Applied Ocean Research 139 (2023) 103709

Contents lists available at ScienceDirect

Applied Ocean Research

journal homepage: www.elsevier.com/locate/apor

Image segmentation in marine environments using convolutional LSTM for temporal context

Kasper Foss Hansen [a], Linghong Yao [b], Kang Ren [a], Sen Wang [c], Wenwen Liu [d], Yuanchang Liu [a,*]

[a] Department of Mechanical Engineering, University College London, Torrington Place WC1E 7JE, UK
[b] Department of Computer Science, University College London, 66-72 Gower Street, WC1E 6EA, UK
[c] Department of Electrical and Electronic Engineering, Imperial College London, UK
[d] School of Automation, Nanjing University of Information, Science and Technology, China

ARTICLE INFO

Keywords:
Unmanned surface vehicles (USVs)
Image segmentation
Obstacle detection
Temporal context
Long short-term memory (LSTM)

ABSTRACT

Unmanned surface vehicles (USVs) carry a wealth of possible applications, many of which are limited by the vehicle's level of autonomy. The development of efficient and robust computer vision algorithms is a key factor in improving this, as they permit autonomous detection and thereby avoidance of obstacles. Recent developments in convolutional neural networks (CNNs), and the collection of increasingly diverse datasets, present opportunities for improved computer vision algorithms requiring less data and computational power. One area of potential improvement is the utilisation of temporal context from USV camera feeds in the form of sequential video frames to consistently identify obstacles in diverse marine environments under challenging conditions. This paper documents the implementation of this through long short-term memory (LSTM) cells in existing CNN structures and the exploration of parameters affecting their efficacy. It is found that LSTM cells are promising for achieving improved performance; however, there are weaknesses associated with network training procedures and datasets. Several novel network architectures are presented and compared using a state-of-the-art benchmarking method. It is shown that LSTM cells allow for better model performance with fewer training iterations, but that this advantage diminishes with additional training.

1. Introduction

In recent years, heavy development in unmanned surface vehicles (USVs) has been motivated by both commercial and scientific ambitions. USVs are typically small-scale vessels (Fig. 1) developed for tasks which would otherwise be impractical, dangerous, or tedious for manned vehicles. Their applications range widely from hydrographic surveying and data collection to disaster management and mine-sweeping (Liu et al., 2016).

Heightening the level of vehicle autonomy is key to unlocking new applications of USVs. One factor of this is the autonomous navigation and obstacle avoidance which can be challenging, especially in shoreline environments where other vessels, buoys, animals, and swimmers are commonly present. Identifying these from on-board sensors is an important task, but factors like waves, reflections, and diverse weather conditions hinder this for example by obscuring clear boundaries between obstacles and surroundings. Furthermore, the addition heavy and expensive sensors eliminate the size, cost, and manoeuvrability benefits of USVs over manned vessels. Hence, solutions to obstacle detection using simple, on-board monocular cameras alone are sought.

One approach to effective perception, or "computer vision", using only monocular camera inputs, is to use convolutional neural networks (CNNs) for identification of obstacles in an image. These networks can be trained on a range of datasets to achieve impressive performance in a variety of tasks (Garcia-Garcia et al., 2017). In USV computer vision applications, CNNs have lately been heavily developed and used to break new ground, but limitations to their performance are not negligible. In particular, phenomena characteristic to marine environments often inhibit model effectiveness. For example, CNNs may misidentify object reflections as real obstacles (Bovcon and Kristan, 2022), thereby impeding any navigational software which would have to avoid said reflections as it would with obstacles. A solution to this could be the introduction of a temporal element in the neural network, with the rationale that reflections and other visual features in the water are warped by waves and ripples over time. Practically, this means allowing a CNN model to observe several video frames and training it to

* Corresponding author.
E-mail address: yuanchang.liu@ucl.ac.uk (Y. Liu).

https://doi.org/10.1016/j.apor.2023.103709
Received 9 May 2023; Received in revised form 14 July 2023; Accepted 18 August 2023
Available online 26 August 2023

Fig. 1. Unmanned Surface Vehicle (USV) selection from Seafloor Systems, Inc. Image source: (Seafloor Systems).

distinguish water features such as reflections, boat wakes, and glitter from real obstacles via their variation in appearance over time. If successful, this would reduce the amount of water features incorrectly labelled as obstacles by the computer vision model, thereby increasing the overall reliability of the model.

The objective of this paper is to explore and demonstrate the utility of implementing recurrent network elements in an established CNN model to allow it to draw information from the temporal attributes of video inputs. Specifically, long short-term memory (LSTM) cells are integrated into the ShorelineNet model (Yao et al., 2021) with the goal of reducing detection of time-varying water phenomena, like reflections and glitter, as real obstacles. Different model architectures and data manipulation approaches are trialled, and the results are compared to both the original model and other state-of-the-art methods. The project code is based on the work of Yao et al. (2021) and is publicly available.[1] All models are trained and evaluated on open-source datasets using recently developed benchmarks. It is shown how the novel implementation of convolutional LSTM cells reduces the incorrect detection of time-varying water phenomena as obstacles, but also that the network architecture, dataset characteristics, and training time may result in overfitting to temporal context data. The experiments carried out in this study add to the tools available to researchers developing USV computer vision algorithms and show improvements over existing lightweight models. Convolutional LSTM has, to the author's knowledge, not been used in this application before.

The main contributions of this paper can be summarised as: (1) Convolutional LSTM has been innovatively integrated with the ShorelineNet model to improve obstacle detection robustness by capturing sequential information, an approach which is aligned and motivated by the nature of maritime visual dataset (video frames are sequential and not independent and identically distributed); (2) the proposed methods can successfully reduce false positive obstacle detections which is one of the main issues associated with ShorelineNet, with the capacity to provide an accurate detection of obstacles that are difficult to identify due to environmental influences while remaining lightweight enough to run in real-time at high frame rates; (3) enriched experiments and cross-dataset validation have been undertaken to demonstrate the performances of the proposed methods and recommendations for practical application are provided.

Fig. 2. Example of image captured on-board a USV. In this example both strong glare and obstacle reflections in the water make the scene challenging for computer interpretation Bovcon et al. (2022).

The rest of the paper is organised as follows: Section 2 provides a comprehensive literature review into USVs and their related CV research. Section 3 describes the main architecture of the proposed networks with Section 4 detailing the training and evaluation processes. Section 5 discusses the main results and provides enriched comparative studies against SOTA. Section 6 concludes the paper and points out future research directions.

2. Literature review

Reductions in fuel and crew costs are some amongst many motivations behind recent developments in USV computer vision (CV) (Vagale et al., 2021). Due to size, cost, weight, and power constraints well-established methods from autonomous land vehicles are not directly transferable, and novel solutions specific to USVs are required. The below literature review serves as a theoretical introduction to the tools utilised in this paper, followed by a review of recent relevant developments in the field.

[1] https://github.com/KFH22/ShorelineNet

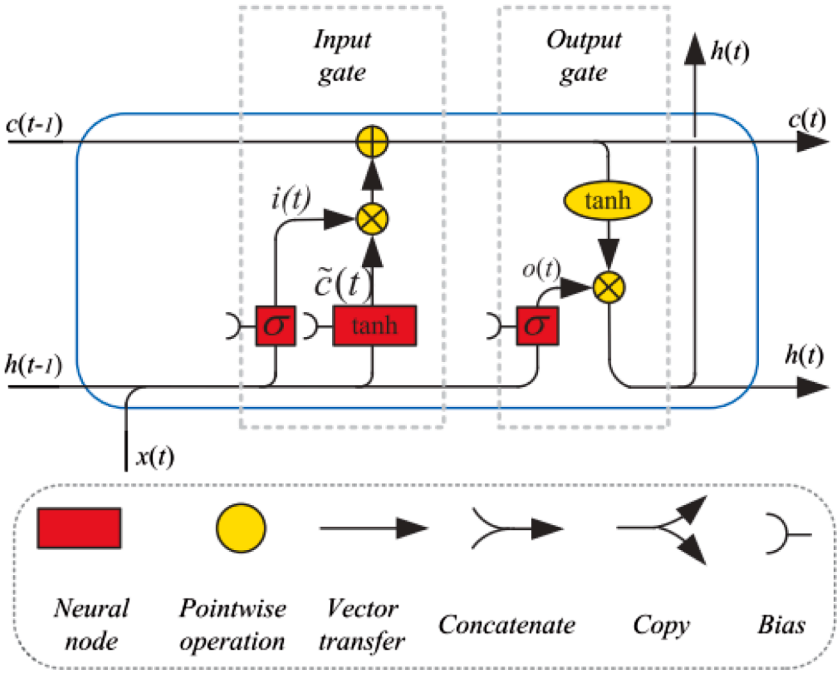

Fig. 3. The basic structure of the original LSTM architecture Yu et al. (2019).

2.1. USV-specific challenges

Cameras have become the sensor of choice for small-scale USVs, as these are generally cheap and lightweight, and have low power consumption when compared to alternatives like LIDAR. In addition, cameras generally allow better perception of small, close, or flat objects in the water, and have a large field of view (Kristan et al., 2016; Liu et al., 2021). As such, the information produced by cameras is rich, but its usefulness is highly dependent on the algorithms interpreting their signals. Images taken by these in marine environments are likely to include effects like reflections, haze, and glare (Fig. 2) which add to the challenge. Therefore, traditional obstacle detection algorithms using background subtraction, and machine learning (ML) models developed for autonomous cars, are often ineffective in marine environments (Prasad et al., 2019; Bovcon et al., 2022). To address this, CV models have been proposed which use CNNs that are trained on tailor-made maritime image datasets. These serve as an alternative to adding additional sensors like inertial measurement units (IMUs) or stereo cameras, which have otherwise shown promising performance (Huntsberger et al., 2011; Bovcon et al., 2018).

2.2. Convolutional neural networks for semantic segmentation

One method for interpreting camera data is to classify each image pixel such that the image is segmented into regions. This method, called semantic segmentation, is widely used in many computer vision applications. Semantic segmentation is generally more computationally demanding than whole-image classification, but its possible applications range widely. Recent developments in deep learning (DL) have given rise to ground-breaking performance of semantic segmentation models (Minaee et al., 2022) often using CNNs. These, first proposed by Fukushima (1980), are a type of deep neural networks exceptionally potent at CV tasks. They work through stacks of neural network layers with at least one convolutional filter application. This filtering is commonly used in image processing, and allows the CNN to extract important features, such as edges, from the image (Prince, 2012). Another benefit of CNNs over conventional fully connected neural networks, is a significant reduction in network parameters (weights and biases), as network weights are shared within layers (Minaee et al., 2022).

Several CNN structures for semantic segmentation have been proposed, many of which utilise an encoder-decoder shape. Here, an encoder consisting of convolutional layers is followed by a decoder of transposed convolutional (deconvolutional) layers. Well-known examples of this structure are "SegNet" (Badrinarayanan et al., 2017) and "U-Net" (Ronneberger et al., 2015). Both models use their encoder to extract coarse image information while retaining finer details through different forms of "skip connections" (He et al., 2016) which connect corresponding encoder-decoder layers. Another commonly seen feature in encoder-decoder architectures is utilisation of encoders pretrained on large datasets, such as "ImageNet" (Deng et al., 2009), in models applied to different data. This may retain good performance while reducing training time and avoiding overfitting on small datasets.

2.2.1. Temporal context and LSTMs

Current obstacle segmentation models for marine environments generally work on a single image, and very few deep learning architectures (e.g. Žust and Kristan (2022b)) utilise the temporal information of a video-feed. However, the time-dependent nature of phenomena like reflections, as well as pitch/yaw/roll of the vehicle, could prove beneficial in correctly identifying objects and avoiding misidentification of water features as obstacles.

In practice, a model architecture and data pipeline can be designed with an input of one video frame and N preceding frames (or "pre-frames") containing additional information. Different approaches have been attempted to effectively utilise said information; Karpathy et al. (2014) set $N+1$ sequential video frames as the input in a classification problem using a conventional CNN and saw slight improvement over using single frames. Varghese et al. (2021) and Liu et al. (2020) trained models using custom loss-functions to penalise inconsistencies between frames in training, before applying this to single-frame inference thereby avoiding latency associated with multi-frame inference. A third option is using a recurrent neural network (RNN) with several input frames drawing on information from previous frames when analysing the current one. In particular "Long Short-Term Memory" (LSTM) networks (Hochreiter and Schmidhuber 1997) maintain a dynamic cell state as the network is applied to several input time steps sequentially. In practice, this gives the network "memory" of previous inputs while analysing

Table 1
Examples of benefit from temporal context and LSTM implementation in different applications.

Method	Improvement over baseline
CNN access to a total of 10 sequential frames (Karpathy et al., 2014)	Correct prediction of activity in 60.9% of videos in the Sports-1 M dataset against 59.3% in baseline
Unsupervised temporal consistency loss (Varghese et al., 2021)	Substantial improvement in consistency between segmentations of sequential frames of the cityscapes dataset with a small loss in absolute accuracy.
Temporal loss and temporal consistency knowledge distillation in training (Liu et al., 2020)	Substantial improvements in both temporal consistency and absolute accuracy of several models in semantic segmentation of different datasets
Deep bidirectional LSTM (Ullah et al., 2018)	Significant improvement in action recognition against other state-of-the-art methods when tested on several datasets
Encoder-decoder ConvLSTM network for sequential frames (Zou et al., 2020)	Substantial improvement in road lane detection accuracy over single frame methods when tested on two extensive datasets
Mobilenet V2 and LSTM for skin disease classification (Srinivasu et al., 2021)	While remaining computationally efficient, a clear benefit of adding LSTM to the Mobilenet V2 architecture was found

current data. Several LSTM structures have been proposed, but the original version is explained below with reference to Fig. 3.

x(t): *input* to the module, in our case the information contained in an image.

h(t): *output* of the LSTM cell containing the result of operations applied throughout the cell.

c(t): *cell state*, which contains the information that persists between time steps, and can be thought of as the *memory* of the cell.

In general, as information is passed through the LSTM module, its cell state is modified by letting inputs pass through "gates". The weights and biases of these gates are trained along with the rest of the network to let relevant information pass through. This allows the LSTM cell to store information and us it in the inference of a later time step.

Since their introduction, LSTM networks have been responsible for most RNN breakthroughs (Yu et al., 2019; Van Houdt et al., 2020). They have been used in diverse applications from financial market forecasting (Sezer et al., 2020) to "remaining useful life" assessment (Wu et al., 2018), and they can be exceptionally powerful in text, speech and language modelling (e.g. (Sundermeyer et al., 2015; Liu and Guo 2019)). When applied to images and video, as is the case in computer vision, LSTM networks often benefit from being integrated in CNN architectures (Van Houdt et al., 2020) and have been effective in e.g. classification of video content (Ullah et al., 2018), road-lane detection (Zou et al., 2020), and skin disease identification (Srinivasu et al., 2021). In Table 1 an overview of several implementations of temporal context and LSTM is given with the purpose of elaborating on the benefit found in different image (or video) segmentation or classification applications. Approaches with and without LSTM are given to illustrate the breadth possible solutions currently being studied.

2.2.1.1. LSTM location. In image applications, a common approach is placing LSTM layers at different locations in a proven CNN structure often consisting of an encoder and decoder. Commonly, the LSTM layer (s) are positioned between the encoder and decoder to maintain consistency in the coarse image elements between frames (Pfeuffer et al., 2019). Alternatively, the LSTM modules can be incorporated in the skip connections (Rochan, 2018) or added after the deconvolution (Xu et al., 2019). Pfeuffer et al. (2019) experimented with the LSTM location using a SegNet (Badrinarayanan et al., 2017) model on the cityscapes dataset (Cordts et al., 2016), and found that a "ConvLSTM" layer positioned after the decoder performed slightly better than when located between the encoder and decoder.

2.2.1.2. LSTM cell variations. Several variations to the original LSTM cell have been suggested. Gers et al. (1999) added a "forget gate" to rid the cell state of irrelevant information and later proposed "peephole connections" in the presence of long time lags (Gers et al., 2002). Chung et al. (2014) suggested a gated recurrent unit (GRU), which is more computationally efficient than an LSTM cell, and proved its performance to be similar to conventional LSTMs. Many other variants have been suggested, but the original LSTM is by far the most broadly used and has been implemented in many ML APIs and platforms.

2.2.1.3. Convolutional LSTM layers. Conventional LSTM cells use multiplications when applying weights to feature maps in the connections. However, for spatiotemporal data, such as image sequences, convolutional operations in the LSTM cell were found to be beneficial, as the dimensionality of images could be retained (Yu et al., 2019). This was first introduced as a "convolutional LSTM cell" (ConvLSTM) by Shi et al. (2015) and has since been implemented in various contexts.

2.3. USV computer vision developments

The development of CV models for maritime environments is a field of intense research, with USVs being applications of major interest. Established methods, e.g. using background subtraction, perform extremely poorly on challenging maritime image datasets (Prasad et al., 2019). Contrarily, current CNN models indicate the possibility of impressive performance. This has prompted rapid exploration into methodologies, as well as collection of increasingly large and well-annotated datasets, as the existence of these is a pre-requisite for effective model training and testing.

2.3.1. USV CV models

Early models focused on detecting the horizon in an input image (Fefilatyev et al., 2006) – an approach used by e.g. Wang et al. (2011) to first detect the horizon before looking for obstacles below it. This method however, falls short in environments close to the shore, where land-masses may obscure the horizon. Kristan et al. (2016) improved upon this weakness by proposing a model based on an assumption of water, sky, and obstacle/fog/shore regions in vertically distinct areas of the input image. They then applied a Markov Random Field model to produce real-time semantic segmentation of input images. This model was further improved by using stereo image and IMU inputs (Bovcon et al., 2017, Bovcon et al., 2018). These efforts effectively created benchmarks for models only using monocular camera data to beat. To do this, Lee et al. (2018) demonstrated how well-established convolutional neural network (CNN) models could be trained on USV-captured datasets to strongly improve their performance in the space. This philosophy has been expanded on with modifications to network architectures in addition to bespoke datasets. For example, a combination of ResNet (He et al., 2016) and DenseNet (Huang et al., 2017) by Liyong et al. (2020) and a modified ENet (Paszke et al., 2016) by Kim et al. (2019) both showed greatly improved results over the unmodified networks.

The most recent contributions are even more impressive. Bovcon and Kristan (2022) proposed the WaSR (Water Segmentation and Refinement) model which was specifically designed for marine environments. It is currently the leading model in several benchmarks and has recently been expanded further by the utilisation of preceding video frames for temporal context (Žust and Kristan 2022b). Similarly, Liu et al. (2021) proposed the combination of a novel horizon-detection method and context from adjacent frames to substantially outperform, amongst others, the SSM model proposed by Kristan et al. (2016) using only a monocular camera input. Chen et al. (2021) proposed the WODIS model which improves on obstacle detection robustness, and performs particularly well on the SMD dataset Prasad et al. 2016). Finally, Xue et al. (2021) proposed a novel model using a simple linear iterative clustering algorithm to improve accuracy of segmentation in the edge-regions

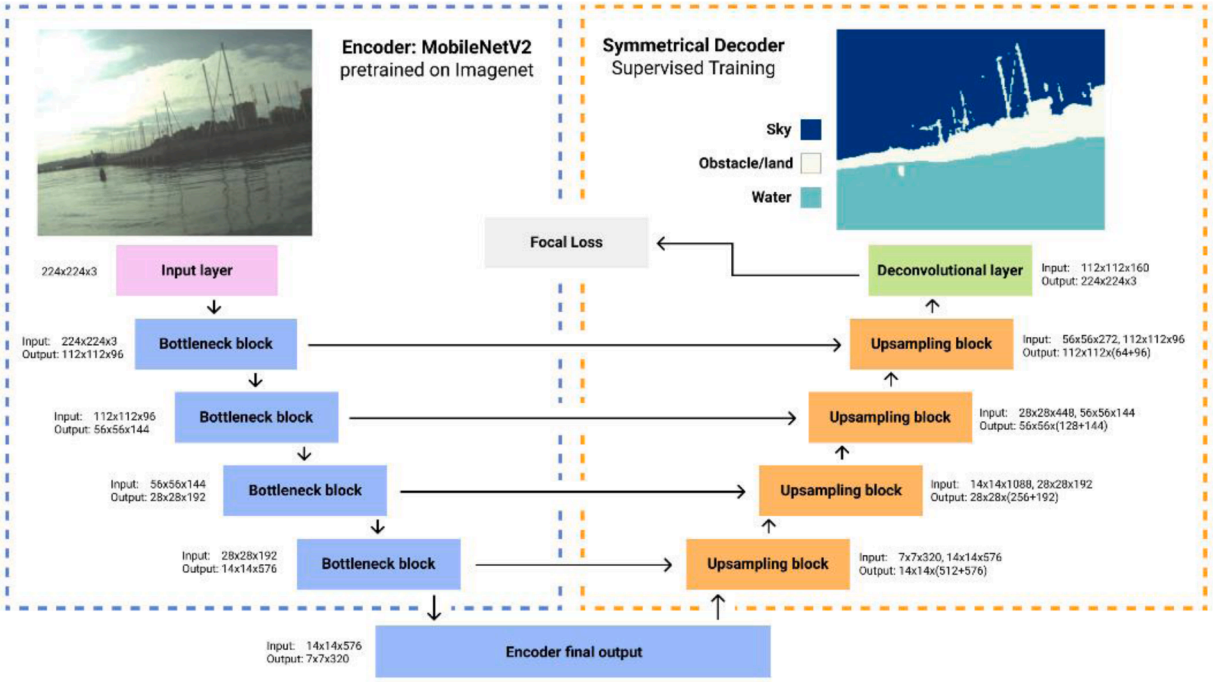

Fig. 4. The ShorelineNet model architecture. A pretrained encoder (blue) from MobileNetV2 is used with a custom decoder (orange) and skip connections between the two to form a structure similar to the U-Net. Figure source: Yao et al. (2021).

between obstacles and water.

The "ShorelineNet" model proposed by Yao et al. (2021) holds particular interest to this work. It accomplishes good segmentation performance but suffers from false positive detections of obstacles caused primarily by glare and reflections. When compared to e.g. the WaSR model, acceptable F1-score is achieved with a model containing ~5% the number of trainable parameters of WaSR. This simplicity is the main focus of the model, hence it is built on the particularly lightweight MobileNetV2 encoder (Sandler et al., 2018). Fig. 4 shows the structure of ShorelineNet. It consists of an encoder-decoder CNN with skip connections, and is trained using a focal cross-entropy loss function which was shown to somewhat reduce false positive detections. The MobileNetV2 encoder is pretrained on the ImageNet dataset (Deng et al., 2009).

Two areas of current research which need addressing can be summarised. Firstly, many CV models struggle to avoid false positive obstacle detections due to environmental influences like reflections. Recent advances have addressed this issue, but they typically rely on larger and more computationally intensive models to improve performance. Secondly, the sequential nature of USV on-board video data and temporal character of many environmental influences is largely unused in the reduction of false positive detections. To the authors' knowledge, only Žust and Kristan (2022b) has used the temporal context from video data to improve this aspect. LSTM is an obvious approach to this and has, in other areas, been used successfully to exploit sequential data, but this has yet to be implemented in USV CV.

2.3.2. Datasets

Several marine environment training and evaluation datasets have been proposed. The development of effective models heavily depends on the quality of annotated image/video data available, but collecting these datasets in varying locations, seasons, and weather conditions is resource intensive. Training datasets for semantic segmentation models are especially tedious to create, as they require pixel-wise labelling of image regions. Hence, many datasets are annotated only with obstacle bounding boxes and water-sky/land boundaries (water-edges) instead. Approaches to training segmentation models on datasets annotated in this way have been proposed (Žust and Kristan 2022a), but are not being considered in this report.

For training, the MaSTr1325 dataset (Bovcon et al., 2019) contains 1325 images and pixel-wise annotations captured on a USV in coastal waters in Slovenia. It was expanded upon by the MaSTr1478 dataset (Žust and Kristan 2022b) which adds 153 new images with particularly challenging reflection conditions. However, these are not always captured from a camera comparable in specifications and location to those on USVs. In addition, the MaSTr1478 set contains 5 un-annotated preceding frames for each annotated one for the purpose of providing

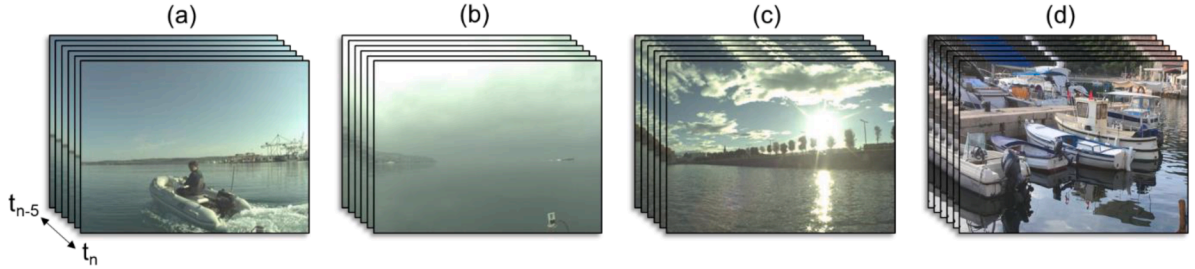

Fig. 5. Examples of image sequences from the MaSTr1478 dataset. (a) through (c) are examples of images from the MaSTr1325 dataset captured on a USV in varying conditions. (d) is an example of the 153 additional images in MaSTr1478 with challenging reflections. It is captured at a significantly different height to the other images. Images from Bovcon et al. (2019), Žust and Kristan (2022b).

Fig. 6. Examples of evaluation images from the MODS dataset with fog (a), glare (b), close-by obstacles (c), and reflections (d). Images from Bovcon et al. (2022).

Table 2
MODS benchmark results of selected segmentation models Bovcon et al. (2022). Each model is evaluated on water-edge detection root mean squared error (RMSE), true positives per 100 images (TPr), false positives per 100 images (FPr), and F1-score.

Model	Trainable parameters	RMSE (pixels)	TPr	FPr	F1
ISSM (Bovcon et al., 2018)	-	181	55.3	44.6	67.1%
ENet (Paszke et al., 2016)	0.4M	78	62.6	42.0	73.8%
PSPNet (Zhao et al., 2016)	56.0M	21	59.4	26.2	78.9%
MobileUNet (Howard et al., 2017)	8.9M	35	54.8	14.5	81.6%
SegNet (Badrinarayanan et al., 2017)	35.0M	23	57.7	15.0	83.8%
DeepLab3+ (Chen et al., 2018)	48.0M	21	60.2	15.1	85.9%
BiSeNet (Yu et al., 2018)	47.5M	**17**	58.4	6.1	90.3%
RefineNet (Lin et al., 2017)	85.7M	18	**60.4**	7.4	91.0%
DeepLab panoptic (Cheng et al., 2020)	46.7M	**17**	59.9	6.6	91.2%
WaSR (Bovcon and Kristan 2022)	84.6M	21	56.8	**2.6**	**91.4%**

temporal context to CV models. Examples of images are given in Fig. 5.

For evaluation, the MODS dataset (Bovcon et al., 2022) is currently the most comprehensive and challenging dataset. It is a set of ~81k stereo images with ~60k annotated objects. It was captured on a USV travelling at a maximum of 2.5 m/s in various locations, conditions, and seasons. The images are recorded at 10 frames per second (fps), and every 10th frame is annotated with bounding boxes and water-edges. This results in ~8k annotated images at 1 fps, which are used in a standardised evaluation procedure. Examples of images are given in Fig. 6.

2.3.3. Model evaluation measures

Currently, the MODS benchmark (Bovcon et al., 2022) is the most comprehensive performance benchmark for USV CV models. It consists of a statistical evaluation of the given model's performance on semantic segmentation of the MODS dataset images. Performance is measured in water-edge location accuracy and ability to detect obstacles protruding from the water surface. In addition to collecting the dataset and developing the benchmark procedure, Bovcon et al. (2022) documented the benchmark performance of various CV models. Selected results of this are shown in Table 2.

In summary of the literature review, monocular cameras have superior features as sensors for USV perception, but challenges characteristic to marine environments inhibit the effectiveness of conventional CV algorithms. Advancements in deep learning and CNNs have proved useful, and several effective CV models have been developed. One challenge limiting model effectiveness is the presence of reflections, glare, and other marine phenomena, but temporal context from preceding camera frames may improve on this weakness e.g., using LSTM cells. An increase in dataset availability could allow developments of lightweight models exploiting such techniques to achieve excellent detection accuracy with little computational cost.

3. Proposed model architecture and data augmentation

A CV model is proposed to utilise the temporal context from a USV camera feed to sequence images into regions of water, sky and obstacle. The model is based on the "ShorelineNet" (Yao et al., 2021), and retains its overall purpose and structure. The objective of reaching segmentation performance comparable to state-of-the-art models with reduced computational load is retained. An approach of using recurrent LSTM components is used, which has, to the author's knowledge, not been applied in marine CV problems before. The main objective of this model is to explore the utility of implementing recurrent components, specifically convLSTM cells, in established CNN models. Different parameters and network architectures will be trialled. Naturally, an improved model performance is sought, but results will be documented for all models as the work is exploratory in nature. The original ShorelineNet model will be used as a baseline for performance, and references will be made to other models for context.

3.1. Neural network

The proposed model (Fig. 7) is named "ShorelineNet-ConvLSTM". It is created by introducing a convLSTM block between the ShorelineNet encoder and decoder. The encoder remains the pretrained MobileNetV2 network, and the decoder consists of 4 blocks with each a deconvolution, batch normalisation, dropout, and activation layer as shown in Fig. 7 (b). Skip connections are established between the encoder and decoder by concatenating selected encoder and decoder block outputs. After the final decoder block, a transposed convolutional layer re-establishes the original image dimensionality for segmentation. Positioning the LSTM layer after the encoder is common practice and is motivated by the encoder first extracting the high-level features before passing them onto the ConvLSTM layer in sequence. Hence, the ConvLSTM layer should be able to infer the complex nature of reflections and environmental effects.

The convLSTM block consists of a single convLSTM layer and a batch normalisation layer for more robust training (Ioffe and Szegedy 2015). The convLSTM layer effectively consists of $N+1$ sequential convLSTM cells where $N+1$ is the length of each image sequence (Fig. 8 (a)). Each convLSTM cell (Fig. 8 (b)) is a standard LSTM cell with forget gate (Gers et al., 1999) with convolutional operations instead of multiplications of weights to feature maps (Shi et al., 2015).

Equations (1) through (6) mathematically describe the operations done on feature maps with reference to Fig. 8 (b). Convolution operations are denoted with *. Weights and biases are denoted with W and b respectively, and sigmoid and hyperbolic tan activation functions are shown with σ and "tanh" respectively.

$$f_n = \sigma\left(W_{fh} * h_{n-1} + W_{fx} * x_n + b_f\right) \quad (1)$$

$$i_n = \sigma\left(W_{ih} * h_{n-1} + W_{ix} * x_n + b_i\right) \quad (2)$$

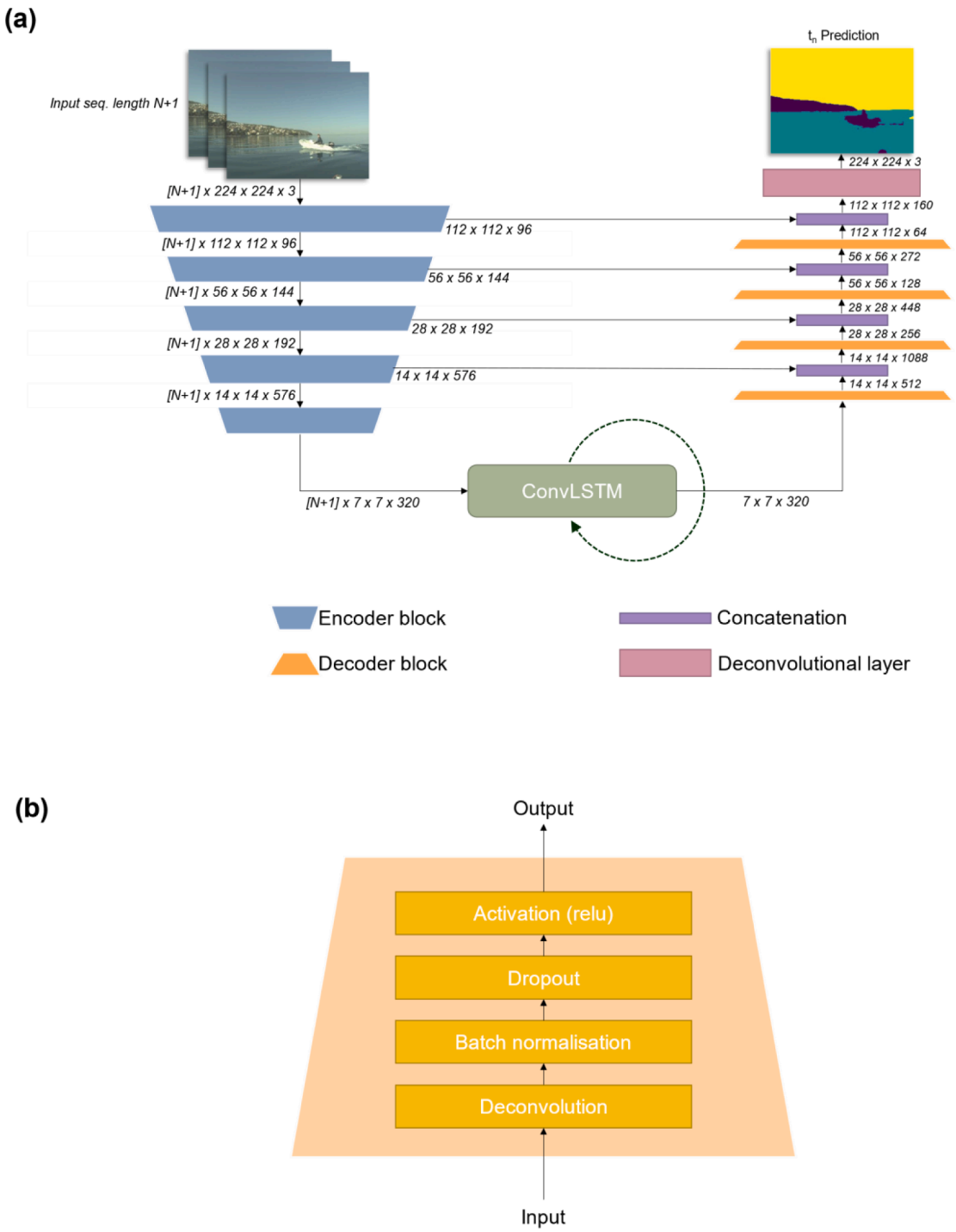

Fig. 7. (a): ShorelineNet-ConvLSTM architecture. Feature map dimensions are given at connections with N being the number of preceding frames used giving an image sequence length of N+1. (b): Decoder block containing a transposed convolutional (deconvolution) layer, batch normalisation layer, dropout layer, and activation layer using the rectified linear unit (relu) activation function.

$$c_n^{'} = \tanh(W_{c'h} * h_{n-1} + W_{c'x} * x_n + b_{c'}) \tag{3}$$

$$c_n = f_n c_{n-1} + i_n c_n^{'} \tag{4}$$

$$o_n = \sigma(W_{oh} * h_{n-1} + W_{ox} * x_n + b_o) \tag{5}$$

$$h_n = o_n \tanh(c_n) \tag{6}$$

In the encoder only the current frame feature map persists, not the data from preceding frames. The complete, "unrolled" recurrent network is therefore as shown in Fig. 9.

With reference to Figs. 8 and 9, the ConvLSTM block works by compiling feature maps from each pre-frame and the current frame produced by the encoder into a single output which is passed to the decoder. In Fig. 9, the "unrolled" RNN structure illustrates the connections between layers. Here it can be seen that the pre-frames are only used for temporal information in the ConvLSTM block by allowing it to establish a cell state relating to the feature maps inferred. The current frame however, also has its feature maps passed through skip connections to the decoder around the ConvLSTM block in order to retain

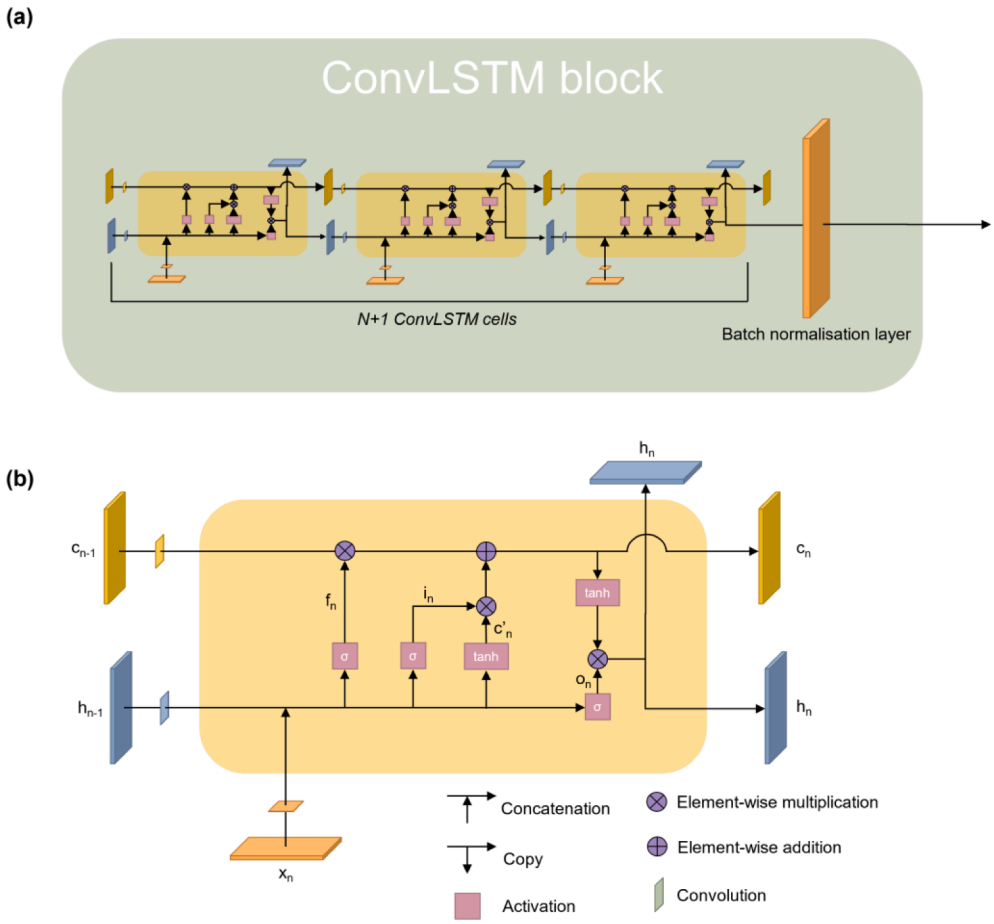

Fig. 8. (a): Complete convLSTM block containing the convLSTM layer and a batch normalisation layer (b): The internal structure of a convLSTM cell. The single cell is repeated N+1 times for the convLSTM layer where N+1 is the length of the input image sequence. *c* is the cell state, *h* is the output, *x* is the input, *n* is the frame timestep.

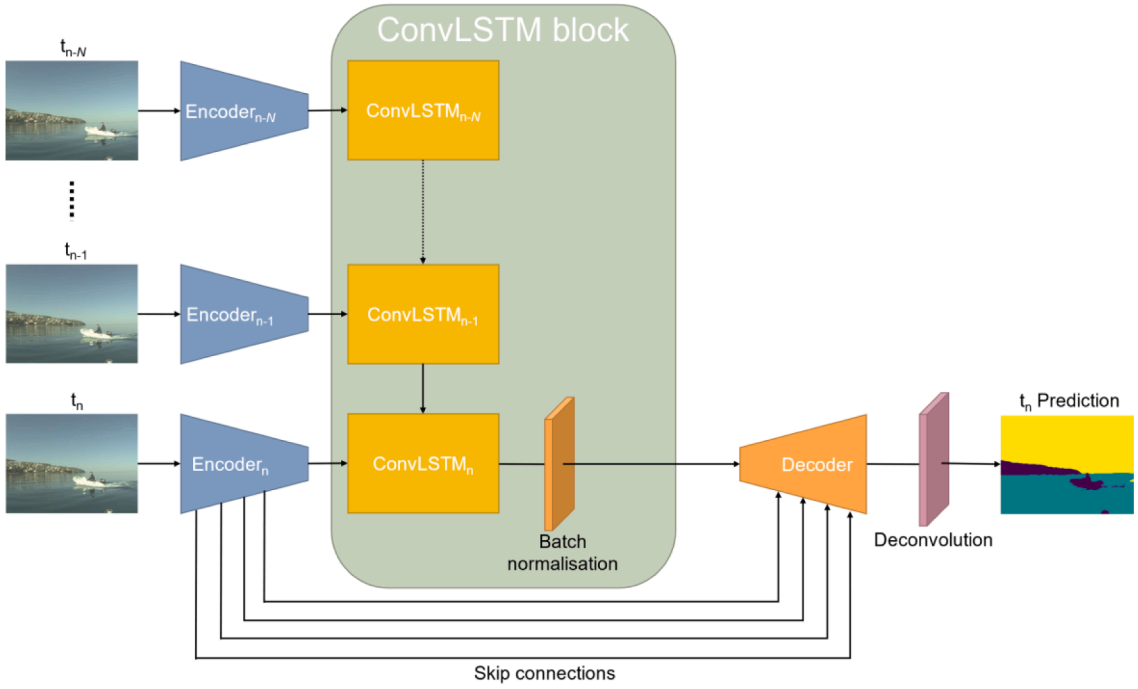

Fig. 9. "Unrolled" network structure of ShorelineNet-ConvLSTM. The green ConvLSTM block is equal to that of Fig. 8 (a) and is the element which extracts temporal information from the frame sequence.

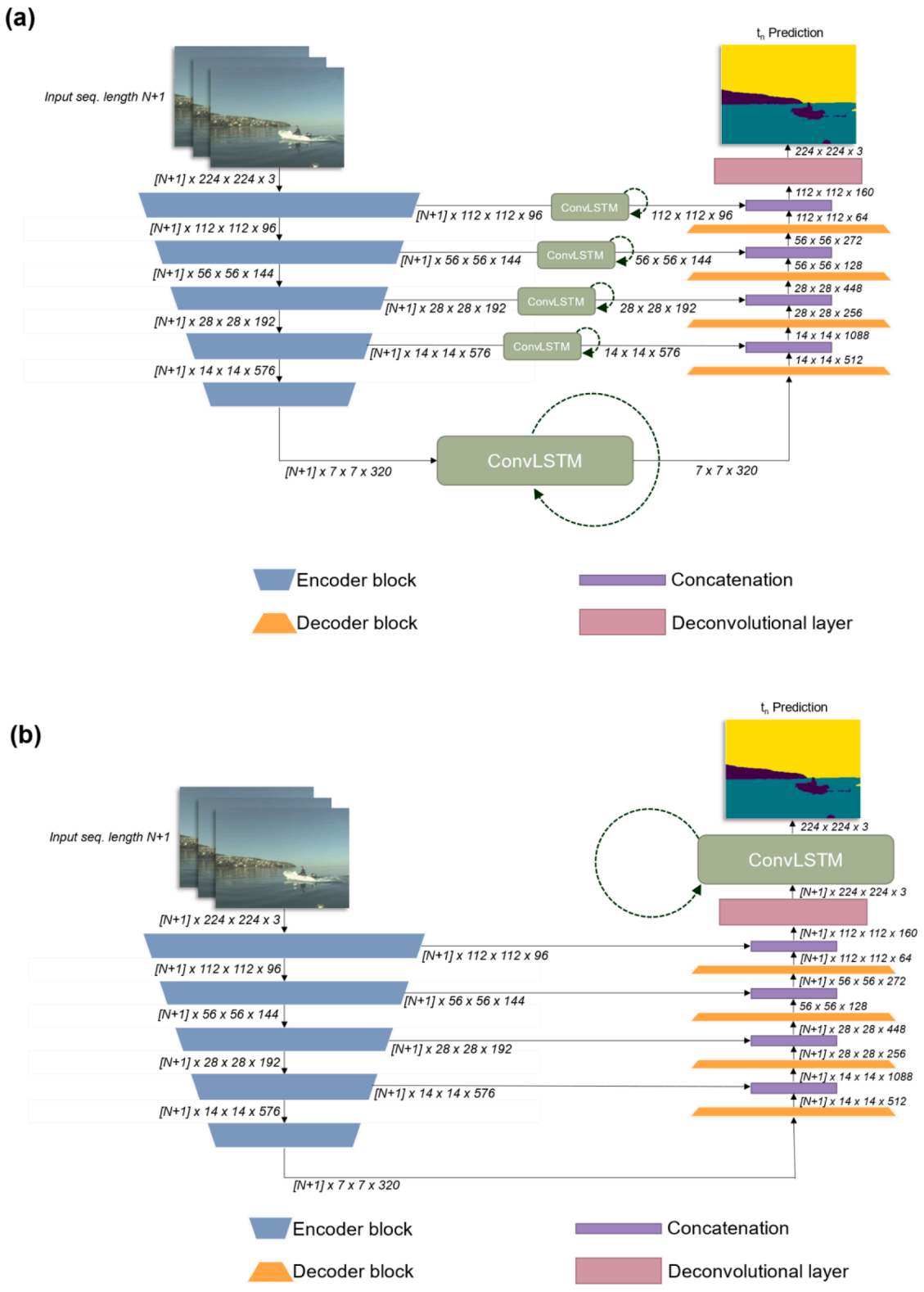

Fig. 10. (a): ShorelineNet-ConvLSTM$_{SKIP}$ architecture using convLSTM blocks in all encoder-decoder connections. (b): ShorelineNet-ConvLSTM$_{END}$ architecture using a single convLSTM block at the end of the network as suggested by Pfeuffer et al. (2019).

spatial information from different levels of the deep network without them being influenced by the ConvLSTM block. This is done as the current frame remains the most important source of information, and spatial analysis of this should not be impeded by all feature maps being impacted by the ConvLSTM block.

Exhaustive model layer details are available in the open-source code[2] used, but the main details which are common between all proposed

[2] https://github.com/KFH22/ShorelineNet

model architectures are outlined below. The Keras "TimeDistributed" layer[3] is used as a wrapper to pass several frames through the encoder such that feature maps from pre-frames are available to the ConvLSTM blocks. The ConvLSTM layers only output feature maps from the last image of the sequence. The encoder, which is a pretrained MobilenetV2 (Sandler et al., 2018) without its fully connected top layer, has outputs defined for skip connections and the deepest layer. The skip connection outputs are taken from the layers "block_1_expand_relu", "block_3_expand_relu", "block_6_expand_relu", and "block_13_expand_-relu". The deepest layer of the encoder is extracted from the mobilenet at 'block_16_project'.

3.1.1. Alternative model architectures

Additional model architectures are trialled to explore the potential for further improvement. Firstly, convLSTM blocks can be incorporated in skip connections as tried by e.g. Rochan (2018). This method drastically increases the network size due to the additional LSTM layers. It allows the model to use temporal context at several levels of the CNN, but also forces all feature maps through ConvLSTM blocks, potentially inhibiting spatial inference in the process. Secondly, a convLSTM block can be positioned as the last component of the network as suggested by Pfeuffer et al. (2019). These two networks are shown in Fig. 10 (a) and (b) respectively and are named ShorelineNet-ConvLSTM$_{SKIP}$ and ShorelineNet-ConvLSTM$_{END}$. In addition, the convLSTM block at the end of the network can be added to the default ShorelineNet-ConvLSTM and ShorelineNet-ConvLSTM$_{SKIP}$ to form unique networks named ShorelineNet-ConvLSTM$_{DEF+END}$ and ShorelineNet-ConvLSTM$_{SKIP+END}$ respectively, where subscript "DEF" refers to the default model architecture.

3.2. Datasets

The ShorelineNet model uses the MaSTr1325 dataset (Bovcon et al., 2019) for training and its performance is evaluated on the MODD2 dataset (Bovcon et al., 2018). Both have since been superseded by the MaSTr1478 (Žust and Kristan 2022b) and MODS (Bovcon et al., 2022) datasets respectively. The suggested models will be trained on predominantly the MaSTr147 dataset and evaluated solely on the MODS set.

3.2.1. Data pipeline

Training, cross-validation, and evaluation data is pre-processed to provide a suitable format for the models in question. The following terminology will be used:

Training dataset: 90% of either the MaStr1325 or MaSTr1478 image dataset with ground truth annotations and preceding frames where necessary.

Validation dataset: Remaining 10% of the training dataset. This will be used during model training for cross validation on unseen data to avoid overfitting.

Pre-frames: Set of preceding frames to each annotated frame in the training/validation dataset. These are either taken from the MaSTr1478 set or created artificially using pitch/yaw/roll transforms.

Mask: Ground-truth annotation of water, sky, obstacle, and unknown regions in an image.

Evaluation dataset: The MODS dataset used for evaluation of model performance on completely unseen image sequences.

3.2.2. Training and cross validation data

The MaSTr datasets are available online.[4] Augmentations are applied to images and pre-frames as suggested by Bovcon et al. (2019) to reduce overfitting risk and increase the transferability to unknown data. Fig. 11 shows the pipeline for training images and pre-frames.

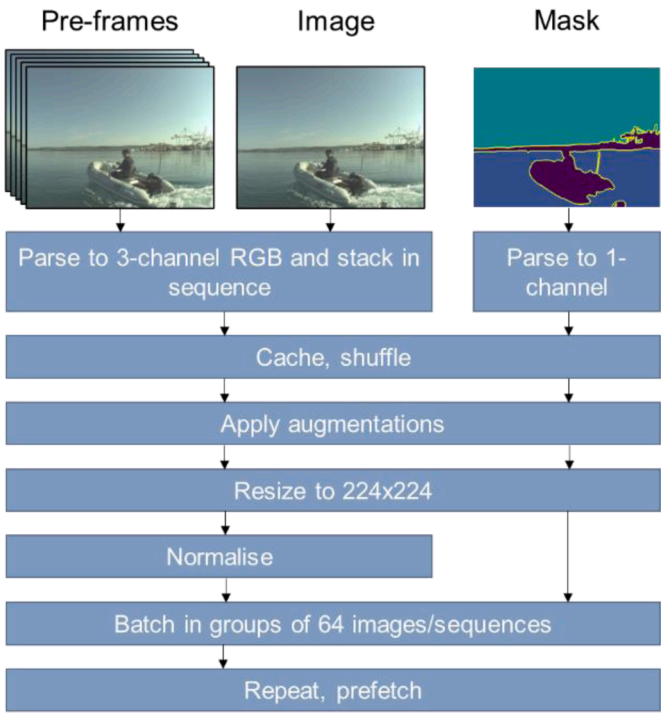

Fig. 11. The data pipeline for training data.

Fig. 12 expands on the specific augmentations applied to the training images. Each augmentation is independently applied to 20% of the input images (50% for the flip augmentation), but augmentations to corresponding pre-frames and masks are consistent. The masks are only augmented geometrically (flipping, rotation, offset and crop).

After augmentations the training images have their pixel-values normalised to values between 0 and 1 and are batched for training. The validation data pipeline is equal to the training data one without any image augmentations.

3.2.3. Evaluation data

The MODS evaluation dataset is available online[5] as 94 sequences of IMU-synchronised stereo videos formatted as individual images. The ShorelineNet-ConvLSTM models take input sequences of $N+1$ length of sequential images and predicts the segmentation masque on the final image in the sequence. Hence, input images are stacked such that each image has N preceding frames with it (Fig. 13). For images where N pre-frames are not available (i.e. the first N frames of any image sequence) copies of the input image are used as substitution.

In summary, a novel network named ShorelineNet-ConvLSTM is proposed to take advantage of sequential input frames from a monocular USV camera. The model adds a single convLSTM block at the deepest layer of the encoder-decoder network and takes an input of N preceding frames along with the current timestep frame. Alternative network architectures are also proposed. The training and evaluation datasets used are the MaSTr1478 and MODS dataset with augmentation applied to the training data as suggested by Bovcon et al. (2019). Cross validation data for measuring training accuracy is an unseen 10% of the training dataset with no augmentations applied.

4. Model training and evaluation

The outlined model architectures trained before evaluating them

[3] https://keras.io/api/layers/recurrent_layers/time_distributed/
[4] https://github.com/lojzezust/WaSR-T
[5] https://vision.fe.uni-lj.si/public/mods/

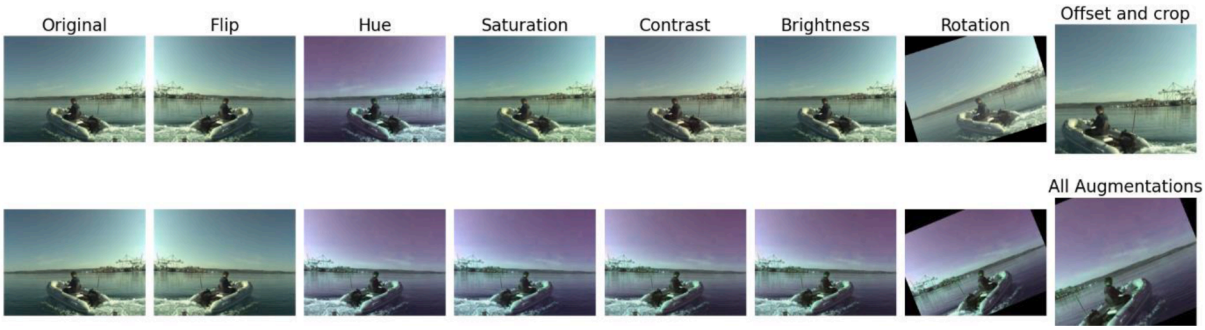

Fig. 12. Augmentations applied to the training dataset. The first row shows the individual augmentations, and the second row shows the cumulative result of augmentations applied sequentially.

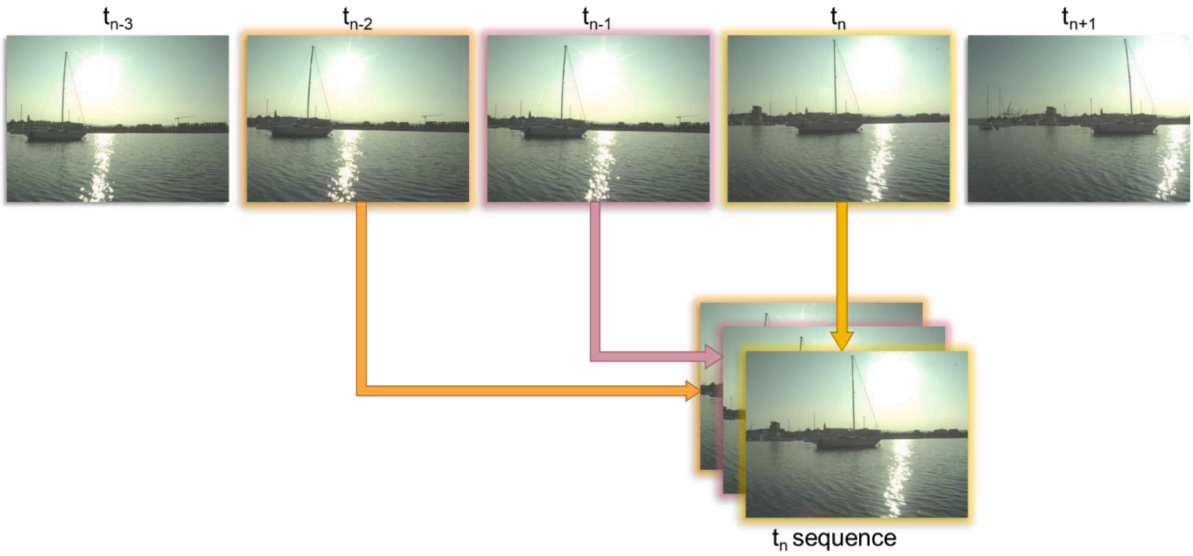

Fig. 13. Evaluation image sequence stacking. Input sequences for the model are generated by taking N preceding frames of the sequence and adding them as pre-frames to the image to be segmented. This process is repeated for every frame in the dataset. In the above example N=2.

using the MODS benchmark (Bovcon et al., 2022). Below, the training and evaluation procedure is outlined.

4.1. Model training

The models were trained on the MaSTr1478 dataset unless other is specified. This is chosen over the MaSTr1325 dataset as the MaSTr1478 contains several scenes with strong reflections (see Fig. 5). In theory, training on more images containing strong reflections should improve model performance in these conditions. Models are generally trained for 200 epochs on data batched into sizes of 64 images/sequences. The raw pixel-wise accuracy on the validation data is taken as the performance criterion, and the best performing model weights after completed training are used for evaluation.

The ShorelineNet model proposed by Yao et al. (2021) is treated as the reference model. The network used is the best performing iteration originally proposed by the authors. The ShorelineNet model is trained manually for 200 epochs to allow like-for like comparison to proposed convLSTM models. However, the best performing model proposed by Yao et al. (2021) was trained for substantially longer (600 epochs). Training for this length is not feasible for all suggested models, and only the main proposed ShorelineNet-ConvLSTM model will be trained for this length for comparison.

The suggested ShorelineNet-ConvLSTM models are trained on the MaSTr1478 dataset as outlined with additional experiments carried out to explore the influence of alternative training settings. These are:

(1) **Training dataset:** The model can be trained on either the MaSTr1325 or extended MaSTr1478 dataset.
(2) **Number of preceding frames:** Number of pre-frames *N* used to provide the model with an image sequence of length *N+1*.
(3) **Pre-frame source:** The preceding frames can either be taken from the MaSTr1478 dataset or artificially constructed using 3D rotation transformations to simulate roll, pitch, and yaw. This method could be used when applying LSTM models to datasets without available pre-frames.
(4) **Training sequence frame rate:** To closer mimic the low frame rate of the evaluation dataset (1fps), training is attempted using only the first pre-frame resulting in an effective training dataset framerate of 2 fps.
(5) **Training time:** The main ShorelineNet-ConvLSTM model is trained for 600 epochs as mentioned for comparison to the best result achieved by Yao et al. (2021).

Exhaustive parameter settings can be found in the code[6] but some general details are given below. The encoder has its weights frozen during training. Both the deconvolutional layers in the decoder and the ConvLSTM layers have dropout of the model weights to avoid overfitting with 10% to 50% of the units dropped depending on the layer in question. The ConvLSTM layers have a kernel size of 3 × 3 (which represent

[6] https://github.com/KFH22/ShorelineNet

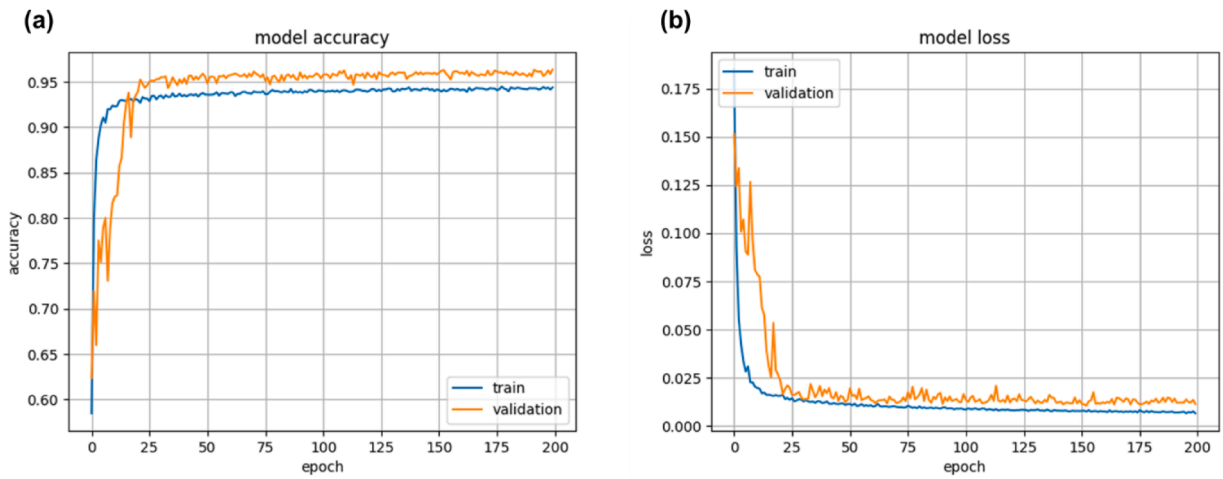

Fig. 14. Learning curves for ShorlineNet-ConvLSTM model. (a): Per-pixel accuracy on augmented training dataset and cross-validation dataset. (b): Custom focal loss on augmented training and cross-validation datasets.

the convolutional window size) and utilise a hyperbolic tan activation function. The complete models are built with a focal cross-entropy loss function (as documented by Yao et al. (2021)) which only calculates the loss in the labelled areas of the ground truth annotations. The optimiser used is a root mean squared propagation with an exponentially decaying learning rate. The initial learning rate is 0.01 and the final rate is 0.0064. During training, the model efficacy is evaluated by its raw accuracy on the validation dataset, and the best model weights are saved continuously. Model tuning was largely based on the work completed by Yao et al. (2021), as one of the goals of this paper is to investigate LSTM efficacy in existing, proven models. However, the decaying learning rate was added as the model quickly reaches an accuracy close to its final value (see Fig. 14). The dropout rate of each ConvLSTM layer is set to the same value as deconvolution layer in the decoder block which the ConvLSTM block in question is connected to.

Generally, all work is be done through Google Colab,[7] wherein a Jupyter[8] notebook file is executed on Google's cloud services. All code is written in Python using Tensorflow[9] and other modules. All training and inference are done using Google Colab Pro+ on a NVIDIA A100 tensor core GPU, with a NVIDIA V100 tensor core GPU used as an alternative when there were no A100s available.

Fig. 14 shows an example of ShorelineNet-ConvLSTM learning curves. These are representative of the alternative model architectures as well, and illustrate how training data augmentation decreases training accuracy. No overfitting is apparent.

Fig. 15 shows examples of segmented images from the validation dataset using the ShorelineNet-ConvLSTM model. These illustrate how the additional images of the MaSTr1478 dataset are challenging to accurately segment.

4.2. Model evaluation

All models are evaluated using the MODS benchmark procedure on the MODS dataset suggested by Bovcon et al. (2022). The evaluation is practically achieved using open-source code from the authors' GitHub page.[10] During evaluation, quantitative measures of water-edge detection accuracy and obstacle detection performance are extracted. In addition to these, figures for qualitative comparison between models are produced from the segmentation masque outputs of each model.

4.2.1. Quantitative performance measures

The accuracy of the water-edge detection is evaluated as the root mean squared error (RMSE) in the vertical direction between the ground truth (GT) water edge and the nearest water edge in the segmentation masque (Eq. (7), where y is the water edge location).

$$RMSE = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\left(y_i - y_{i,\,GT}\right)} \tag{7}$$

The obstacle detection performance is quantified by the model's ability to correctly label pixels within GT obstacle bounding boxes and to avoid mislabelling pixels outside bounding boxes. True positive (TP) detections are obstacles where enough pixels within the bounding box are correctly labelled. If not enough pixels are identified, the obstacle is counted as a false negative (FN). Regions misidentified as obstacle are counted as false positives (FP). Detection accuracy can be expressed concisely by its precision (Pr), recall (Re), and F1-measure Eqs. (8)–((10)).

$$Pr = \frac{TP}{TP + FP} \tag{8}$$

$$Re = \frac{TP}{TP + FN} \tag{9}$$

$$F1 = \frac{2 \cdot Pr \cdot Re}{Pr + Re} \tag{10}$$

A region of 15 m radius around the USV is defined as the "danger zone" where correctly identifying obstacles is particularly important. Measures within this zone are given as additional information.

In summary, all suggested models were trained under equal conditions for comparison of performance. The training was stopped after 200 epochs due to time limitations, but one iteration of the final proposed model was trained for 600 epochs for comparison to the best weights reached by Yao et al. (2021). All model performances were evaluated using the MODS benchmark (Bovcon et al., 2022) wherein water-edge and obstacle detection accuracy is extracted.

5. Results and discussion

Evaluation results are outlined and discussed below. The primary measure of model performance is F1-score both globally and in the "danger zone". Additionally, the number of trainable parameters and inference times are documented. Generally, numbers in brackets are measures within the danger zone. For qualitative comparisons, modified

[7] https://colab.research.google.com/
[8] https://jupyter.org/
[9] https://www.tensorflow.org/
[10] https://github.com/bborja/mods_evaluation

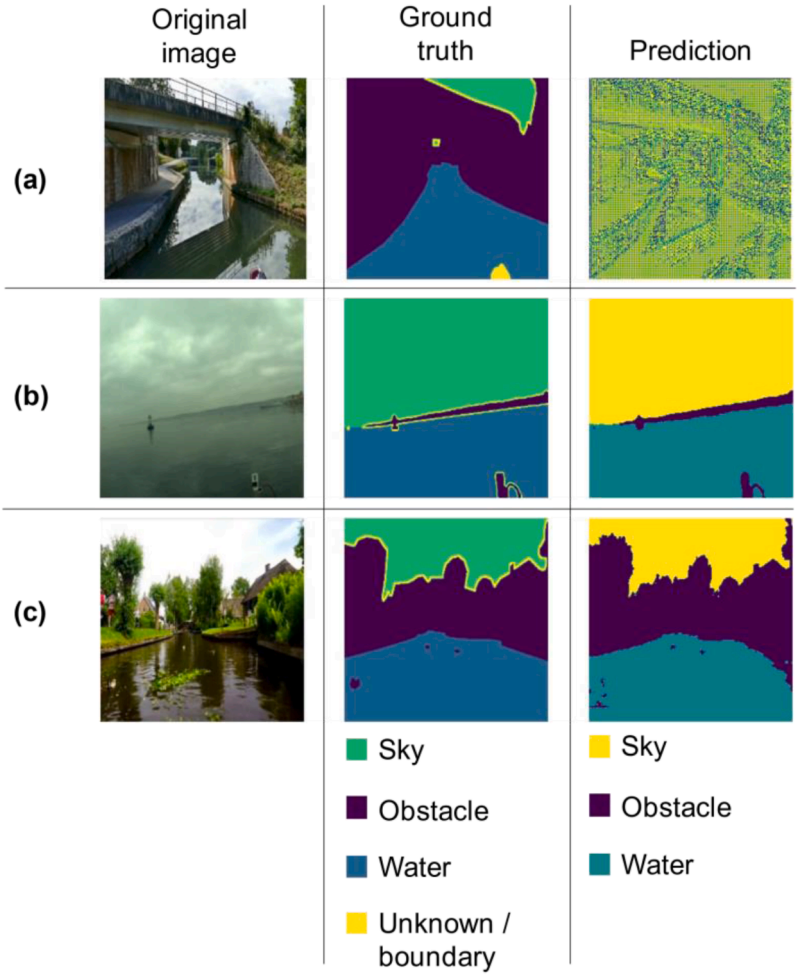

Fig. 15. Examples of segmentation using ShorelineNet-ConvLSTM (a): before training of the model (b): image from MaSTr1325 dataset after training with high segmentation accuracy. (c): image from extended MaSTr1478 dataset after training illustrating worse performance in challenging environment.

Table 3
Network architecture performance comparison.

Model	TP	FP	FN	F1
ShorelineNet baseline	45,766	19,342	6722	77.8%
ShorelineNet-ConvLSTM	40,880	**8590**	11,608	80.2%
ShorelineNet-ConvLSTM$_{SKIP}$	44,203	13,856	8285	80.0%
ShorelineNet-ConvLSTM$_{END}$	43,292	14,443	9196	78.6%
ShorelineNet-ConvLSTM$_{DEF+END}$	41,228	10,759	11,260	78.9%
ShorelineNet-ConvLSTM$_{SKIP+END}$	**46,688**	15,439	**5800**	**81.5%**

Table 4
Network architecture performance comparison in danger zone.

Model	TP	FP	FN	F1
ShorelineNet baseline	2900	4126	338	56.5%
ShorelineNet-ConvLSTM	2774	2789	464	63%
ShorelineNet-ConvLSTM$_{SKIP}$	2813	**1490**	425	**74.6%**
ShorelineNet-ConvLSTM$_{END}$	2825	3571	413	58.6%
ShorelineNet-ConvLSTM$_{DEF+END}$	2764	2186	474	67.5%
ShorelineNet-ConvLSTM$_{SKIP+END}$	**3005**	4547	**233**	55.7%

code from Bovcon et al. (2022)[11] was used to generate figures. Unless otherwise specified 1 pre-frame is used (N=1), and training is done on the MaSTr1478 dataset.

5.1. Performance of different network architectures

Table 3 shows the MODS benchmark scores for ShorelineNet and the 5 convLSTM models proposed. All 5 convLSTM models outperform the baseline to different degrees, the best performing being ShorelineNet-ConvLSTM$_{\mathbf{SKIP+END}}$. However, the most substantial reduction in false positive detections is obtained when using a single convLSTM block at the deepest level of the network (ShorelineNet-ConvLSTM). This comes at a cost to TP and FN detections though. Naturally, any decrease in TP detections results in an equivalent rise in FN detections as the two always sum to the total number of obstacles annotated in the MODS set.

Table 4 shows the obstacle detection performance in the danger zone. Here, the otherwise best performing model (ShorelineNet-ConvLSTM$_{\mathbf{SKIP+END}}$) has a substantial increase in FP detections. All other convLSTM models outperform the baseline in F1-score, the best model being ShorelineNet-ConvLSTM$_{SKIP}$ due to its low FP rate.

The results from Tables 3 and 4 can be interpreted by TP, FP, and FN detections to give detailed insight into the effect of the LSTM layers. Firstly, the baseline outperforms all suggested architectures but ShorelineNet-ConvLSTM$_{\mathbf{SKIP+END}}$ both globally and within the danger zone when it comes to TP and FN detections. On the contrary, all suggested models surpass the effectiveness of the baseline model in global FP detection rate, with the ShorelineNet-ConvLSTM$_{\mathbf{SKIP+END}}$ being the worst of the ConvLSTM models. The ShorelineNet-ConvLSTM$_{\mathbf{SKIP+END}}$ model actually has more FP detections within the danger zone than the

[11] Available here: https://github.com/bborja/mods_evaluation

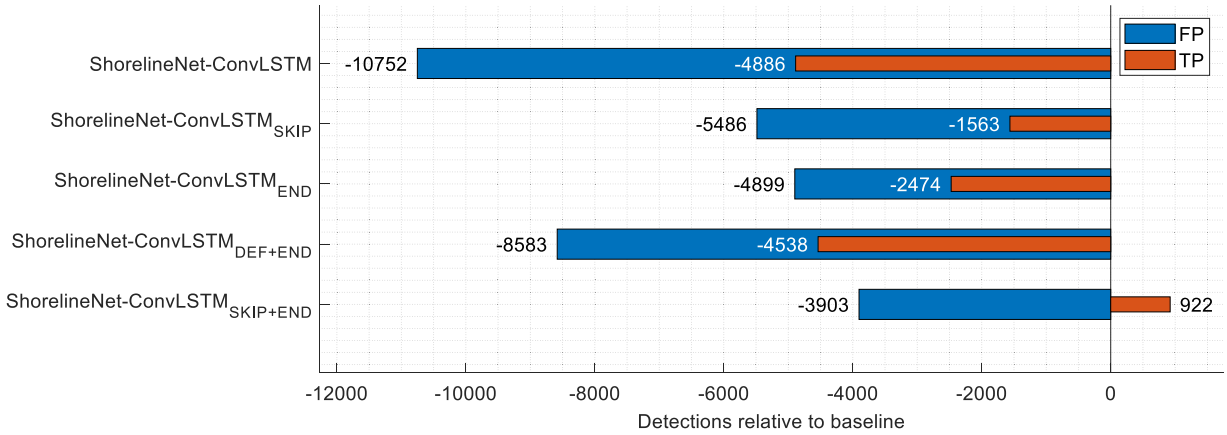

Fig. 16. TP and FP detections relative to baseline ShorelineNet. All ConvLSTM models apart from ShorelineNet-ConvLSTM$_{SKIP+END}$ show a reduction in both FP and TP detections. The decrease in FP detections is consistently larger than the decrease in TP detections. ShorelineNet-ConvLSTM$_{SKIP+END}$ has both an increase in TP detections and a decrease in FP detections, but the latter is the smallest decrease of all the proposed architectures.

Table 5
Water-edge detection accuracy comparison for different model architectures.

qModel	RMSE (pixels)
ShorelineNet baseline	38
ShorelineNet-ConvLSTM	37
ShorelineNet-ConvLSTM$_{SKIP}$	**26**
ShorelineNet-ConvLSTM$_{END}$	32
ShorelineNet-ConvLSTM$_{DEF+END}$	32
ShorelineNet-ConvLSTM$_{SKIP+END}$	28

baseline model. Together, these observations indicate that the addition of LSTM layers generally makes the models predict less total obstacle instances. This behaviour is shown in Fig. 16, where the TP and FP detection rate of the suggested models, relative to that of the baseline model, are compared. Here, it is seen this the reduction in FP detections is larger than the reduction in TP detections for ConvLSTM models, demonstrating how the reduction in obstacle predictions in ConvLSTM models predominantly happens in places where an FP detection happened and less in areas of actual obstacles.

As mentioned, an outlier is ShorelineNet-ConvLSTM$_{\mathbf{SKIP+END}}$ which has better TP detection rate than the baseline model and the lowest reduction in FP detections of all the ConvLSTM models. In addition, this model architecture has more FP detections than the baseline in the danger zone. As ShorelineNet-ConvLSTM$_{\mathbf{SKIP+END}}$ has the most ConvLSTM layers (see Fig. 10) it is likely that this behaviour is due to the network relying a lot on the temporal dimension and its "memory" compared to the spatial observations made frame by frame. This effect is echoed in later investigations (see Section 5.2) and demonstrates the importance of allowing models to take temporal information from earlier frames into account without it taking precedence over the spatial information in the current frame.

Table 5 shows the RMSE of the water edge detection (in pixels) of the 6 models. The baseline model is again outperformed by all convLSTM models, and the best water-edge detection accuracy is reached by ShorelineNet-ConvLSTM$_{\mathbf{SKIP}}$. Although the water-edge detection accuracy is important for shoreline detection, the primary performance measure remains the obstacle detection accuracy as this, and FP detection rate in particular, is the main area of interest for the application of LSTM in this context.

When comparing model performance, sheer network size is generally a benefit, but it comes at a cost to computational load. Table 6 shows the number of trainable parameters and GPU inference time in the 6 models.

To quantify model performance relative to number of trainable parameters, the ratio of increase in F1-score and network size relative to ShorelineNet is found (Eq. (11)).

$$\text{F1 to network size} = \frac{F1 - F1_{baseline}}{\left(\frac{Parameters}{Parameters_{baseline}}\right)} \tag{11}$$

Table 6
Comparison of networks in terms of trainable parameters, inference time, and inference frames per second (FPS). This test was carried out on a NVIDIA V100 Tensor Core GPU.

Model	Trainable parameters	Inference time	FPS
ShorelineNet baseline	**4.66M**	**32.5 ms**	**30.8**
ShorelineNet-ConvLSTM	12.03M	38.4 ms	26.0
ShorelineNet-ConvLSTM$_{SKIP}$	40.74M	46.5 ms	21.5
ShorelineNet-ConvLSTM$_{END}$	4.66M	37.1 ms	27.0
ShorelineNet-ConvLSTM$_{DEF+END}$	12.04M	40.1 ms	24.3
ShorelineNet-ConvLSTM$_{SKIP+END}$	40.74M	48.0 ms	20.8

Fig. 17 shows this measure indicating that ShorelineNet-ConvLSTM achieves the best increase in global F1-score for the increase in number of parameters. In the danger zone, the ShorelineNet-ConvLSTM$_{DEF+END}$ model is most efficient relative to the network size, but globally the F1-score of this model is low relative to ShorelineNet-ConvLSTM.

The results shown in Tables 3 and 4 indicate that ShorelineNet-ConvLSTM$_{SKIP}$ is the best overall model as F1-score in the danger zone is prioritised higher than the global score due to the importance of obstacle detection in the vicinity where a collision may be imminent. However, this model has significantly more parameters and longer inference time than ShorelineNet-ConvLSTM. Therefore, ShorelineNet-ConvLSTM, with only one convSLTM at the deepest layer, is generally preferred, as its performance relative to network size is the best of the suggested architectures. Furthermore, it also shows a good reduction of false positive detections relative to the baseline (demonstrated qualitatively in Fig. 18) and when compared to the alternative architectures (demonstrated qualitatively in Fig. 19).

The investigations into different ConvLSTM model architectures have demonstrated that the addition of ConvLSTM layers generally reduces the FP detections made by the model, but also reduces the total obstacle detection rate in with a negative impact on TP and FN detections. However, the improvements in FP detection are enough to improve the overall performance for all ConvLSTM models as reflected in the F1-score. For a balanced performance at a small increase in network size, ShorelineNet-ConvLSTM is preferred and will be used in the remaining investigations of this paper, but other models may be preferred under certain circumstances. For instance, if the USV is not

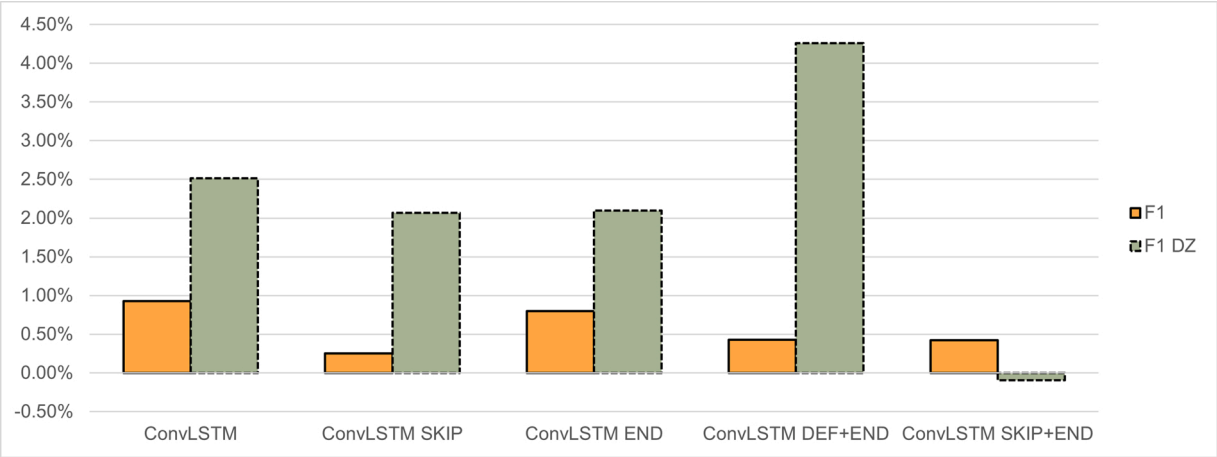

Fig. 17. Comparison of additional F1-score of models per multiple of the baseline model's parameters.

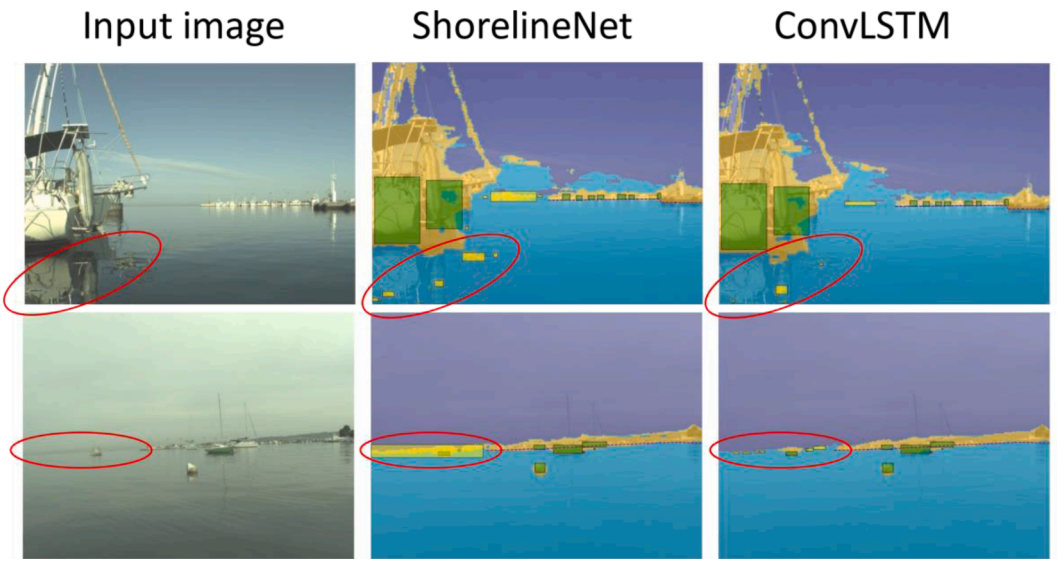

Fig. 18. Qualitative comparison of ShorelineNet and ShorelineNet-ConvLSTM. Yellow rectangles indicate FP detections - areas where the model detects an obstacle although there is none. The top row of images shows performance on an image with a strong reflection (marked in the red ellipse). It can be seen that the ConvLSTM model has significantly less FP detections in this area. The bottom row shows a comparison of performance in an image with an unclear horizon (red ellipse). Here, the ConvLSTM model better disregards the hazy horizon line as a non-obstacle as seen from reduced FP detections.

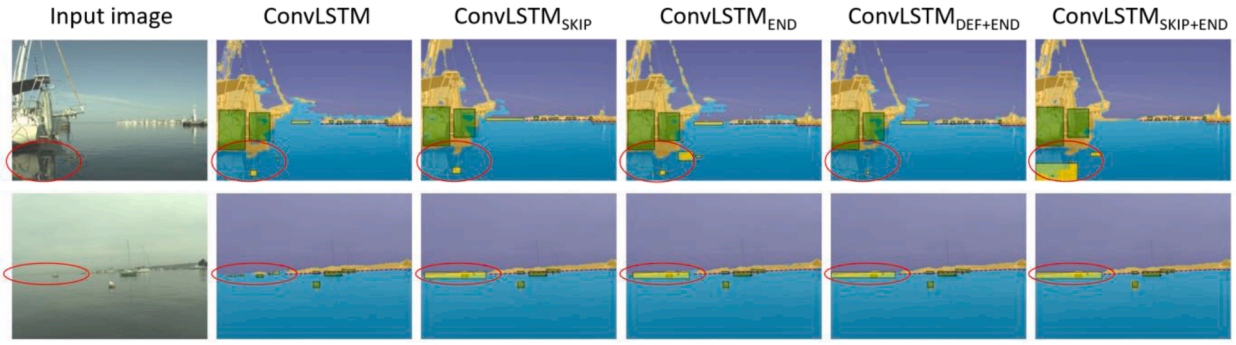

Fig. 19. Qualitative comparison of proposed model architectures. Each row shows a comparison of alternative ConvLSTM model architectures' performance on a given frame. A region of interest (red ellipse) is given in each row, and it can be seen that the default ConvLSTM model architecture generally has the least FP detections (yellow rectangles) when compared to alternative architectures.

Table 7
Performance comparison of ShorelineNet-ConvLSTM model using different number of pre-frames, *N*.

N	RMSE (pixels)	TP	FP	FN	F1
1	37	40,880 (2774)	**8590** (2789)	11,608 (464)	**80.2%** (56.5%)
2	33	42,794 (2821)	12,628 (4105)	9694 (417)	79.3% (55.5%)
3	32	43,209 (2865)	13,682 (3536)	9279 (373)	79.0% (59.4%)
4	**28**	43,312 (2827)	12,834 (2428)	9176 (411)	79.7% (66.6%)
5	37	**46,582** (2962)	24,260 (9688)	**5906** (276)	75.5% (37.3%)

limited by hardware, for example by having a powerful GPU installed, the larger network ShorelineNet-ConvLSTM$_{SKIP}$ is preferred as its obstacle detection in the danger zone is significantly more robust than all other tested models at only a small cost to global F1-score. In general, adding a ConvLSTM block at the end of the network as suggested by Pfeuffer et al. (2019) showed the smallest benefit and is not preferred. Only when combined with ConvLSTM blocks in the skip connections a significant improvement was found, but the increase in FP detections compared to ShorelineNet-ConvLSTM$_{SKIP}$ is too large to warrant the change, as especially the danger zone performance is inhibited by this.

5.2. Effect of number of preceding frames

Table 7 shows results of using different numbers of pre-frames for temporal context. Using more frames improves the model in terms of water edge, TP, and FN detections, but results in more FPs. This is especially clear when N=5, wherein the number of FP detections severely inhibits overall performance. This large number of FP detections is likely also the cause of a worse RMSE score due to FP detections in the water-edge region.

As shown, the number of FP detections rises substantially with the number of pre-frames used. This may be a result of the low frame rate of videos in the evaluation dataset (1 fps) (Bovcon et al., 2022) since obstacle locations may retained unintendedly in the cell state. Fig. 20 illustrates this issue, where the difference between obstacle location in sequential frames is substantial.

Fig. 21 shows qualitative comparisons of using 1 and 5 pre-frames. A large number of FP detections are present when N=5, likely from retention of obstacle locations in the cell state.

In conclusion, the best obstacle detection performance was in this case reached by only using one pre-frame (N=1), as the model could here account for the temporal information without losing the frame-by-frame spatial capability. When the number of pre-frames is increased, and especially at the highest number (N=5), the FP detections rise along with the TP detections resulting in behaviour similar to that exhibited by ShorelineNet-ConvLSTM$_{SKIP+END}$ in Section 5.1. This result suggests that, in both cases, the temporal information is valued too highly resulting in model "sluggishness", as it retains pre-frame information in its predictions instead of reacting strongly to data contained in the current frame. It should be noted that when using higher frame rate videos, it might be beneficial to increase the number of pre-frames above 1.

5.3. Effect of artificially created pre-frames

Table 8 shows the results of using an artificially made pre-frame created by applying 3D rotations to the current frame. The performance deficit in FP detections is too large to warrant further investigation.

Experimentation with artificially created pre-frames is motivated by the logistical challenges of dataset acquisition and annotation. For

Table 8
ShorelineNet-ConvLSTM results using pre-frames from the MaSTr1325 dataset, and artificially made pre-frames rotated up to 2 ° about each axis.

Preframe source	RMSE (pixels)	TP	FP	FN	F1
MaSTr1478	37	40,880 (2774)	**8590** (2789)	11,608 (464)	**80.2%** (56.5%)
Artificial	**34**	**42,726** (2859)	14,653 (4305)	**9762** (379)	77.8% (55.0%)

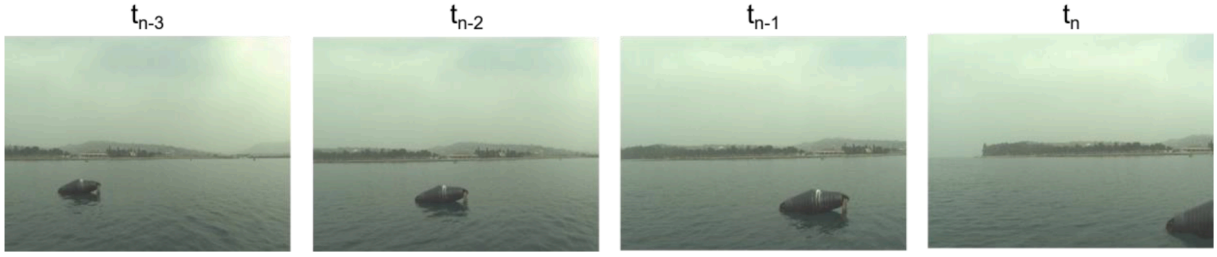

Fig. 20. The difference between subsequent annotated frames in the MODS evaluation dataset.

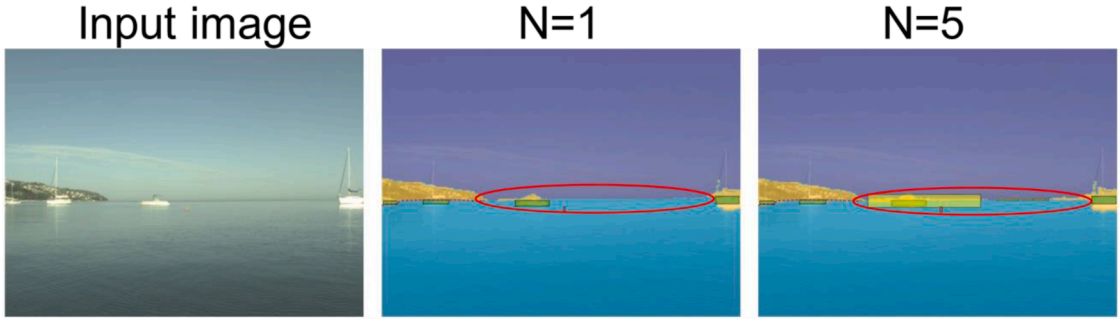

Fig. 21. A substantial increase in FP detections (yellow boxes) decrease both the F1-score and water-edge detection RMSE of the 5-preframe experiment.

Table 9
ShorelineNet-ConvLSTM results with a single pre-frame. The frame rate of the training sequence is lowered by taking the first of the 5 available pre-frames instead of the last one.

Training sequence fps	RMSE (pixels)	TP	FP	FN	F1
10	37	40,880 (2774)	**8590** (2789)	11,608 (464)	80.2% (56.5%)
2	**24**	**44,527** (2993)	9984 (4583)	**7961** (245)	**83.2%** (55.4%)

instance, the MaSTr1325 dataset (Bovcon et al., 2019) took 24 months to gather to ensure variability in seasons, locations, time of day, and conditions. However, it is still gathered on a single USV in a limited area in the Slovenian gulf of Koper. In addition, per-pixel annotation is currently a requirement for effective training of semantic segmentation models substantial time is required for annotation by human annotators, approximately 20 minutes per image in the case of MaSTr1325. Together, they limit the important work of acquiring and annotating new datasets, hence additional studies into semi-supervised training or artificially created pre-frames should be done.

5.4. Effect of lower frame rate of training dataset

Table 9 shows the results of training on an image sequence recorded at 10 fps and 2 fps. A substantial increase in performance is seen, highlighting the sensitivity of convLSTM models to having similar frame rates in training and application.

Fig. 22 shows a qualitative comparison of using a lower frame rate training dataset. Especially small, far-away obstacles are better identified when training on a low frame rate set, indicating that this model does not rely on the convLSTM cell "remembering" the location of obstacles.

5.5. Training dataset effect on results

Tables 10 and 11 show the effect of training on different datasets. The baseline model performs best when trained on the MaSTr1325 dataset and fails to translate the 153 additional images of the MaSTr1478 set into better performance. For the ShorelineNet-ConvLSTM model the opposite case is clear, as training on the MaSTr1478 set brings a substantial reduction in FP detections, albeit with a cost to FN and TP detections.

Qualitatively, the additional images in the MaSTr1478 dataset are substantially different to the original 1325 images, and the number of FP detections by the convLSTM model trained on MaSTr1325 could therefore indicate that some degree of overfitting is being prevented by the variation of the extra images. However, the additional 153 images also differ substantially from the evaluation images in the MODS dataset which may explain why ShorelineNet performs better when trained on only the MaSTr1325 set. This result underlines the need for larger datasets with substantial variation in the environment, and with the presence of more reflections, as these are evidently needed for robust FP avoidance in LSTM networks.

Table 10
ShorelineNet performance using different training datasets.

Training dataset	RMSE (pixels)	TP	FP	FN	F1
MaSTr1478	38	**45,766** (2900)	19,342 (4126)	**6722** (338)	77.8% (56.5%)
MaSTr1325	**33**	44,693 (2874)	**16,709** (3641)	7795 (364)	**78.5%** (58.9%)

Table 11
ShorelineNet-ConvLSTM performance using different training datasets.

Training dataset	RMSE (pixels)	TP	FP	FN	F1
MaSTr1478	**37**	40,880 (2774)	**8590** (2789)	11,608 (464)	**80.2%** (56.5%)
MaSTr1325	38	**46,890** (3025)	24,399 (12,180)	**5598** (213)	75.8% (32.8%)

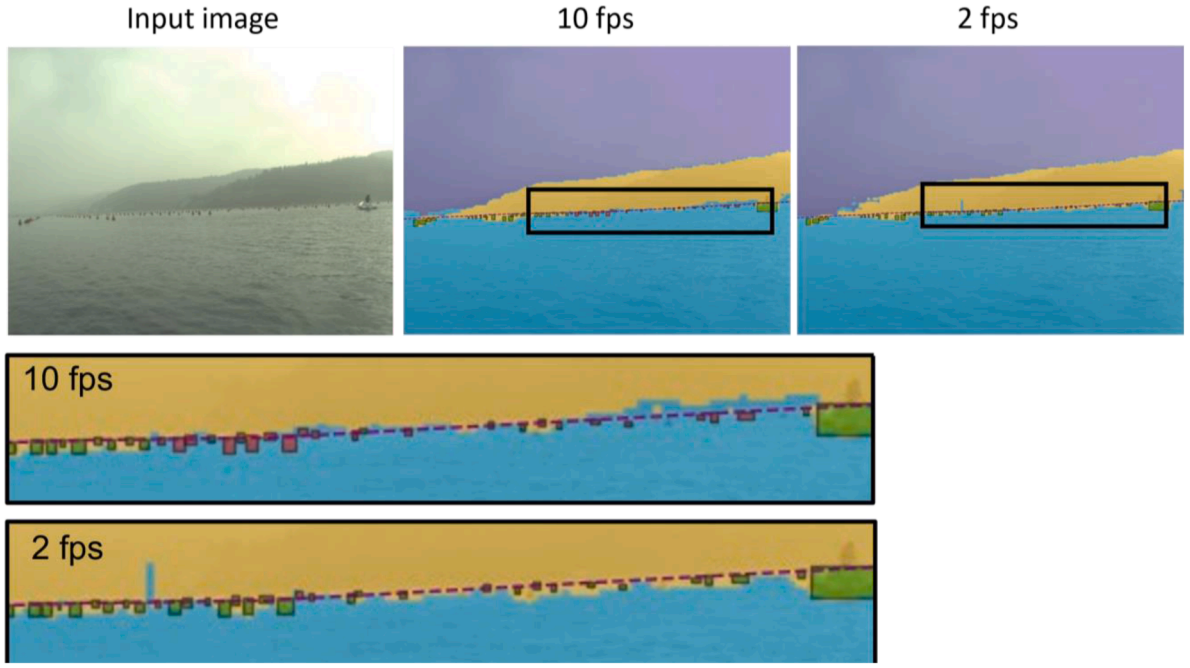

Fig. 22. Qualitative comparison of evaluation of models using 10 fps and 2 fps training sequences. The zoomed-in regions show many FN detections (red squares) for the high frame rate model.

Table 12
Results using models training for 600 epochs with lower learning rate. Numbers in parentheses are values within the "danger zone".

Model	RMSE (pixels)	TP	FP	FN	F1
ShorelineNet Baseline	30	44,255 (3019)	**5570 (2350)**	8233 (219)	**86.5% (70.2%)**
ShorelineNet-ConvLSTM	24	**45,680 (3037)**	10,825 (3885)	**6808 (201)**	83.8% (59.8%)

5.6. Effect of longer training

The best performing ShorelineNet model was trained by Yao et al. (2021) for longer to see if better accuracy could be reached. Specifically, it was trained 600 epochs with a lower learning rate – an approach which has been reproduced with ShorelineNet-ConvLSTM. The results in Table 12 show that the baseline ShorelineNet model performs better overall and in FP rate.

Training the ShorelineNet-ConvLSTM model for 600 epochs results in better TP and FN results but a degraded FP performance (Fig. 23). This may result from the convLSTM model attributing too much value to obstacle locations in the cell state, and effectively overfitting to this with more training. This effect could be similar to that seen in the frame rate experiment. This result cements the importance of training set similarity to the wanted application data when it comes to frame rate.

5.7. Comparison to state-of-the-art

A main contribution of Bovcon et al. (2022) is the benchmarking of existing models using their suggested procedure. Below, Table 13 shows selected models compared to ones suggested in this report for context to state-of-the-art methods. TP and FP numbers are given in rates per 100 images (TPr and FPr).

When compared to state-of-the-art, the ShorelineNet (including convLSTM) models are relatively lightweight and reach average inference accuracy. All proposed models achieve inference rates over 26 fps on the given hardware, whereas Bovcon et al. (2022) reached inference rates of 15.6 fps for WaSR, 8.3 fps for DeepLab panoptic, 26.0 fps for RefineNet, and 56.5 fps for BiSeNet using hardware of similar theoretical performance. In addition to inference rate, short training time is desired for developmental efficiency. Training the models in this project for 600 epochs took several hours on very powerful hardware, and the

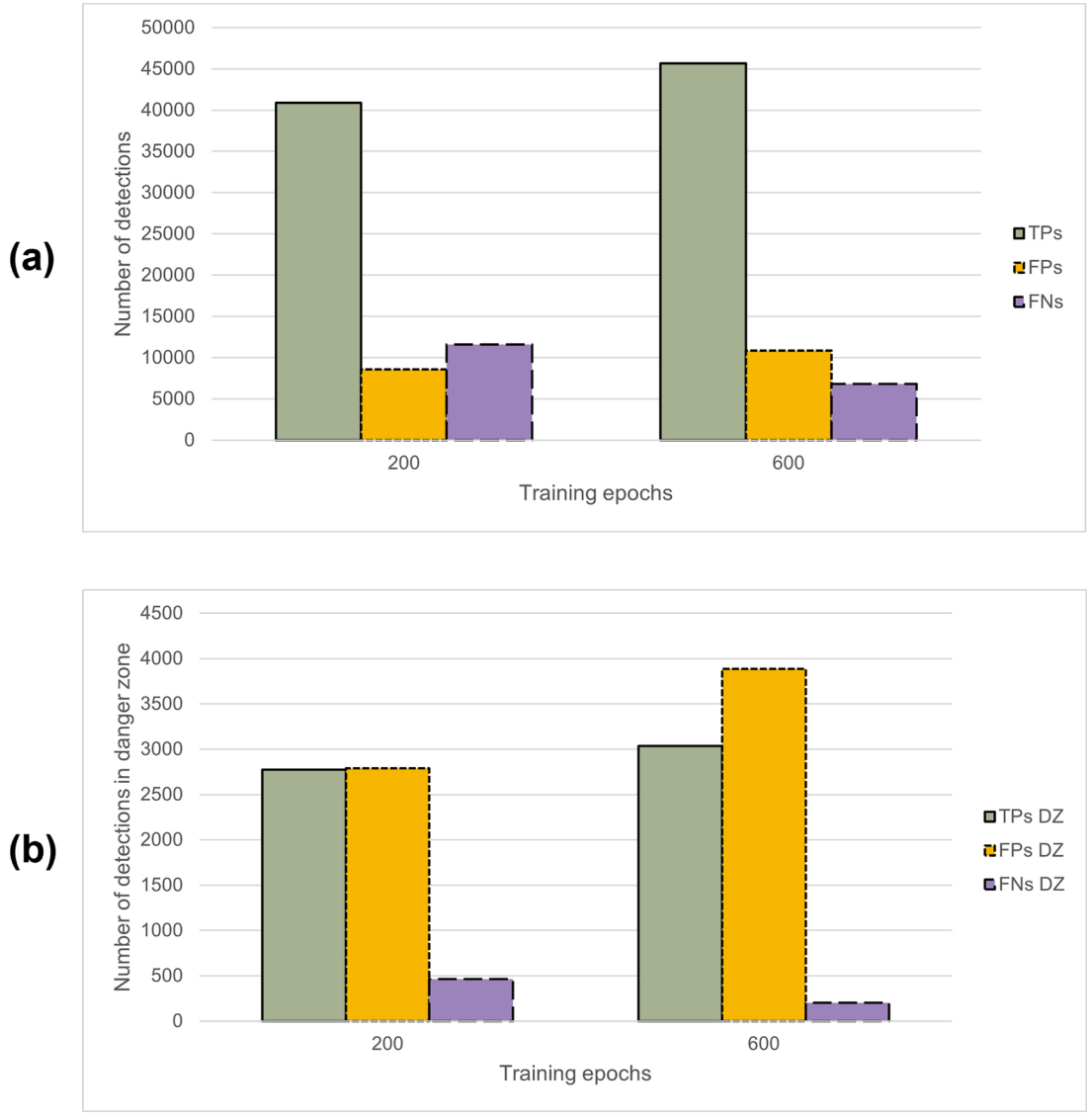

Fig. 23. Effect on true positive, false positive, and false negative detections of length of training. (a): Global obstacle detection performance. (b): Danger zone obstacle detection performance.

Table 13
Comparison of performance on the MODS benchmark to selected models. Underlined rows are models evaluated as part of this report, the rest were done by Bovcon et al. (2022).

Model	Trainable parameters	RMSE (pixels)	TPr	FPr	F1
ISSM (Bovcon et al., 2018)	-	181	55.3	44.6	67.1%
ENet (Paszke et al., 2016)	**0.4M**	78	62.6	42.0	73.8%
ShorelineNet 200 epochs training (Yao et al., 2021)	4.66M	37	56.6	23.9	77.8%
PSPNet (Zhao et al., 2016)	56.0M	21	59.4	26.2	78.9%
ShorelineNet-ConvLSTM	12.03M	37	50.6	10.6	80.2%
MobileUNet (Howard et al., 2017)	8.9M	35	54.8	14.5	81.6%
ShorelineNet-ConvLSTM 2 fps training set	12.03M	24	55.1	12.4	83.2%
SegNet (Badrinarayanan et al., 2017)	35.0M	23	57.7	15.0	83.8%
DeepLab3+ (Chen et al., 2018)	48.0M	21	60.2	15.1	85.9%
ShorelineNet 600 epochs training (Yao et al., 2021)	4.66M	30	54.8	6.9	86.5%
BiSeNet (Yu et al., 2018)	47.5M	**17**	58.4	6.1	90.3%
RefineNet (Lin et al., 2017)	85.7M	18	**60.4**	7.4	91.0%
DeepLab panoptic (Cheng et al., 2020)	46.7M	**17**	59.9	6.6	91.2%
WaSR (Bovcon and Kristan 2022)	84.6M	21	56.8	**2.6**	**91.4%**

implementation of convLSTM cells clearly allowed much better results after shorter training periods.

6. Conclusions

This paper is an investigation into the use of LSTM cells in CNNs to improve CV model performance in marine environments. A network structure, ShorelineNet-ConvLSTM, is proposed using convolutional LSTM cells to perform semantic segmentation of marine environment images into regions of sky, water, and obstacles. The network utilises temporal context in the form of sequential video frames, a resource which has rarely been utilised in USV CV. This network was shown to improve performance over a baseline model, in particular by reducing false positive detections of environmental features like reflections.

Auxiliary experiments

In addition to the proposed network, 4 additional CNN structures were trialled with varying success. In general, adding convolutional LSTM elements in encoder-decoder skip connections leads to improved segmentation accuracy, but at a disproportional increase to network parameters. Furthermore, experiments with different training datasets, number of preceding frames, etc. were done. In general, these showed that convLSTM cells are beneficial when using a single preceding context frame recorded at a frame rate as close as possible to that of the evaluation dataset.

Limitations

The proposed network showed improved performance when training time was limited, but this improvement is diminished when training time is increased substantially. When increasing training iterations from 200 to 600 epochs on the dataset, the suggested model has an increase in F1-score from 80.2% to 83.8% against the baseline model's increase from 77.8% to 86.5%. The primary inhibitor of ShorelineNet-ConvLSTM when trained for 600 epochs is that it overfits to the temporal context frames thereby experiencing an increase in FP detections. Using the training dataset in this paper, the optimal training time for FP reduction using ShorelineNet-ConvLSTM likely lies between 200 and 600 epochs, but with a lower fps training dataset it is possible that this could be increased to improve performance while avoiding overfitting. Similarly, providing more context frames or having too high frame rate of the training dataset leads to the convLSTM cell state storing obstacle location data, which inhibits the model when this changes a lot between frames. This is also likely to be one of the causes of diminishing model efficiency with additional training, as the network overfits to rates of change in obstacle location.

Future work

Areas of suggested further research are outlined below. Firstly, the frame rate dependency of convLSTM networks should be further investigated by training and evaluating on image sequences recorded at the same frame rate and USV velocity. This could be done by using un-annotated images from the MODS dataset[12] for evaluation sequence pre-frames. Secondly, to increase variety and extent of the training dataset, experiments training on frames only annotated with obstacle bounding boxes and water-edges should be attempted, as in the work of Žust and Kristan (2022a). This could potentially improve LSTM network flexibility as this trend was seen between training on the MaSTr1325 and MaSTr1478 datasets.

When it comes to network architectures several approaches are suggested for further experiments. Firstly, adding convLSTM blocks in other well-performing encoder-decoder networks could expand on the experiences made in this project. Secondly, using other LSTM cell structures such as those with peephole connections (Gers et al., 2002) or GRU cells (Chung et al., 2014) could lead to improved results. Thirdly, using LSTM cells on stereo-images instead of time-sequence frames could be an alternative application, in a similar way to the application of LSTM networks in the medical field, e.g. no MRI "slices" (Xu et al., 2019). Finally, using LSTM cells in network architectures specifically designed for these is likely to result in improved performance. For instance, "branching" models, which capture spatial and temporal information in separate branches before combining these have been applied in prediction of future video frames (Fan et al., 2019). Similar structures could present an even better application for LSTM cells in marine environment CV applications.

CRediT authorship contribution statement

Kasper Foss Hansen: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing. **Linghong Yao:** Conceptualization, Supervision, Writing – original draft. **Kang Ren:** Methodology, Writing – original draft. **Sen Wang:** Visualization, Methodology, Writing – original draft. **Wenwen Liu:** Writing – original draft, Writing – review & editing, Funding acquisition. **Yuanchang Liu:** Methodology, Investigation, Supervision, Writing – original draft, Writing – review & editing, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

[12] Un-annotated MODS frames are available here: https://vision.fe.uni-lj.si/public/

Acknowledgment

This work is partially funded by the Research Grant, The Royal Society (RGS\R2\212343) and Nanjing Science and Technology Innovation Project (R2022LZ01).

References

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (12), 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615.

Bovcon, B., Kristan, M., 2022. WaSR—a water segmentation and refinement maritime obstacle detection network. IEEE Trans. Cybern. 52 (12), 12661–12674. https://doi.org/10.1109/TCYB.2021.3085856.

Bovcon, B., Mandeljc, R., Perš, J., Kristan, M., 2017. Improving vision-based obstacle detection on USV using inertial sensor. In: Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis. Ljubljana, Slovenia. https://doi.org/10.1109/ISPA.2017.8073559, 18-20 Sept. 2017.

Bovcon, B., Mandeljc, R., Perš, J., Kristan, M., 2018. Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation. Robot. Auton. Syst. 104, 1–13. https://doi.org/10.1016/j.robot.2018.02.017.

Bovcon, B., Muhovič, J., Perš, J., Kristan, M., 2019. The MaSTr1325 dataset for training deep USV obstacle detection models. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Macau, China. https://doi.org/10.1109/IROS40897.2019.8967909, 3-8 Nov. 2019.

Bovcon, B., Muhovič, J., Vranac, D., Mozetič, D., Perš, J., Kristan, M., 2022. MODS—a USV-oriented object detection and obstacle segmentation benchmark. IEEE Trans. Intell. Transp. Syst. 23 (8), 13403–13418. https://doi.org/10.1109/TITS.2021.3124192.

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference On Computer Vision (ECCV). https://doi.org/10.48550/arXiv.1802.02611.

Chen, X., Liu, Y., Achuthan, K., 2021. WODIS: water obstacle detection network based on image segmentation for autonomous surface vehicles in maritime environments. IEEE Trans. Instrum. Meas. 70, 1–13. https://doi.org/10.1109/TIM.2021.3092070.

Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C., 2020. Panoptic-deeplab: a simple, strong, and fast baseline for bottom-up panoptic segmentation. In: Proceedings of the IEEE/CVF Conference On Computer Vision And Pattern Recognition. https://doi.org/10.48550/arXiv.1911.10194.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv preprint. https://doi.org/10.48550/arXiv.1412.3555.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition. https://doi.org/10.48550/arXiv.1604.01685.

Deng, J., Dong, W., Socher, R., Li, L.J., Kai, L., Li, F.F., 2009. ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Miami, FL, USA. https://doi.org/10.1109/CVPR.2009.5206848, 20-25 June 2009.

Fan, H., Zhu, L., Yang, Y., 2019. Cubic LSTMs for video prediction. Proc. AAAI Conf. Artif. Intell. 33 (01), 8263–8270. https://doi.org/10.1609/aaai.v33i01.33018263.

Fefilatyev, S., Smarodzinava, V., Hall, L.O., Goldgof, D.B., 2006. Horizon detection using machine learning techniques. In: Proceedings of the 5th International Conference on Machine Learning and Applications (ICMLA'06). Orlando, FL, USA. https://doi.org/10.1109/ICMLA.2006.25, 14-16 Dec. 2006.

Fukushima, K., 1980. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. 36 (4), 193–202. https://doi.org/10.1007/BF00344251.

Garcia-Garcia, A., Orts, S., Oprea, S., Villena-Martinez, V., Rodríguez, J.G., 2017. A Review on Deep Learning Techniques Applied to Semantic Segmentation. ArXiv preprint. https://doi.org/10.48550/arXiv.1704.06857.

Gers, F., Schraudolph, N., Schmidhuber, J., 2002. Learning precise timing with LSTM recurrent networks. J. Mach. Learn. Res. 3, 115–143. https://doi.org/10.1162/153244303768966139.

Gers, F.A., Schmidhuber, J., Cummins, F., 1999. Learning to forget: continual prediction with LSTM. In: Proceedings of the Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470). Edinburgh, UK. https://doi.org/10.1049/cp:19991218, 7-10 Sept. 1999.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional Neural Networks For Mobile Vision Applications. arXiv preprint. https://doi.org/10.48550/arXiv.1704.04861.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.48550/arXiv.1608.06993.

Huntsberger, T., Aghazarian, H., Howard, A., Trotz, D.C., 2011. Stereo vision–based navigation for autonomous surface vessels. J. Field Robot. 28 (1), 3–18. https://doi.org/10.1002/rob.20380.

Ioffe, S., Szegedy, C., 2015. Batch Normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning, 37. PMLR, pp. 448–456. https://doi.org/10.48550/arXiv.1502.03167. B. Francis and B. David. Proceedings of Machine Learning Research.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA. https://doi.org/10.1109/CVPR.2014.223, 23-28 June 2014.

Kim, H., Koo, J., Kim, D., Park, B., Jo, Y., Myung, H., Lee, D., 2019. Vision-based real-time obstacle segmentation algorithm for autonomous surface vehicle. IEEE Access 7, 179420–179428. https://doi.org/10.1109/ACCESS.2019.2959312.

Kristan, M., Kenk, V.S., Kovačič, S., Perš, J., 2016. Fast image-based obstacle detection from unmanned surface vehicles. IEEE Trans. Cybern. 46 (3), 641–654. https://doi.org/10.1109/TCYB.2015.2412251.

Lee, S.J., Roh, M.I., Lee, H.W., Ha, J.S., Woo, I.G., 2018. Image-based ship detection and classification for unmanned surface vehicle using real-time object detection neural networks. In: Proceedings of the 28th International Ocean and Polar Engineering Conference. Sapporo, Japan.

Lin, G., Milan, A., Shen, C., Reid, I., 2017. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition. https://doi.org/10.48550/arXiv.1611.06612, 2017.

Liu, G., Guo, J., 2019. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing 337, 325–338. https://doi.org/10.1016/j.neucom.2019.01.078.

Liu, J., Li, H., Liu, J., Xie, S., Luo, J., 2021. Real-time monocular obstacle detection based on horizon line and saliency estimation for unmanned surface vehicles. Mob. Netw. Appl. 26 (3), 1372–1385. https://doi.org/10.1007/s11036-021-01752-2.

Liu, Y., Shen, C., Yu, C., Wang, J., 2020. Efficient semantic video segmentation with per-frame inference. In: Computer Vision–ECCV 2020: 16th European Conference. Glasgow, UK, pp. 352–368. https://doi.org/10.1007/978-3-030-58607-2_21.

Liu, Z., Zhang, Y., Yu, X., Yuan, C., 2016. Unmanned surface vehicles: an overview of developments and challenges. Annu. Rev. Control 41, 71–93. https://doi.org/10.1016/j.arcontrol.2016.04.018.

Liyong, M., Wei, X., Haibin, H., 2020. Convolutional neural network based obstacle detection for unmanned surface vehicle. Math. Biosci. Eng. 17 (1), 845–861. https://doi.org/10.3934/mbe.2020045.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D., 2022. Image segmentation using deep learning: a survey. IEEE Trans. Pattern Anal. Mach. Intell. 44 (7), 3523–3542. https://doi.org/10.1109/TPAMI.2021.3059968.

Paszke, A., Chaurasia, A., Kim, S., Culurciello, E., 2016. Enet: A deep Neural Network Architecture For Real-Time Semantic Segmentation. arXiv preprint. https://doi.org/10.48550/arXiv.1606.02147.

Pfeuffer, A., Schulz, K., Dietmayer, K., 2019. Semantic segmentation of video sequences with convolutional LSTMs. In: 2019 IEEE Intelligent Vehicles Symposium (IV). Paris, France. https://doi.org/10.1109/IVS.2019.8813852, 9-12 June 2019.

Prasad, D.K., Prasath, C.K., Rajan, D., Rachmawati, L., Rajabally, E., Quek, C., 2019. Object detection in a maritime environment: performance evaluation of background subtraction methods. IEEE Trans. Intell. Transp. Syst. 20 (5), 1787–1802. https://doi.org/10.1109/TITS.2018.2836399.

Prince, S.J.D., 2012. Computer Vision. Cambridge University Press Textbooks, Cambridge.

Rochan, M., 2018. Future Semantic Segmentation With Convolutional Lstm. arXiv preprint. https://doi.org/10.48550/arXiv.1807.07946.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: convolutional networks for biomedical image segmentation. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Munich, Germany. https://doi.org/10.48550/arXiv.1505.04597. October 5-9, 2015.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C., 2018. Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition. https://doi.org/10.48550/arXiv.1801.04381.

Seafloor Systems, I., Unmanned Surface Vehicles (USVs). Unmanned Systems Technology: USV selection from Seafloor Systems.

Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M., 2020. Financial time series forecasting with deep learning: a systematic literature review: 2005–2019. Appl. Soft Comput. 90, 106181 https://doi.org/10.1016/j.asoc.2020.106181.

Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c., 2015. Convolutional LSTM network: a machine learning approach for precipitation nowcasting. Adv. neural inf. process. syst. 28.

Srinivasu, P.N., SivaSai, J.G., Ijaz, M.F., Bhoi, A.K., Kim, W., Kang, J.J., 2021. Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM. Sensors 21 (8), 2852. https://doi.org/10.3390/s21082852.

Sundermeyer, M., Ney, H., Schlüter, R., 2015. From feedforward to recurrent LSTM neural networks for language modeling. IEEE ACM Trans. Audio Speech Lang. Process. 23 (3), 517–529. https://doi.org/10.1109/TASLP.2015.2400218.

Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., Baik, S.W., 2018. Action recognition in video sequences using deep bi-directional LSTM with CNN features. IEEE Access 6, 1155–1166. https://doi.org/10.1109/ACCESS.2017.2778011.

Vagale, A., Oucheikh, R., Bye, R.T., Osen, O.L., Fossen, T.I., 2021. Path planning and collision avoidance for autonomous surface vehicles I: a review. J. Mar. Sci. Technol. 26 (4), 1292–1306. https://doi.org/10.1007/s00773-020-00787-6.

Van Houdt, G., Mosquera, C., Nápoles, G., 2020. A review on the long short-term memory model. Artif. Intell. Rev. 53 (8), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1.

Varghese, S., Gujamagadi, S., Klingner, M., Kapoor, N., Bär, A., Schneider, J.D., Maag, K., Schlicht, P., Hüger, F., Fingscheidt, T., 2021. An unsupervised temporal consistency (TC) loss to improve the performance of semantic segmentation networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Nashville, TN, USA. https://doi.org/10.1109/CVPRW53098.2021.00010, 19-25 June 2021.

Wang, H., Wei, Z., Wang, S., Ow, C.S., Ho, K.T., Feng, B., 2011. A vision-based obstacle detection system for unmanned surface vehicle. In: Proceedings of the IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM). Qingdao, China. https://doi.org/10.1109/RAMECH.2011.6070512, 17-19 Sept. 2011.

Wu, Y., Yuan, M., Dong, S., Lin, L., Liu, Y., 2018. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. Neurocomputing 275, 167–179. https://doi.org/10.1016/j.neucom.2017.05.063.

Xu, F., Ma, H., Sun, J., Wu, R., Liu, X., Kong, Y., 2019. LSTM multi-modal UNet for brain tumor segmentation. In: Proceedings of the EEE 4th International Conference on Image, Vision and Computing (ICIVC). Xiamen, China. https://doi.org/10.1109/ICIVC47709.2019.8981027, 5-7 July 2019.

Xue, H., Chen, X., Zhang, R., Wu, P., Li, X., Liu, Y., 2021. Deep learning-based maritime environment segmentation for unmanned surface vehicles using superpixel algorithms. J. Mar. Sci. Eng. 9 (12), 1329. https://doi.org/10.3390/jmse9121329.

Yao, L., Kanoulas, D., Ji, Z., Liu, Y., 2021. ShorelineNet: an efficient deep learning approach for shoreline semantic segmentation for unmanned surface vehicles. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Prague, Czech Republic. https://doi.org/10.1109/IROS51168.2021.9636614, 27 Sept.-1 Oct. 2021.

Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N., 2018. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). https://doi.org/10.48550/arXiv.1808.00897, 2018.

Yu, Y., Si, X., Hu, C., Zhang, J., 2019. A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. 31 (7), 1235–1270. https://doi.org/10.1162/neco_a_01199.

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2016. Pyramid scene parsing network. In: Proceedings of the IEEE Conference Computer Vision Pattern Recognition (CVPR), pp. 6230–6239. https://doi.org/10.48550/arXiv.1612.01105.

Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., Wang, Q., 2020. Robust lane detection from continuous driving scenes using deep neural networks. IEEE Trans. Veh. Technol. 69 (1), 41–54. https://doi.org/10.1109/TVT.2019.2949603.

Žust, L., Kristan, M., 2022a. Learning maritime obstacle detection from weak annotations by scaffolding. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 3–8. https://doi.org/10.1109/WACV51458.2022.00195. Jan. 2022.

Žust, L., Kristan, M., 2022b. Temporal context for robust maritime obstacle detection. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Kyoto, Japan. https://doi.org/10.1109/IROS47612.2022.9982043, 2022.