of channels while increasing the spatial dimension ending with a single-channel output, $\mathbb{R}^{(\frac{H}{32},\frac{W}{32},512)} \rightarrow \mathbb{R}^{(H,W,1)}$.

More details on each of these modules is in the Supplemental materials Tab. 6.

3.4. Denoising Metric

Typically [19, 22, 26, 28, 44], denoising networks use Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [53] as evaluation metrics. PSNR is the ratio of maximum signal power to noise power, and SSIM measures perceived image quality. Both metrics rely on having clean imagery for comparison. Our denoising approach is unsupervised – we do *not* have clean imagery. As a result, we design a new metric, Foreground-to-Background Divergence (FBD), to evaluate denoised performance when we have detection annotations, and we rely on downstream performance as a proxy for denoised performance on datasets with different task annotations.

3.4.1. Foreground-to-Background Divergence (*FBD*) Metric for Unsupervised Denoising

Recall that we assume that our *downstream* models are supervised. Therefore, for detection tasks, we can assume we have bounding boxes or segmentation masks. For simplicity, we call detection annotations "boxes", though the same approach works for segmentation masks.

FBD measures how well a denoising method makes objects distinguishable from background by computing the KL-Divergence between a region containing an object (foreground) and the same region at a different time without it (background). An example is shown in Fig. 3. Unlike PSNR, which requires a clean reference, FBD does not, which is critical in many real-world applications.

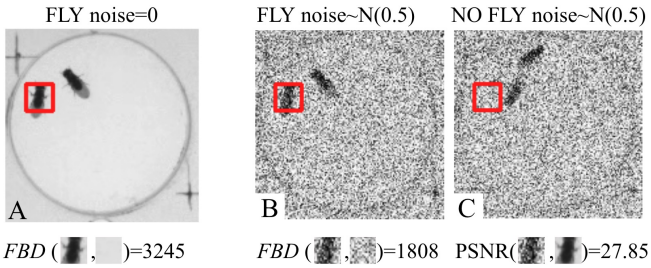


Figure 3. High-SNR visualization and calculations for FBD and PSNR on Fly-vs-Fly [13] with synthetic noise. In order to calculate the PSNR of **(B)**, a 'clean' image **(A)** is needed. **(A)** is not needed for FBD, though **(C)** is.

For each object's box b in each frame I of each video, we take the density of pixel intensity values: $d_b = I[b]$, where b are the indices associated with the box. Then, we take the density of pixel intensity values from a different frame $\tilde{I}$ of the *same* video at the same box location b where we know there is no object, $\tilde{d}_b$. If the distributions are separable *i.e.*, the distribution of object pixels is distinct from the distribution of background pixels, then the denoising method works as intended. We measure the separability between object and non-object via the Kullback-Leibler (KL) Divergence [27]. KL divergence measures the distance between two probability distributions P and Q as follows:

$$D_{KL}(P\|Q) = \int p(x) \log(\frac{p(x)}{q(x)}) dx \quad (4)$$

To generate a metric for a data split, we average the D_{KL} over all N bounding boxes to get

$$FBD_{D_{KL}} = \frac{1}{N} \sum_{b \in N} D_{KL}(d_b \| \tilde{d}_b) \quad (5)$$

A visualization of this metric can be seen in Supplementary materials Fig. 12. A larger score (divergence) indicates greater distinguishability of objects from background.

4. Experiments

We demonstrate that SAVeD can improve performance in low-SNR videos across medical and ecological applications (Sec. 5). We evaluate our processed images on downstream tasks for detection, tracking, counting, and classification.

4.1. Datasets

Caltech Fish Counting 2022 (CFC22) [24] is designed for detection, tracking, and counting fish in low-signal-to-noise

sonar video. This dataset contains 1,567 sonar videos from seven different cameras on three rivers in Alaska and Washington. The videos are grayscale, their resolutions range from 288x624 to 1,086x2,125, their frame rates range from 6.7 to 13.3 fps, and each video is on average 336 frames (38s) in duration [24]. In total, there are 527,215 frames with 8,254 unique fish, totaling 516k bounding boxes and 16.7 hours of video [24]. The dataset includes significant domain shifts (*e.g.*, background topology, occlusion, fish densities, fish sizes, camera noise), requiring models to generalize effectively across conditions.

Breast Lesion Ultrasound Video Dataset [32] (BUV) is designed for classifying (benign or malignant) and localizing breast lesions. The dataset contains 188 videos, of which 113 are malignant and 75 are benign. These videos collectively have 25,272 images, each with 1 detection; the number of ultrasound images in each video range from 28 to 413. Each video has a complete scan of the abnormal tissue. The dataset has a random train–test split of 150–38 videos respectively[32].

The Point-of-care Ultrasound dataset (POCUS) [5, 6] contains convex and linear probe lung ultrasound images and videos for classifying COVID-19 and pneumonia. It includes 247 videos and 59 images; we use only the videos. Of these, 70 show COVID cases, 45 *possible* COVID, 51 bacterial pneumonia, 6 viral pneumonia, and 75 healthy lungs. Videos are sampled at 10Hz, and frames are grouped by video as in Born et al. [5, 6]. In total, we extract 9,184 frames with an average size of 499×463 pixels.

Fluorescence microscopy dataset [52] (Fluo) is a dataset of fluorescence-microscopy recordings of live cells in [52]. We use the same videos as UDVD [44]: Fluo-32DL-MSC (CTC-MSC), of mesenchymal stem cells, and Fluo-N2DH-GOWT1 (CTC-N2DH), of GOWT1 cells. This dataset also contains no ground-truth clean data. There are a total of 560 frames and four videos.

4.2. Training Procedure – Denoiser

We train SAVeD using the reconstruction objective in Eq. (3). During training, we rescale CFC22 and POCUS images to 1024x512 and BUV and Fluo images to 1024x1024. For POCUS, we use a sample rate of 10Hz, as that is what the downstream process uses. For all other datasets, we use all frames. Because SAVeD is self-supervised, when training the *denoiser* for each dataset, we train over all data. We train for 20 epochs for CFC22, 120 epochs for POCUS, 40 epochs for BUV, and 1000 epochs for Fluo; we found these numbers of epochs sufficient for training to converge. These took 20 hours, 0.5 hours, 2 hours, and 2 hours, respectively, on 2 RTX 4090 GPUs; this is less time than other network-based denoising methods as seen in Tab. 1. Additional details, including hyperparameter configurations, are in the Supplemental Materials Sec. B.2.

After training SAVeD, we generate denoised frames for all splits. In the case of CFC22, we combine the denoised image as two channels and the background-subtracted frame, $(I_v)_t - \bar{I}_v$, as the last channel. For POCUS, BUV, and Fluo, we combine the denoised image as two channels and the median-filtered image as the last channel.

4.3. Evaluation procedure – Downstream Tasks

Given that none of our videos have clean (noise-free) versions, we use the downstream performance tasks' metrics as proxies for our denoised performance. We use FBD from Sec. 3.4 when detection annotations are available.

Denoising for Detection, Tracking, and Counting. For CFC22, which has detection, tracking, and counting as downstream tasks, we follow a simplified version of the detection pipeline from Kay et al. [24] – we train a YOLOv5 model for 5 epochs with the longest side of an image set to 896 and no augmentations. We remove duplicate predictions using non-maximal suppression. We use mAP_{50} [12] to evaluate detection performance frame-by-frame. We use a pretrained-frozen ByteTrack tracker and calculate MOTA [4], HOTA [34], and IDF1 [40] scores for evaluation. More details and hyperparameter settings are in the Supplemental Materials Sec. B.3 and B.4. For counting, we use trajectories from the tracks to create nMAE scores, defined in Kay et al. [24], for each domain. The tracking and counting pipelines are training-free.

For BUV, we follow the training procedure of Lin et al. [32]; we also follow their final fine-tuning step and evaluation to generate an AP_{50} metric. Note that we know that breast lesions are darker (rather than brighter) spots in ultrasounds. As a result, we invert our reconstruction error to take the minimum difference rather than the maximum:

$$\text{inv}S_{t,T} = \min(0, I_{t+T} - I_t) + I_t + \min(0, I_{t-T} - I_t) \quad (6)$$

Denoising for Classification. For POCUS, we perform 5-fold cross-validation following Born et al. [5, 6], ensuring all frames from a video remain in the same fold. We adopt their fine-tuning strategy and hyperparameters. For evaluation, we compute per-class precision, recall, and F1, then average across folds to obtain overall metrics.

5. Results

SAVeD boost signal of objects of interest in low signal-to-noise video. It improves a range of downstream tasks in a way that is computationally less resource-intensive and yields higher performance than other denoising methods.

5.1. Denoising Performance.

SAVeD produces clear contiguous objects, where other methods do not, shown in Fig. 4 & 5. SAVeD is also able train to convergence in 22% of the time of the next-quickest network-based method – more can be seen in Tab. 1.

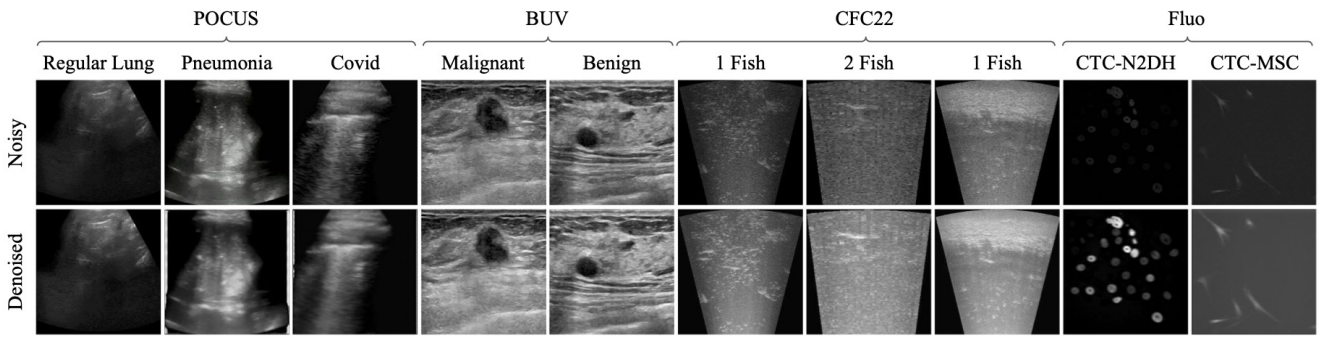


Figure 4. **Qualitative raw-denoised pairs of SAVeD**. Qualitative results for SAVeD trained on POCUS (lung health categorization), BUV (breast lesion detection), CFC22 (fish detection, tracking, and counting), and Fluo (cell denoising).

Train-Time (hours)	CFC22	POCUS	BUV
N2V [26]	144	0.75	1.5
UDVD [44]	96*	12	23
UMVD [1]	91*	34	42
LG-BPN [54]	101	6	4
SAVeD (Ours)	**20**	**0.5**	**1**

Table 1. **SAVeD is resource-efficient during training.** Note that UDVD and UMVD (U*VD) took 8 days to train CFC22, but U*VD trained CFC22 only for one epoch. For all other datasets, U*VD both trained for 10 epochs on 2 NVIDIA RTX 4090 GPUs.

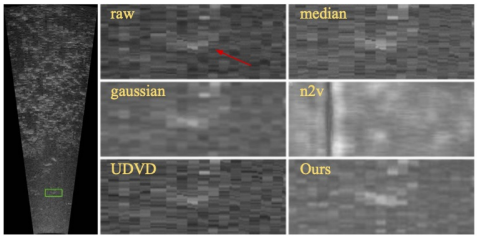


Figure 5. **Qualitative denoising performance on CFC22**. We can see that the fish is easiest to spot as a bright patch after processing with our denoiser. The green box highlights the fish location. Each denoised image zooms in to that green bounding box. The red arrow in the raw frame points to the fish location. Additional example visualizations are in Fig. 15 in Supp. Mat.

Fish Denoising: CFC22. SAVeD increases the contrast between fish and background (Fig. 5). As such, the distributions of pixel intensities at the same location when fish are present and when they are not are distinct. This is shown in Tab. 2, where SAVeD's FBD is significantly higher than that of other methods.

Fluorescent Cells Denoising: Fluo. SAVeD increases the cells' brightness relative to the background, as seen in Fig. 4. As is standard[44], and because the data size is small, we only perform qualitative analysis on Fluo.

	FBD (↑)		
	Train	Val	Test
Raw	1005	652	860
CFC22++[24]	*63.7*	368	458
Median-filtered[16]	523	334	364
Gaussian-filtered[16]	793	507	756
N2V[26]	227	*194*	*180*
UDVD[44]	402	254	272
SAVeD (Ours)	**1366**	**994**	**1458**

Table 2. ***FBD* Eq. (5) between P(Fish) vs. Q (Non-fish).** For all ground truth bounding boxes, P and Q are composed as follows: P – we take the set of pixels in each box from frames with objects. Q – we extract the set of pixels from the same box location from a frame where there is no object at that location. *Raw*=raw noisy frame I_t, CFC22++[24] = 3-channel image (raw, background-subtracted, frame-to-frame difference), SAVeD=denoised with motion augmentation, as in Sec. 3.2. We calculate the KL-divergence metric, discussed in Sec. 3.4.1. ↑ indicates the metric is better the larger it is. Best values are **bolded**, worst values are in *italics*.

5.2. Detection Performance

SAVeD outperforms other denoising methods when evaluated on downstream detection tasks of CFC22 and BUV.

Fish Detection: CFC22. The detection performance of SAVeD-processed frames is better than detection performance of other processed frames for CFC22. This is shown in Fig. 5 & 6 and Tab. 3. SAVeD improves detection performance in areas where objects and signal are rare. SAVeD frames result in an improvement of 43.2% and 9.4% test accuracy compared to the raw and background-subtracted frames respectively, and a 5.1% boost in performance compared to a three-channel image (raw, background-subtracted, and frame-to-frame-absolute difference) described as baseline++ in Kay et al. [24], but hereon referred to as CFC22++.

Breast Lesion Detection: BUV. SAVeD clarifies the breast lesion imagery, as seen in Fig. 4 and results in a boost of

	CFC22 (Test)			POCUS (5-fold-CV)			BUV(Test)
Method	mAP_{50}[12]↑	MOTA[4]↑	nMAE[24]↓	AP↑	AR↑	F1↑	mAP_{50}[12]↑
Classical							
Baseline	73.8	37.4	54.8	82.6	82.0	80.4	46.4
Median-Filter[16]	73.7	37.8	53.0	86.2	85.5	85.3	52.4
Mean-Filter[16]	76.4	44.3	41.4	84.0	84.7	83.2	52.6
Gaussian-Filter[16]	74.9	27.6	56.8	84.1	84.3	83.3	46.5
Deep-Learning-Based Image/Video Denoising/Restoration							
N2V[26]	67.2	34.2	34.3	83.7	82.7	82.4	46.6
UDVD[44]	67.2	28.1	41.9	83.7	84.6	83.4	49.9
UMVD[1]	70.4	38.2	34.0	80.4	82.6	80.7	53.9
LG-BPN[54]	72.6	41.9	34.2	31.9	34.3	21.5	49.7
RVRT[31]	62.9	29.3	43.8	81.8	79.8	78.7	26.0
AverNet[61]	65.4	30.3	35.7	83.7	83.4	82.6	47.0
DAEs							
AE	67.8	34.3	41.7	82.3	84.7	82.1	46.9
UNet [42]	73.9	34.1	56.3	83.7	84.6	83.4	51.0
UNet3D[42]	66.9	32.4	35.4	83.5	80.1	80.6	47.6
SAVeD (Ours)	**77.6**	**47.4**	**33.9**	**87.5**	**86.7**	**86.3**	**59.5**

Table 3. **Downstream results.** SAVeD does well across all datasets and downstream tasks. Best performance is **bolded**. Baseline refers to raw for medical ultrasound (POCUS [5, 6] and BUV [32]) and the strengthened baseline CFC22++[24] for fish sonar (CFC22 [24]. AP=average precision, AR=average recall, F1=average F1, mAP_{50}=mean average precision of detections at IOU threshold 0.5, MOTA=Multi-Object Tracking Accuracy[4], nMAE=normalized mean absolute counting error[24]. More tracking results are in Fig. 7.

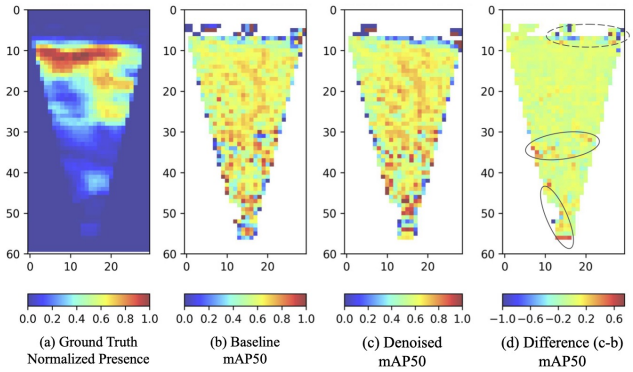


Figure 6. **Denoising improves detections where signal is infrequent**. **(a)** the ground truth fish patch locations (from bounding box labels) normalized over the dataset; most fish pass by in the top region, fish crossings below are infrequent, thus there is more training signal in the top part of the videos. **(b&c)** patchwise detection performance of CFC22++[24] and SAVeD, respectively, on the CFC22 dataset. Heatmaps indicate mAP_{50} performance over all frames of the test set at pixel patches. The more red a patch is, the higher the mAP_{50} of that patch; the more blue the patch is, the lower the mAP_{50}. **(d)** the difference, SAVeD - CFC22++, with solid ellipses at regions of heightened performance and dashed ellipses around areas of lowered performance. Denoising improves detections in areas where signal is infrequent. On the other hand, detection performance declines in areas where signal is abundant. Additional patch maps can be seen in Fig. 9 in the Supp Mat.

over 11% in breast lesion detection mAP_{50} scores compared to the next best method (UMVD). More is in Tab. 3.

Figure 7. **Quantitative tracking improvements.** CFC22++ consists of the three-channel (background-subtracted, absolute-difference, raw) frames. The "Perfect Denoiser" refers to frames that have black backgrounds and white masks at the bounding box locations. Denoising results in higher MOTA scores for val and test; SAVeD boosts IDF1 and HOTA scores in test moreso than in val.

5.3. Tracking and Counting Performance

Fish Tracking and Counting: CFC22. Compared to classical and other DNN methods, SAVeD frames yield higher tracking and counting performance (Fig. 7, Tab. 3). Stronger fish–background separation reduces false negatives and increases true positives in detections, subsequently improving tracker and count accuracy.

5.4. Categorization Performance

Lung Health Categorization: POCUS. Images processed through SAVeD yield the best 5-fold cross-validation image classification score compared to classical and network-based denoising methods on lung categorization, shown in

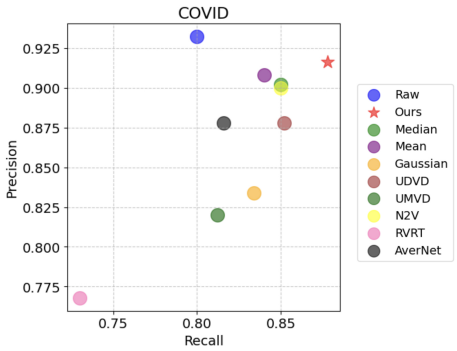


Figure 8. **Covid Precision-Recall across denoising methods**. SAVeD has the highest average precision and average recall across denoising methods. Additional class-wise performance comparisons are in Fig. 11 in the Supplementary materials.

Tab. 3. Fig. 8 shows the precision-recall for the denoising methods on the Covid class – SAVeD has the highest accuracy for Covid classification. Additional per-class performance analysis is in Sec. A.2 of the Supp. mat.

5.5. Ablations

We ablate the reconstruction target and the denoising autoencoder to find their relative importance.

Reconstruction Target. PFDwT1 is the most effective reconstruction target for increasing the accuracy of downstream tasks. We compared PFDwT1 to PFDwT2, σ (the standard deviation over input frames), $\Sigma - 5\bar{I}$ (the sum of 5 consecutive frames - 5*mean frame), and $\Sigma - 3\bar{I}$ (the sum of 3 consecutive frames - 3*mean frame). The results are shown in Tab. 4 and Tab. 5d in the Supplementary material.

Autoencoder vs. no Autoencoder. Using a DAE improves downstream detection performance over using the reconstruction targets alone for all targets. More detail is in Tab. 4 and Tab. 5d in the Supplementary material.

Architectures Our small architecture has better performance on downstream tasks compared to larger architectures. Comparisons of SAVeD with a vanilla Autoencoder, a UNet[42], and a UNet[42] with 3D convolutional kernels are shown in Tab. 3. For additional details and architectures, see Tab. 5 in the Supp. mat. The vanilla Autoencoder's architecture is also explicitly defined in Tab. 7.

6. Discussion and Conclusion

While SAVeD is a clear improvement, we recognize that there are further improvements to be made and research directions to be explored.

Limitations. As the spatiotemporal component of our method relies an object's location to be overlapping in sequential frames, very fast moving objects may decrease performance on downstream tasks using SAVeD. On the other hand, if objects are stationary, SAVeD does not improve performance, though it also should not be detrimental.

We recognize that combining noisy frames adds to the noise of the overall signal rather than removing it. Work [3] has shown that denoising methods capture clean data's underlying structure. Denoising autoencoders purposefully corrupt training data by adding noise or masking some of the input values [17, 51]. We rely on the autoencoder to remove noise implicitly by focusing on the largest reconstruction areas to minimize loss. This assumes that the objects of interest are larger than the noise signature. Indeed, we found that detection performance occasionally dipped for very small (*e.g.* $\leq$ 10 pixel) objects, when this assumption is not held.

We observe a change in frame noise when there is an object of interest vs. when there is no object of interest: namely, when there is no object of interest, the whole frame is bright, vs. when an object is in the frame, the object pops and the background becomes dim: ideally, the background would stay dim regardless. An approach to handle this would be adding a component to the loss to focus on background consistency.

Despite these challenges, SAVeD-processed frames outperform other denoising methods on downstream tasks while requiring fewer training resources.

Future work. We are interested in training end-to-end: combining the representations from the denoiser and the downstream tasks. We are also interested in exploring how to broaden and emphasize the motion signature. Finally, we recognize the shared qualities of each of these datasets and also understand that self-supervised methods are data-hungry [29, 46]. As such, one could explore the performance benefit of training on all datasets collectively to learn general low-SNR video properties.

Conclusion. We present SAVeD, a self-supervised denoising method that improves downstream performance in low-SNR videos without requiring clean data. Our approach is general and applicable to a range of low-SNR video tasks and domains. It is based on the confirmed intuition that while there is motion in the foreground and background, motion signatures between foreground and background are distinct, and a simple model can separate them to improve the SNR. We also introduce a new metric, FBD, to capture this relationship. SAVeD enhances object motion while leveraging autoencoders' denoising capabilities to boost downstream performance efficiently.

Acknowledgments. This work was supported by the Caltech Resnick Sustainability Institute Impact Grant "Continuous, accurate and cost-effective counting of migrating salmon for conservation and fishery management in the Pacific Northwest." We also thank Neehar Kondapaneni, Angela Gao, Sara Beery, Justin Kay, and Laure Delisle for helpful feedback.

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

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

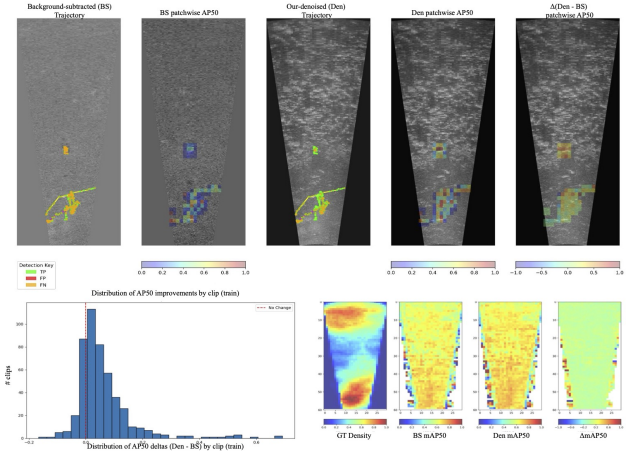
We fine-tune a YoloV5-small model pretrained on COCO using the default training settings from Ultralytics over 5 epochs with a batch size of 16. As in Kay et al. [24], we resize all inputs to have 896 pixels as their longest side; the learning rate is 0.0025. We select the best model checkpoint based 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

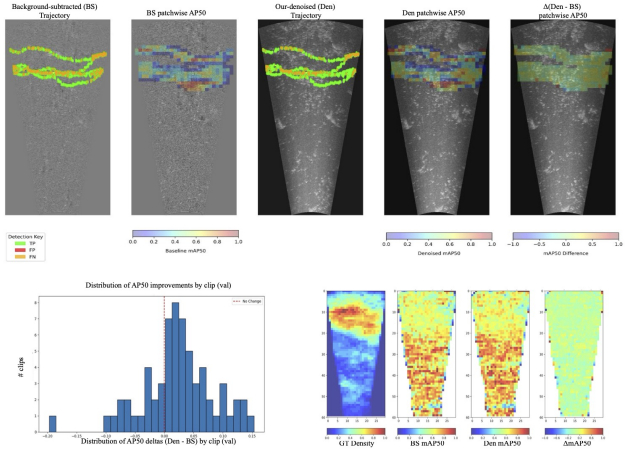
We use a pretrained ByteTrack tracker with hyperparameters selected as the optimal hyperparameters for tracking performance on the validation set. Max age, the time until a missing or occluded object is assigned a new id, is 20; Min hits, the minimum number of frames with a track for the track to be 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

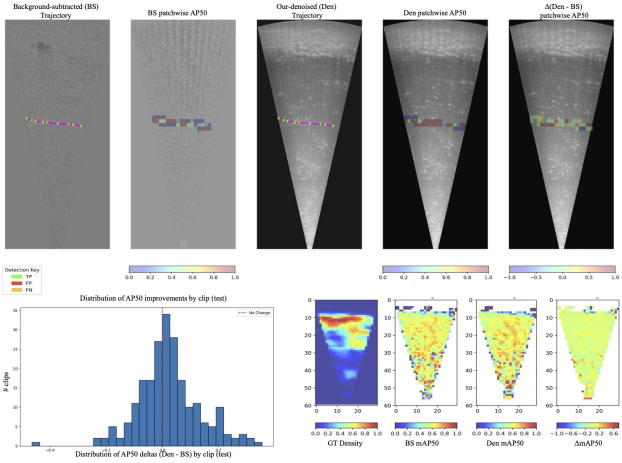
Additional visualizations of the denoising performance on fish in sonar (CFC22[24]) can be seen in Fig. 15).



(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.

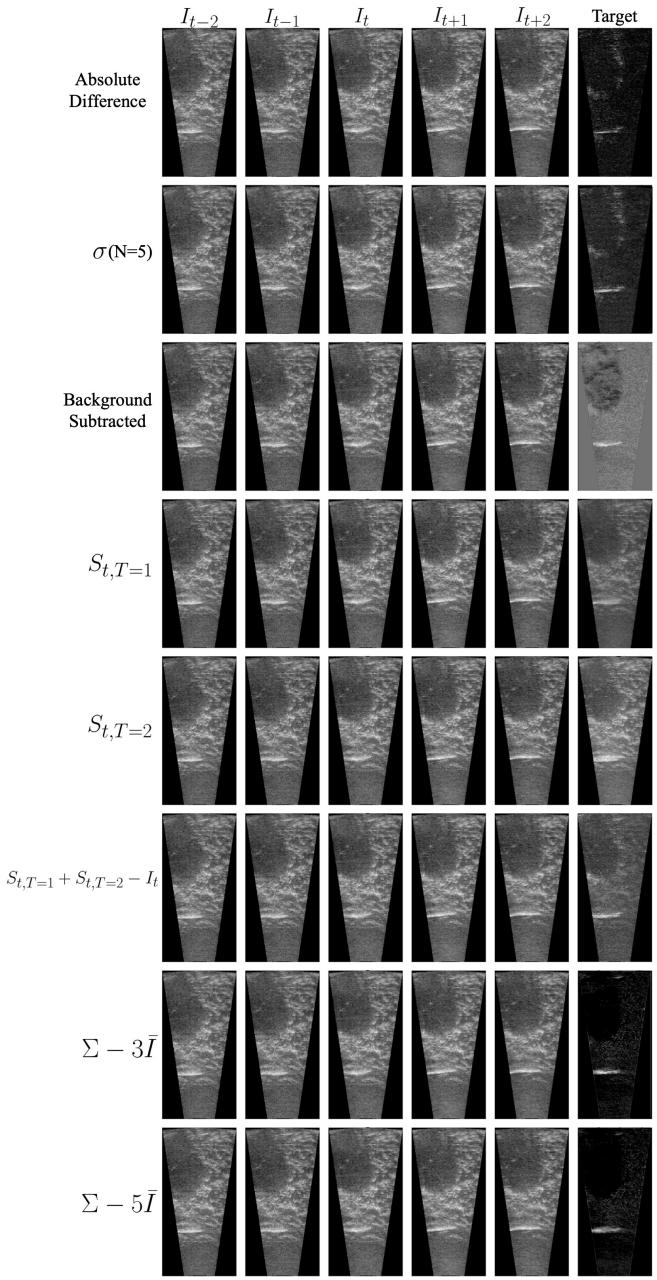


Figure 10. **Reconstruction Targets**. The window T=5 set of frames is shown with each reconstruction target we experimented with on CFC22. While $\Sigma - N\bar{I}$ frames appear strong in this example, we found that empirically they struggled to capture fish that did not move significantly between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

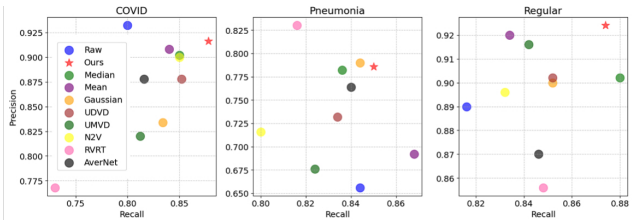


Figure 11. SAVeD (starred) has high precision and high recall across all POCUS classes.

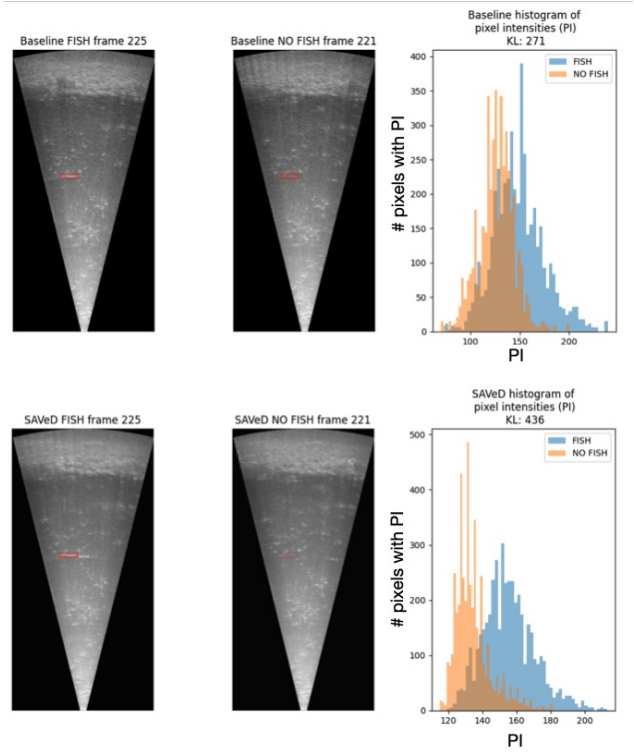


Figure 12. **Visualization of** FBD. Both images on the left are noisy images. The image on the far left has a fish located in the red bounding box. The image in the middle is a frame from the same video clip but with no fish in the red box. The histogram compares the pixel intensity values of the pixels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

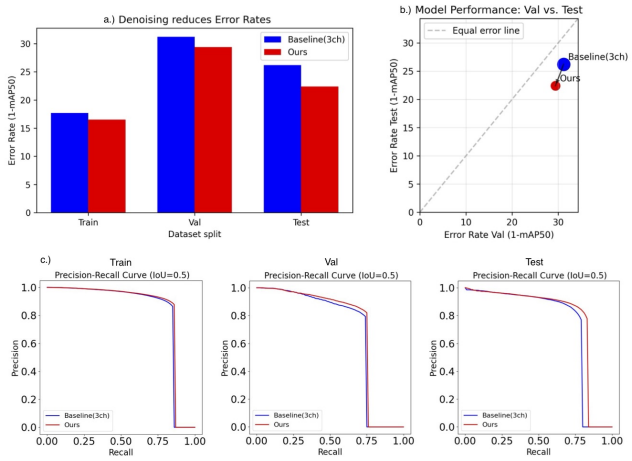


Figure 13. **Denoising lowers detections error-rates by improving precision and recall** (a) shows baseline detection error (1-mAP_{50}) compared to our detection error after our denoising preprocessing step. For all splits train, val, and test, denoising results in lower error. (b) compares error rates 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.

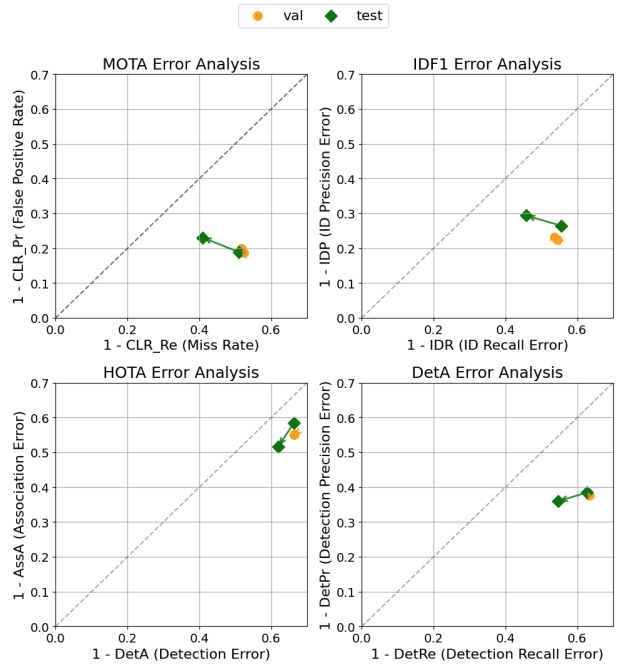


Figure 14. **Breakdown of track performance improvements for CFC22 val and test**. We can see test improves far more than val, as is standard for the CFC22 dataset.

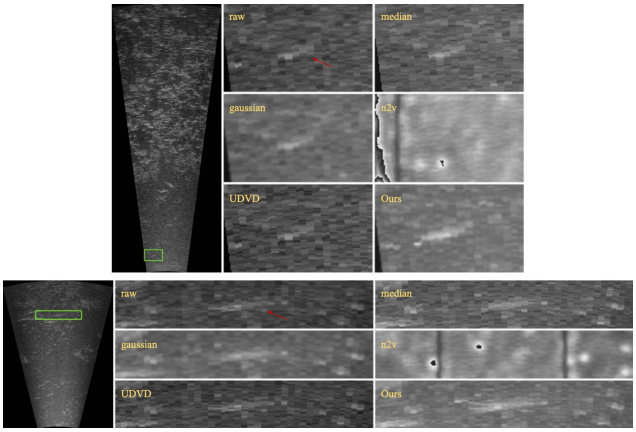


Figure 15. **Additional visualizations of denoising methods on CFC22**

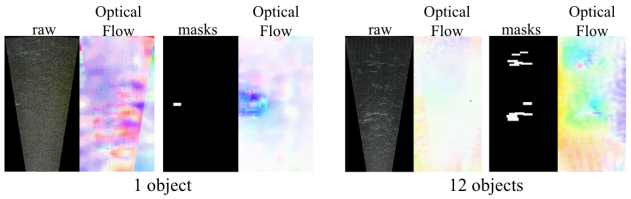


Figure 16. **RAFT [50] on CFC22 imagery and bounding box masks.** On the left, we can see the optical flow signal does not find the fish. When looking at the motion from the bounding-box mask of the fish (making the background movement stationary), the optical flow signal area is far greater than the 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.