

Fast Region of Interest Proposals on Maritime UAVs

Benjamin Kiefer¹ and Andreas Zell¹

Abstract—Unmanned aerial vehicles assist in maritime search and rescue missions by flying over large search areas to autonomously search for objects or people. Reliably detecting objects of interest requires fast models to employ on embedded hardware. Moreover, with increasing distance to the ground station only part of the video data can be transmitted. In this work, we consider the problem of finding meaningful region of interest proposals in a video stream on an embedded GPU. Current object or anomaly detectors are not suitable due to their slow speed, especially on limited hardware and for large image resolutions. Lastly, objects of interest, such as pieces of wreckage, are often not known a priori. Therefore, we propose an end-to-end future frame prediction model running in real-time on embedded GPUs to generate region proposals. We analyze its performance on large-scale maritime data sets and demonstrate its benefits over traditional and modern methods.

I. INTRODUCTION

Autonomous vision aboard UAVs has grown to an important research area [1]–[4]. Next to traffic surveillance [5], [6] and agriculture [7], also the field of search and rescue (SaR) has been tackled [4], [8]. However, while several works focus on path-planning and mission implementation [9]–[11], few works address the actual vision part, necessary for autonomously searching certain areas.

Finding interesting regions on the sea is a hard problem, since objects of interest are often not known a priori or have a vast variety of different appearances which is why supervised methods often fail in these scenarios. Even if object categories are known beforehand, current methods focus on object detection, which is not viable for large image resolutions and real-time (for rigor defined here to be $>25\text{FPS}$) performance on embedded hardware [12]. Both constraints occur in reliable SaR missions.

Furthermore, labeled data sets in these environments are scarce as the data acquisition process is a complicated undertaking, requiring strict safety regulations for all subjects, and is expensive [13]. Instead, it is considerably easier and cheaper to obtain raw data of sea surfaces.

What is more, often a low bandwidth, possibly due to large distances or suboptimal weather conditions, does not allow for the whole footage being transmitted to a ground station. This becomes especially severe in maritime scenarios, where the drone is far away from any ground station [14], [15]. While compression can be done on-board, it is often not sufficient and furthermore results in image quality loss across the whole image, i.e. also possibly quality loss in regions

of the image that need to be analyzed more thoroughly to exclude false positives or negatives.

Recently, special purpose video codecs that allow few regions of an image to be coded with near constant picture quality have been proposed to tackle this problem [16]. The high quality regions that are transmitted can subsequently be combined with an actual classical object detection system on a ground station with much more hardware resources.

This methodology separates the problem into two stages: generating few high-recall regions of interest of a high dimensional image in real-time in a low resource environment and classifying these regions into known classes on a ground station with more resources. Motivated by these observations, in this work, we formulate and formalize the former problem and propose an autoencoder-based future frame prediction model that generates meaningful regions of interest on sea surfaces which can run in real-time on an embedded GPU. Owing to the nature of maritime environments, we show that classical methods perform poorly due to dynamic backgrounds, wave movements, sun reflections and others while modern methods are too slow. As this method is a type of anomaly detection method, it does not require bounding box annotations. We introduce a metric that measures the recall at a given amount of footage being transmitted and show that this method outperforms classical methods on multiple benchmarks and metrics.

To train the proposed model, we capture over 60 minutes of 4K video footage with several cameras depicting the sea surface from different angles and altitudes at different days and waters. We provide a webserver where we host the Maritime Anomaly Detection Benchmark, where researchers can upload their predictions, which will be evaluated on the server side to allow for a fair comparison.

Our contributions are as follows:

- We formulate a novel problem of obtaining high-recall regions of interest in a high-resolution and real-time scenario and propose a future frame prediction autoencoder to detect these regions in real-time on an embedded GPU.
- We capture over 60 minutes of video footage of the sea surface in various conditions as train set for our method and make it publicly available¹. We host a web server and propose the Maritime Anomaly Detection Benchmark with upload options.
- We analyze the proposed method and compare it to traditional and modern methods on two large-scale public data sets.

^{*}This work was supported by the German Ministry for Economic Affairs and Energy, Project Avalon, FKZ: 03SX481B.

¹Both authors are with the Faculty of Computer Science, University of Tuebingen, Germany. `firstname.surname@uni-tuebingen.de`

¹<https://seadronesee.cs.uni-tuebingen.de/>

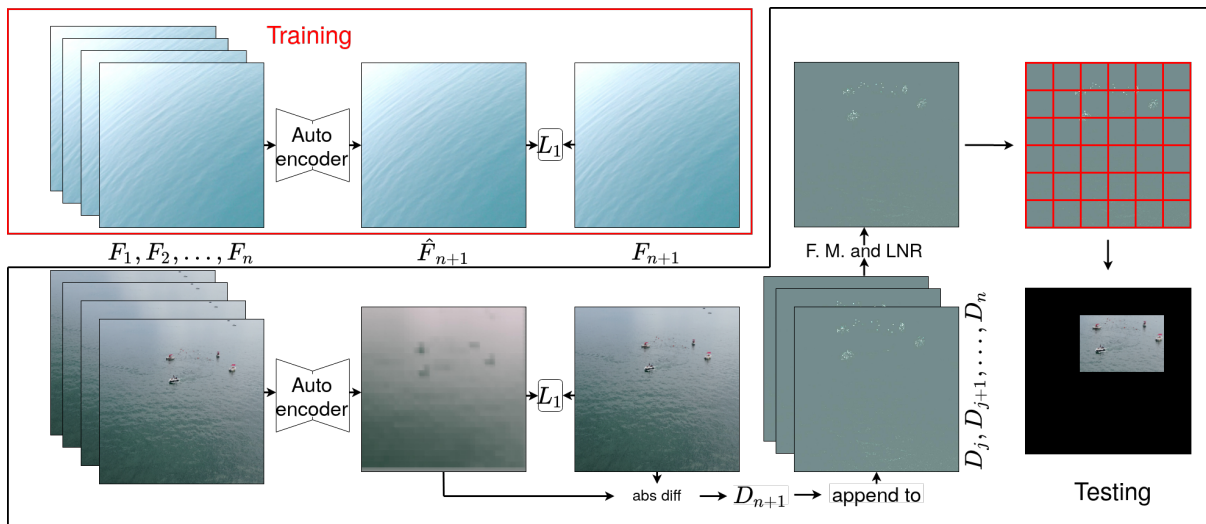


Fig. 1. Best viewed digitally. Future frame prediction autoencoder pipeline. The frames F_1, \dots, F_n are concatenated and input into the autoencoder, which learns to predict F_{n+1} via \hat{F}_{n+1} and the L_1 loss. The error frame (absolute difference between the two), D_{n+1} , is concatenated to the last D_j, \dots, D_n . Then, frame momentum and local noise reducer are applied until a final grid is put on the resulting error frame to yield final regions of interests.

A. Related Work

1) *Computer Vision in Maritime Environments*: Airborne maritime data sets are scarce and mostly focus on synthetic aperture radar satellite imagery and ships [17]–[20]. [3], [8], [21] provide UAV-based maritime detection data sets. While the data set in [3] features only stock photos scraped from the internet, the Seagull data set [21] and SeaDronesSee [8] provide video material with objects of interest. Of these data sets, only Seagull provides frames that do not contain objects. However, the videos suffer from heavy lens distortion and distortion caused by a rolling shutter. We collect 60 minutes of video footage (>100000 frames) depicting the sea surface in various altitudes at different angles and days with multiple cameras. We weakly annotate the footage such that no objects of interest are visible in any of the frames.

2) *UAV-based Detection*: Rudimentary vision methods in SaR scenarios aboard a UAV are done in [22], using color, text or shape cues using OpenCV [23] to detect objects of interest. Similarly, [24] use Haar features to detect objects of interest in SaR missions. Among the learning-based methods [25] consider the application of ship detection, classifying images into positives (containing ships) and negatives. While it is also an unsupervised method, they ignore the localization. [3] describe a complete SaR system from path planning over detection to action. However, their detection system is a basic YOLO variant unsuitable for large resolutions and real-time. Furthermore, it is restricted to the objects it is trained on. Generally, all literature regarding supervised UAV object detection can be considered related [1], [5], [6], [8], [26]–[28], albeit not viable, since they do not work in real-time large-resolution scenarios on embedded hardware.

3) *Region proposal networks*: Selective search [29], [30] generates many thousand little informative boxes for use of an object detector at a later stage, which makes it inappli-

cable in embedded environments. Common region proposal networks [31] are used in two-stage object detectors, such as Faster R-CNN [32], but require bounding box supervision to be learned. Methods for weakly supervised object detection often employ region proposal networks [33], but require image-level annotations.

Background subtraction methods [34] are used to separate the background from the foreground, which is defined by the scene captured by a static camera. Most of the methods are not suitable for dynamically changing scenes caused by camera and background movement. Furthermore, these methods do not focus on obtaining meaningful bounding box locations but only on the obtained segmentation maps.

(Video) Anomaly detection methods [35]–[37] learn on normal samples and detect anything previously not seen as anomalies. In images, this is often done for industrial parts [38], [39], and in static videos for surveillance in traffic and crowded scenes [40]–[43]. Earlier methods only focus on classifying images or frames [25], [43], while newer methods also consider localizing anomalies [44], [45]. However, these methods either ignore the temporal dimension or are not suitable for real-time use. Furthermore, video anomaly methods are designed for static scenes. Our work focuses on dynamic scenes and requires models running in real-time on embedded hardware. Furthermore, the focus is on generating high recall meaningful bounding box regions that potentially contain objects of interest.

II. METHOD

We are given a high resolution (e.g. 4K) video stream depicting the sea surface. Furthermore, we have an embedded GPU (e.g. Nvidia Xavier AGX). The task is to select regions of interest in every frame which are to be transmitted down (e.g. via a streaming FPGA [16]). Each region is defined via four bounding box locations, describing the corners of

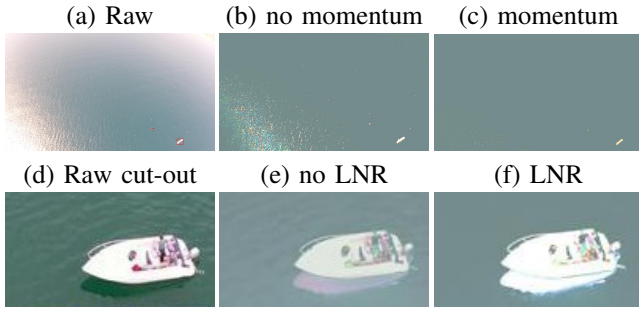


Fig. 2. Qualitative errors on Seagull (top) and SeaDronesSee (bottom). Note that the random noise at the bottom left of (c) almost vanished.

the region in pixels (similar to classical object detection). Depending on the exact use case, the remaining regions are either also transmitted with lower quality or completely omitted.

We propose an autoencoder-based future frame prediction architecture to detect anomalies (See Fig. 1). We train a shallow autoencoder on sequences of normal images F_1, \dots, F_n depicting the sea surface such that the model learns to predict the next normal frame. Subsequently, the predicted frame \hat{F}_{n+1} is subtracted from the original next frame F_{n+1} . The hypothesis is that the autoencoder fails to reconstruct objects that differ from the sea surface in their colors, shapes and textures (e.g. see Fig. 4 & 5). Furthermore, by incorporating the last few frames, the autoencoder learns temporal correlations of water movements.

Common future frame prediction networks employ large networks, such as UNet [43], [46] or even larger models [47]. They operate on small video resolutions and are not suitable for employment on embedded hardware. Applying models in real-time on embedded hardware and for high resolutions requires us to fall back to shallow autoencoder architectures. We follow the basic principle of an encoder-decoder architecture, but only employ small channel dimensions for the filters as these make up for a large computational overhead. We refrain from using depth-wise separable convolutions [48] or more advanced methods [49], since they are not optimized for embedded GPUs. For the first layer, we concatenate the past n frames along the channel dimension and apply a regular 2D convolution with filter dimension $n \times 4$, kernel size 3×3 and stride 2. We perform the same convolution six times, while halving the channel dimension each time due to performance. The decoder performs the symmetric operations via deconvolutions.

We hypothesize that reconstruction errors due to wave patterns and sun reflections are more temporally unstable than actual anomalies. Thus, we propose to include an error frame momentum term, which averages over the past n error frames D_1, \dots, D_n . This assumes that the camera movement is not too quick as then, the actual anomalies also move quickly in the image plane, eliminating the error frame momentum effect. However, for frame rates of roughly 30, this is negligible. Figure 2 (c) and Section IV show the advantage of using this component.

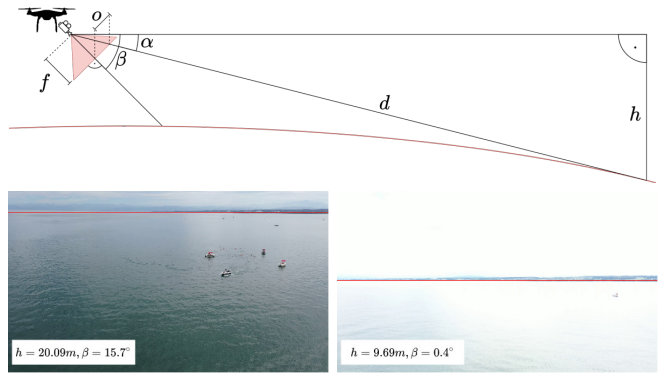


Fig. 3. Illustration of the horizon cutter (top) and predictions (bottom). The curvature is just for visualization purpose (ignored for computation).

To counteract the local noise induced by an imperfect reconstruction coming from sun reflections and wave patterns, we introduce a local noise remover (LNR). Channel-wise, we multiply each pixel of the error frame by its immediate vertical and horizontal surrounding neighbour. We repeat this procedure three times. This ensures that only regions of larger error areas are detected as anomalies (as opposed to noisy areas) in subsequent steps. This can be seen as a morphological operation, however also different, since we do not use a structuring element [50]. See how the waves are eliminated in Fig. 2 (f), while the boat is amplified.

Importantly, this method is sensitive to regions above the horizon. Therefore, we leverage meta data from the UAV's on-board sensors to determine the horizon line in open water. This allows us to ignore this region in the autoencoder training and inference phase which, in turn, results in more robust anomaly detection performance and faster inference times. Notably, this computation has virtually no overhead. The horizon line can be computed using the UAV's height, camera gimbal pitch and roll angle, and the camera intrinsics. Ignoring the effect of atmospheric refraction, we can estimate the distance to the horizon as a function of the height of the observer as $d = 3.57h^{1/2}$. This approximation is fairly accurate for heights that are typical for SaR-UAVs (far below 1000m) [51]. We furthermore ignore the curvature of the earth, which is also negligible for these heights. We compute the angle α to the horizon via $\alpha = \arcsin(h/d)$. Using the focal length f (in pixels) and the camera gimbal pitch β , we can then compute the camera perspective projection on the image plane, which yields the height offset o in pixels to the horizontal center line of the image plane as $o = \tan(|\alpha - \beta|) \cdot f \cdot \text{sgn}(\alpha - \beta)$. Naturally, we truncate o to be within the range of the number of horizontal pixels. To account for the roll angle γ of the UAV (or camera gimbal), we can simply add a roll angle induced offset at the left and subtract at the right of the image given as $o_r = \tan(\gamma) \cdot p_w / 2$, where p_w is the pixel width of the video. While the horizontal pixel location o is an approximation, it is quite robust to errors in the altitude h . Since an exact error analysis is not in the scope of this work, we just report values for altitudes that are common in the data set SeaDronesSee. For

TABLE I
ERROR TO THE GROUND TRUTH HORIZON AS MEASURED BY ROLL
ANGLE γ AND PIXEL OFFSET o IN A 4K IMAGE.

Errors of	γ	o	$o/2160$
horizon visible (5%)	0.8°	71px	3.3%
horizon not visible (95%)	–	2px	0.1%
total	–	5.45px	0.3%

$\beta = 0 - 20^\circ, h < 300m$ it holds that 10m in altitude error results in approx. 1px offset change in a 4K image. However, o is very sensitive to errors in the gimbal angle β . For example, for $h = 130m, \beta = 16^\circ, 1^\circ$ in angle error results in approx. 40px offset change. Therefore, it is essential to have a well-calibrated gimbal and UAV IMU. The latter can be accurate up to 0.1° when configured properly [52]. See Figure 3 for an illustration. Empirically, we show that the horizon cutter performs well despite the occurrence of nearby land. We manually annotate the horizon line for a subset of the SeaDronesSee-Tracking validation set and compute the pixel offset error and the roll angle γ error. Table I shows that despite some land mass blocking the horizon (also see Fig. 3), the error is negligible.

Lastly, we apply a grid of size $m_1 \times m_2$ on the error frame and for every grid window we average over the error frame pixels contained in it. We select the maximum number of boxes outputted given a certain bandwidth, which we simply break down in $p\%$ of the area of the whole frame.

III. DATA SET GENERATION AND WEBSERVER

To test our approach, we gathered 60 minutes of 4K video footage on open water at three different days with three different cameras. We made sure to include altitudes and viewing angles from $5 - 120m$ and $0 - 90^\circ$. We manually filtered out the sequences that contained objects considered anomalous, such as humans, boats, life jackets and buoys. Each frame is annotated with its corresponding meta data information, such as altitude, all angles of the UAV principal axes, camera gimbal pitch angle, time, GPS and others.

This data, called **OpenWater**, comes along with over 20 minutes of bounding box annotated footage in open water where we annotated the same classes as in SeaDronesSee, serving as anomalies. Everything but the test annotations will be uploaded to avoid researchers from overfitting. We will upload the test videos, on a web server and propose the Weakly Supervised Maritime Anomaly Detection Benchmark where researchers can upload their predictions, which will be evaluated and published on the server side for fair comparisons.

IV. EXPERIMENTS

For our experiments, we choose a grid size of 48×27 , predict the fifth frame from the past four, and use an error frame momentum of two. Influences of the components are discussed in further sections.

As we operate on high resolution videos and in real-time scenarios, we compare to three methods commonly

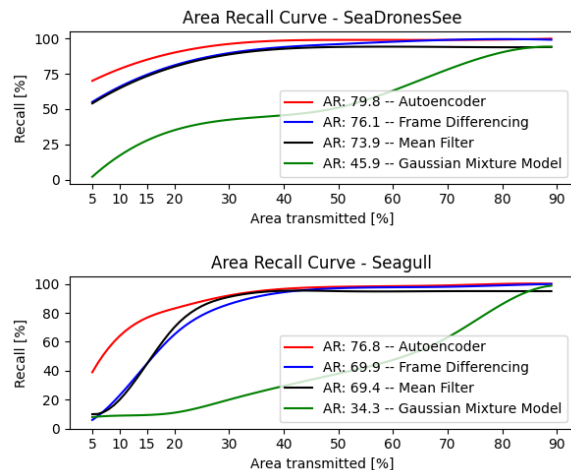


Fig. 4. Area recall curves for SeaDronesSee and Seagull. $AR \hat{=} avg.$ recall.

used for background subtraction and anomaly detection: Mean filter (MF) [53], frame differencing (FD) [54] and Gaussian mixture model (GMM) [55]. For GMM we use three Gaussians. We extend every method with the grid component and employ the horizon cutter.

We evaluate on the following datasets (see also Fig. 5):

- We use our **Open Water** data set as the training set for SeaDronesSee as the latter does not consist of frames without objects. It consists of > 100000 frames of open water captured on multiple days with three different 4K video cameras. Similar to SeaDronesSee, we provide precise meta annotations and manually annotate bounding boxes with the same classes as SeaDronesSee for each of the frames.
- We experiment on the multi-object tracking track of **SeaDronesSee**. It depicts humans, boats and other objects in open water (incl. bboxes) serving as our anomalies. The frames are of 4K resolution and each frame is annotated with precise meta data information. It is challenging since it can contain multiple objects scattered around the frames and due to its diversity in the altitude and angle of view distribution, resulting in different objects' appearances and sizes.
- The **Seagull** data set features video data showing boats, ships, life rafts and other objects from a fixed wing UAV. It also features video clips containing no objects. The latter serve as our training set for the Seagull test set. The videos are of Full HD resolution and have a heavy lens distortion and distortion caused by a rolling shutter. See also Figure 5 for examples.

We measure the recall given a certain bandwidth (percentage of the video frame transmitted averaged over all frames). Therefore, we consider as evaluation metric $R^{p\%}$, which is the recall over all frames, given that at most $p\%$ of the image may be transmitted. Each region to be transmitted must be encoded by a rectangular bounding box. We consider an object to be correctly detected if there is an overlap with the predicted region of at least 50%, which is common for aerial

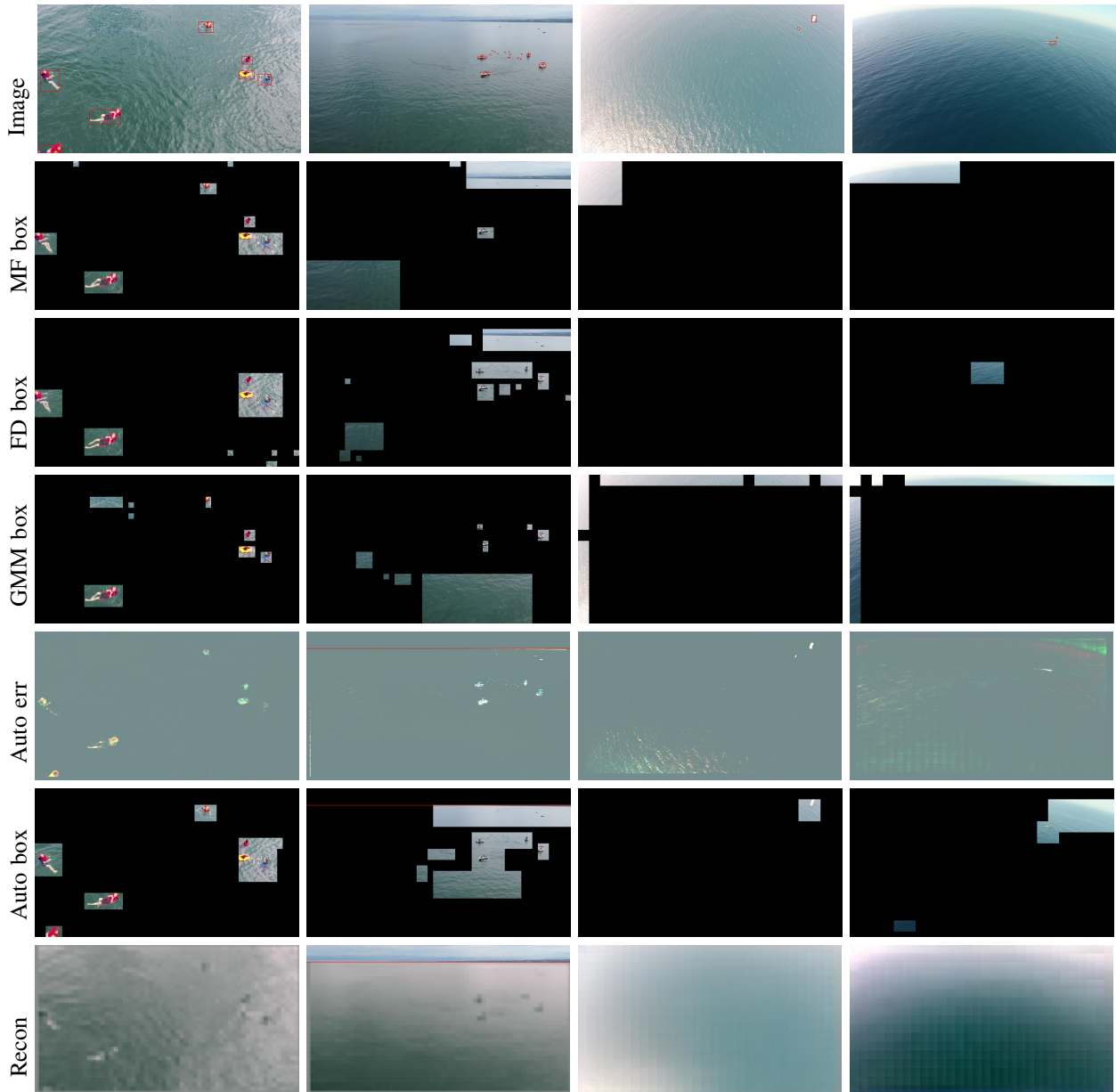


Fig. 5. Qualitative results for mean filter (MF), frame differencing (FD), Gaussian mixture model (GMM) and the autoencoder (Auto) on SeaDronesSee (left two columns) and Seagull (right two columns). For Auto, we plot the error heat map and the reconstructed image (Recon).

object detection [6], [8], [56]. Furthermore, we report the average recall (AR) averaged over 10 equidistant percentages p from $p = 0.05$ to $p = 0.95$, denoted AR .

A. Anomaly Detection Performance

Fig. 4 shows AR and the recall values for all the methods for the 10 transmission percentages p , while we interpolate in between. The autoencoder consistently outperforms the baselines for all values of p and for AR . However, the difference is especially visible for low values of p , which is the primary use case in this application scenario [16]. For example, for $p = 5\%$ the autoencoder achieves 70.1% and 40.0% recall for SeaDronesSee and Seagull, respectively, which is over 15 resp. 32 percent points more than the best

TABLE II
AVERAGE L_1 RECON. ERROR WITHIN BOXES AND OUTSIDE FOR $p = 5\%$.

	SeaDronesSee			Seagull		
	err_b	err_r	Δ_r	err_b	err_r	Δ_r
MF	78.3	0.3	78.0	0.37	0.3	0.07
FD	34.7	2.5	32.2	1.7	0.3	1.4
GMM	4.3	0.2	4.1	0.3	0.2	0.1
Auto	79.5	0.2	79.3	2.2	0.2	2.0

baseline. Subsequently, we focus on the case of low p .

We report the average reconstruction errors err_b (average L_1 reconstruction error within ground truth boxes), err_r (average L_1 reconstruction error rest) and their differences Δ_r in Table II. The autoencoder yields higher Δ_r , which

TABLE III

AUTOENCODER ABLATION EXPERIMENT ON SEADRONESSEE.

Future Frames	–	✓	✓	✓
Local Noise Remover	–	–	✓	✓
Frame Momentum	–	–	–	✓
$R^{p=5\%}$	60.3	66.2	68.6	70.1

shows its ability to discriminate better between normal and anomalous regions. Notably, the values for SeaDronesSee are generally much higher than for Seagull due to Seagull’s lower image quality and higher blurriness (see Figure 5).

Table III analyzes the influence of different components. When using future frames, we take the past four frames to predict the fifth. For frame momentum, we use the past two frames. It shows that using future frames yields the greatest benefits. All components improve the performance.

We analyze the influence of the horizon cutter on the performance of the autoencoder. As only SeaDronesSee incorporates meta data, we perform experiments on this data set. Only 5% of all frames actually show the horizon. Therefore, we restrict the influence of the horizon cutter to only that portion, as it does not have any on the other part. Remarkably, the autoencoder with horizon cutter achieves 86.3% recall whereas it only achieves 47.0% without. As the autoencoder only is trained on frames of open water, the different image statistics of the sky skew the image reconstruction error on these parts which expectedly results in a loss in performance. Both experiments used $p = 5\%$.

So far, we considered the case where we only have access to normal frames as training data. However, often we are given some labeled training data. Thus, we propose to use an adversarial training objective where we maximize the prediction penalty of the autoencoder within ground truth boxes and minimize it everywhere else. That way, the model is punished for learning to reconstruct actual anomalies. We evaluate this strategy by comparing it to its naive counterpart, i.e. not backpropagating the loss within boxes.

We compare these two approaches by training on the SeaDronesSee tracking train set and testing on the SeaDronesSee tracking test set. For $p = 5\%$, the ignoring yields a recall of 65.7% in contrast to 67.2% with adversarial loss.

B. Obtaining Fewer Bounding Boxes

Aside from the restriction of choosing at most $p\%$ of the frames, which may be imposed due to a potentially low bandwidth, another restriction may come from common video codecs’ inability to process a large number of regions of interest. Therefore, another type of restriction on a region proposer may be the number of regions it yields.

Thus, we propose to merge regions of interest touching each other at corners using Suzuki’s border following method [57]. As this may yield a larger than allowed area to be transmitted, each resulting box is ranked based on its reconstruction error. This leads to fewer and larger bounding boxes at the expense of a potentially lower recall.

TABLE IV

FEWER NUMBER OF BOUNDING BOXES (#B) VIA REDUCED RECALL ON SEADRONESSEE.

Method	Not Merging		Merging	
	#B	$R^{p=5\%}$	#B	$R^{p=5\%}$
MF [53]	65	54.8	7	49.6
FD [54]	65	55.0	8	53.2
GMM [55]	65	0.2	5	0.1
Auto	65	70.1	6	64.3

TABLE V

RUNNING TIMES IN FPS. BOLD VALUES DEPICT REAL-TIME METHODS.

	MF	FD	GMM	U-NET	CFLOW	Auto
1K	64	70	35	8	12	48
4K	50	62	17	1	3	27

Table IV shows the number of boxes #B and the recall for the standard and the merging method for $p = 5\%$. Note that without merging we have the same number of bounding boxes for all the methods since we allow 5% of the area of the image to be transmitted. We can substantially decrease the number of bounding boxes at the cost of a slightly lower recall. We note that this also highly depends on the anomaly distribution since for clustered anomalies it is easier to merge bounding boxes (see Fig. 5).

C. Running Times

Finally, we consider the running times of the individual methods on embedded hardware. We deploy them on an NVIDIA Xavier [58] mounted on a DJI Matrice 100. We transform all methods into optimized engines using TensorRT [59] and set the Xavier to MAX-N mode and report the running times averaged over 1000 frames. Table V shows the speed comparison between traditional and modern (U-NET [43], CFLOW [60]) methods. The much simpler baselines run in real-time, while the modern methods are slow.

For completeness, we replaced our architecture with the popular UNet architecture and trained it on SeaDronesSee using halved resolution and filter dimensions (more did not fit into a 3090Ti w/ 24GB). Interestingly, the performance trailed the performance of our method (78.1 AR). This led us to the conjecture that the high resolution is crucial in this application, which makes sense if we consider that many objects are of $\approx 20\text{px}$ size.

V. CONCLUSION AND OUTLOOK

We formulated the novel problem in maritime SaR of finding relevant regions of interest in a low-resource real-time and high-resolution scenario. We show that an autoencoder-based future frame prediction model is a promising direction even in a resource constrained setting. The benchmark is publicly available and we hope that the field of maritime SaR will be advanced by means of fast neural networks in the future.

REFERENCES

- [1] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, Q. Nie, H. Cheng, C. Liu, X. Liu, *et al.*, “Visdrone-det2018: The vision meets drone object detection in image challenge results,” in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.
- [2] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, “Uav-enabled intelligent transportation systems for the smart city: Applications and challenges,” *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, 2017.
- [3] E. Lygouras, N. Santavas, A. Taitzoglou, K. Tarchanidis, A. Mitropoulos, and A. Gasteratos, “Unsupervised human detection with an embedded vision system on a fully autonomous uav for search and rescue operations,” *Sensors*, vol. 19, no. 16, p. 3542, 2019.
- [4] B. Mishra, D. Garg, P. Narang, and V. Mishra, “Drone-surveillance for search and rescue in natural disaster,” *Computer Communications*, vol. 156, pp. 1–10, 2020.
- [5] H. Fan, L. Wen, D. Du, P. Zhu, Q. Hu, H. Ling, M. Shah, B. Wang, B. Dong, D. Yuan, *et al.*, “Visdrone-sot2020: The vision meets drone single object tracking challenge results,” in *European Conference on Computer Vision*. Springer, 2020, pp. 728–749.
- [6] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian, “The unmanned aerial vehicle benchmark: Object detection and tracking,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 370–386.
- [7] D. C. Tsouros, S. Bibi, and P. G. Sarigiannidis, “A review on uav-based applications for precision agriculture,” *Information*, vol. 10, no. 11, p. 349, 2019.
- [8] L. A. Varga, B. Kiefer, M. Messmer, and A. Zell, “Seadronessee: A maritime benchmark for detecting humans in open water,” *arXiv preprint arXiv:2105.01922*, 2021.
- [9] G. Bevacqua, J. Cacace, A. Finzi, and V. Lippiello, “Mixed-initiative planning and execution for multiple drones in search and rescue missions,” in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 25, 2015, pp. 315–323.
- [10] S. Hayat, E. Yanmaz, C. Bettstetter, and T. X. Brown, “Multi-objective drone path planning for search and rescue with quality-of-service requirements,” *Autonomous Robots*, vol. 44, no. 7, pp. 1183–1198, 2020.
- [11] S. Mayer, L. Lischke, and P. W. Woźniak, “Drones for search and rescue,” in *1st International Workshop on Human-Drone Interaction*, 2019.
- [12] D. Cazzato, C. Cimarelli, J. L. Sanchez-Lopez, H. Voos, and M. Leo, “A survey of computer vision methods for 2d object detection from unmanned aerial vehicles,” *Journal of Imaging*, vol. 6, no. 8, p. 78, 2020.
- [13] “SeaDronesSee Benchmark data acquisition,” <https://seadronessee.cs.uni-tuebingen.de/dataacquisition>, accessed: 2022-07-05.
- [14] U. of Tuebingen, “Avalon Project,” <https://seadronessee.cs.uni-tuebingen.de/avalon>, accessed: 2022-07-05.
- [15] T. Dortmund, “Larus Project,” <http://larus.kn.e-technik.tu-dortmund.de/>, accessed: 2022-07-05.
- [16] F. Steinert and B. Stabernack, “Architecture of a low latency h. 264/avc video codec for robust ml based image classification,” *Journal of Signal Processing Systems*, pp. 1–16, 2022.
- [17] Airbus, “Airbus Ship Detection Challenge,” <https://www.kaggle.com/c/airbus-ship-detection>, accessed: 2022-07-05.
- [18] K. Chen, M. Wu, J. Liu, and C. Zhang, “Fgsd: A dataset for fine-grained ship detection in high resolution satellite images,” *arXiv preprint arXiv:2003.06832*, 2020.
- [19] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, “A sar dataset of ship detection for deep learning under complex backgrounds,” *remote sensing*, vol. 11, no. 7, p. 765, 2019.
- [20] T. Zhang, X. Zhang, J. Li, X. Xu, B. Wang, X. Zhan, Y. Xu, X. Ke, T. Zeng, H. Su, *et al.*, “Sar ship detection dataset (ssdd): Official release and comprehensive data analysis,” *Remote Sensing*, vol. 13, no. 18, p. 3690, 2021.
- [21] M. M. Marques, P. Dias, N. P. Santos, V. Lobo, R. Batista, D. Salgueiro, A. Aguiar, M. Costa, J. E. da Silva, A. S. Ferreira, *et al.*, “Unmanned aircraft systems in maritime operations: Challenges addressed in the scope of the seagull project,” in *OCEANS 2015-Genova*. IEEE, 2015, pp. 1–6.
- [22] J. Scherer, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, H. Hellwagner, and B. Rinner, “An autonomous multi-uav system for search and rescue,” in *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, 2015, pp. 33–38.
- [23] G. Bradski, “The opencv library,” *Dr. Dobbs’s Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.
- [24] P. Rudol and P. Doherty, “Human body detection and geolocalization for uav search and rescue missions using color and thermal imagery,” in *2008 IEEE aerospace conference*. Ieee, 2008, pp. 1–8.
- [25] N. Ferreira and M. Silveira, “Ship detection in sar images using convolutional variational autoencoders,” in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020, pp. 2503–2506.
- [26] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, “Dota: A large-scale dataset for object detection in aerial images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3974–3983.
- [27] B. Kiefer, M. Messmer, and A. Zell, “Leveraging domain labels for object detection from uavs,” *arXiv preprint arXiv:2101.12677*, 2021.
- [28] E. Price, G. Lawless, R. Ludwig, I. Martinovic, H. H. Bülthoff, M. J. Black, and A. Ahmad, “Deep neural network-based cooperative visual tracking through multiple micro aerial vehicles,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3193–3200, 2018.
- [29] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.
- [30] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, “Segmentation as selective search for object recognition,” in *2011 international conference on computer vision*. IEEE, 2011, pp. 1879–1886.
- [31] Z. Zhong, L. Sun, and Q. Huo, “An anchor-free region proposal network for faster r-cnn-based text detection approaches,” *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 22, no. 3, pp. 315–327, 2019.
- [32] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, 2015.
- [33] P. Tang, X. Wang, A. Wang, Y. Yan, W. Liu, J. Huang, and A. Yuille, “Weakly supervised region proposal network and object detection,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 352–368.
- [34] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, “Comparative study of background subtraction algorithms,” *Journal of Electronic Imaging*, vol. 19, no. 3, p. 033003, 2010.
- [35] L. Deecke, R. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft, “Image anomaly detection with generative adversarial networks,” in *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 2018, pp. 3–17.
- [36] W. Sultani, C. Chen, and M. Shah, “Real-world anomaly detection in surveillance videos,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6479–6488.
- [37] T.-N. Nguyen and J. Meunier, “Anomaly detection in video sequence with appearance-motion correspondence,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1273–1283.
- [38] B. Staar, M. Lütjen, and M. Freitag, “Anomaly detection with convolutional neural networks for industrial surface inspection,” *Procedia CIRP*, vol. 79, pp. 484–489, 2019.
- [39] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, “Towards total recall in industrial anomaly detection,” *arXiv preprint arXiv:2106.08265*, 2021.
- [40] V. Saligrama and Z. Chen, “Video anomaly detection based on local statistical aggregates,” in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 2112–2119.
- [41] Y. Zhao, B. Deng, C. Shen, Y. Liu, H. Lu, and X.-S. Hua, “Spatio-temporal autoencoder for video anomaly detection,” in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1933–1941.
- [42] J. T. Zhou, J. Du, H. Zhu, X. Peng, Y. Liu, and R. S. M. Goh, “AnomalyNet: An anomaly detection network for video surveillance,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2537–2550, 2019.
- [43] W. Liu, W. Luo, D. Lian, and S. Gao, “Future frame prediction for anomaly detection—a new baseline,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6536–6545.

- [44] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, “Cutpaste: Self-supervised learning for anomaly detection and localization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9664–9674.
- [45] S. Szymanowicz, J. Charles, and R. Cipolla, “Discrete neural representations for explainable anomaly detection,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2022, pp. 148–156.
- [46] —, “Discrete neural representations for explainable anomaly detection,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 148–156.
- [47] W. Yu, Y. Lu, S. Easterbrook, and S. Fidler, “Efficient and information-preserving future frame prediction and beyond,” 2020.
- [48] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.
- [49] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, “Speeding-up convolutional neural networks using fine-tuned cp-decomposition,” *arXiv preprint arXiv:1412.6553*, 2014.
- [50] X. Zhuang and R. M. Haralick, “Morphological structuring element decomposition,” *Computer vision, graphics, and image processing*, vol. 35, no. 3, pp. 370–382, 1986.
- [51] C. F. Bohren and A. B. Fraser, “At what altitude does the horizon cease to be visible?” *American Journal of Physics*, vol. 54, no. 3, pp. 222–227, 1986.
- [52] T. Suzuki, Y. Takahashi, and Y. Amano, “Precise uav position and attitude estimation by multiple gnss receivers for 3d mapping,” in *Proceedings of the 29th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2016)*, 2016, pp. 1455–1464.
- [53] R. Zhang and J. Ding, “Object tracking and detecting based on adaptive background subtraction,” *Procedia Engineering*, vol. 29, pp. 1351–1355, 2012.
- [54] S. S. Mohamed, N. M. Tahir, and R. Adnan, “Background modelling and background subtraction performance for object detection,” in *2010 6th International Colloquium on Signal Processing & its Applications*. IEEE, 2010, pp. 1–6.
- [55] Z. Zivkovic, “Improved adaptive gaussian mixture model for background subtraction,” in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2. IEEE, 2004, pp. 28–31.
- [56] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, “Vision meets drones: A challenge,” *arXiv preprint arXiv:1804.07437*, 2018.
- [57] S. Suzuki *et al.*, “Topological structural analysis of digitized binary images by border following,” *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.
- [58] Nvidia, “Nvidia Xavier,” <https://www.nvidia.com/de-de/autonomous-machines/embedded-systems/jetson-agx-xavier/>, accessed: 2022-07-05.
- [59] H. Vanholder, “Efficient inference with tensorsrt,” in *GPU Technology Conference*, vol. 1, 2016, p. 2.
- [60] D. Gudovskiy, S. Ishizaka, and K. Kozuka, “Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 98–107.