Buoy Light Pattern Classification for Autonomous Ship Navigation using Recurrent Neural Networks

Schöller, Frederik Emil Thorsson; Nalpantidis, Lazaros; Blanke, Mogens

Published in:
IEEE Transactions on Intelligent Transportation Systems

Link to article, DOI:
10.1109/TITS.2021.3122275

Publication date:
2022

Document Version
Peer reviewed version

Buoy Light Pattern Classification for Autonomous Ship Navigation using Recurrent Neural Networks

Frederik E. T. Schöller and Lazaros Nalpantidis, *Senior Member, IEEE* and Mogens Blanke *Senior Member, IEEE*

***Abstract*—In near coast navigation, buoys and beacons convey essential information about dangers. At night-time, selected buoys send out individual blink-sequences that are marked in sea charts. International regulations require that navigation officer on watch makes visual confirmation of objects and their type in order to navigate safely. With rapid developments of highly automated vessels, this duty needs be carried out by algorithms that detect and locate objects without human intervention. At night-time, this requires algorithms that decode blink sequences and are able to classify from this information. The paper investigates this problem and suggests an algorithm that solves the problem.**

Convolutional Neural Networks (CNN) with Gated Recurrent Units (GRU) are developed for classification. A dedicated architecture is suggested that includes both temporal and color decoding to obtain unique precision. We demonstrate how networks are trained on synthetically generated data, and the paper shows, on real-world data, how the suggested approach yields 100.0% accurate results on 44 real-world recordings while being robust to inaccuracy in actual blink sequences.

Comparison with baseline signal processing and with a recent state-of-the-art 3D CNN model shows that the new blink-sequence classifier outperforms alternative algorithms.

A showcase of the results of this work is available in this video: https://youtu.be/KEi8qNnKV2w.

***Index Terms*—Autonomous navigation, Autonomous marine vessels, computer vision, deep learning, sequence classification**

I. Introduction

Marine autonomous surface ship technologies are gaining significant momentum. These technologies promise to provide services that include decision support for the manned vessel, making periodically unattended navigation possible, while smaller vessels could navigate completely unmanned. Proficient situational awareness is required day and night to provide such functionalities, and any mistake could cause a safety hazard.

Gaining situational awareness includes three main elements: perception, understanding of the environment, and anticipation of the development [1]. Whether a vessel is manned or unmanned, situational awareness is crucial for safe navigation [2], [3], and significant efforts have been reported on daytime awareness. Perception and understanding at daytime have included classification and tracking of objects, such as ships and buoys, are active topics of research [4], [5], [6], and robust classification from weather-degraded [7] or generally poorly annotated image data [8] is a challenge. At night-time, the same task becomes different and more demanding. Safe navigation in coastal waters requires that a ship avoids the areas of shallow water or other danger that are shown by buoys in the water and are marked in sea charts. Observing the bearing to light-buoys and beacons with flash sequences is the reliable and simple way to confirm own ship's position relative to areas of danger. Electronic means are efficient when they are operational, but defects, whether of physical or malicious nature, can easily mislead a ship into unsafe navigation [9]. Safe aids to navigation must therefore rely on buoys and beacons. At nighttime, these are recognized by signaling a flash sequence that indicates their type. The overall light pattern signals the type, but patterns are also coded to distinguish the same type at different locations along a route. A particular buoy or beacon hence emits a pre-selected pattern of light, which is part of the information found in navigational charts. Some buoys provide radio transmission of their position by AIS, while others respond to vessels' radar signals. These are, however, very few, and vision-based, automated classification of buoys at nighttime is of paramount importance for safe navigation.

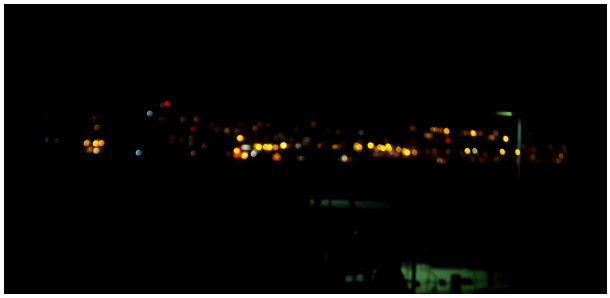


Fig. 1: Image taken along the Øresund strait that includes some of the blinking buoy patterns used for testing.

Marine vessels rely to a large extent on global navigation satellite systems (GNSS) for navigation, but spoofing and jamming attempts have become increasingly frequent [10]. Automatic buoy classification can help combat spoofing if supplemented by a comparison between environment and sea charts, similar to landmark recognition [11], [12]. The mapping from international regulations to required features of autonomous navigation architecture and algorithms are discussed in [13].

In this work, we investigate a camera-based approach for

All authors are with the Automation and Control Group, Department of Electrical Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark e-mail: {fets,lanalpa,mb}@elektro.dtu.dk

This work is sponsored by the Danish Innovation Fund, The Danish Maritime Fund, Orients Fund and the Lauritzen Foundation through the ShippingLab project (grant 8090-00063B).

Revised manuscript submitted October 29, 2021;

classifying buoy light patterns from image sequences. Figure 1 shows an overlook of a scene that contains a harbor line of lights plus a few blinking buoys along the coastline. The light buoys are essential for safe navigation. We are harnessing the strengths of deep learning and specifically of Convolutional Neural Networks (CNN) with Gated Recurrent Units (GRU). Furthermore, we choose to generate synthetic data for training, thus alleviating the laborious task of obtaining enough real-world data for training deep neural networks. Our method is meant to be deployed on a moving vessel in real-time applications. The vessel will be equipped with cameras that can provide a video stream of the buoy image sequences to be classified. The contributions of this paper are:

- A novel Recurrent Neural Network configuration with simultaneous color and pattern recognition coupled with a novel decision algorithm for deciding when a prediction is finalized and video acquisition can stop. The approach of having a decision algorithm between the network and the output has, to the authors' knowledge, not been published earlier.
- A method to simultaneously capture video and analyze its contents until a confident decision on the pattern is made.
- A simpler light pattern recognition method based on classical computer vision to be used as a reference in this work.
- A novel method for generating synthetic buoy light pattern videos.

The contents of the paper are as follows. Having introduced the problem and its context, we present related work in Section II and list the internationally standardized blink sequences that buoys and beacons emit in Section III, which also explains our approach to generate synthetic data for neural network training. Section IV introduces a standard solution for time-sequence identification from detection theory, which we use as a baseline. A novel method is suggested in Section V, describing a gated recurrent unit, attention method, architecture, and decision algorithm. Section VI presents results from testing on recorded videos from the sea and demonstrates the efficacy of the approach. Section VII concludes this work.

II. Related Work

The computer vision community has been interested in detecting and decoding signals transmitted by means of light pulses sequences. Such decoding systems have found application in a variety of scenarios. The use of optical recognition of Morse code signals was investigated within an Internet of Things (IoT) smart home security system [14]. Closer to this work is sequence analysis in the domain of video. A classical approach to video analysis can consist of detecting objects of interest using foreground segmentation and conclude based on the movement of the objects. [15] used this approach to analyze surveillance video data in order to detect vehicle collisions. Similarly, [16] used a Bayesian network to interpret the behavior of vehicles in traffic videos.

The extraction of patterns from video inputs is the focus of video content classification. A common approach to extract spatio-temporal features from videos for action recognition is by using 2D convolutional kernels within two-stream neural networks, i.e. networks that process RGB and optical flow information in parallel [17]. An extension of this work considers residual networks within the two-stream architecture [18] for training.

Another way of extracting spatio-temporal features is by using 3D convolutional kernels to process video input. Hara et. al [19] proposed a 3D CNN architecture inspired by the ResNet type networks [20]. The use of residual connections in a CNN with 3D convolutions showed to reduce overfitting and enabled the use of deeper networks for action recognition. Tran et. al [21] extended on this by factorizing the 3D convolutional filters into separate spatial and temporal components, which showed to significantly improve performance.

The work of [22] also used the 3D convolutional approach for video analysis but combined this with a CNN-RNN network for emotion recognition. Here, a combination of 3D CNNs and CNN-RNNs was used as it was argued that each method brought something different to the table which could be used for the recognition. While effective, 3D convolutions tend to be slow to compute making them suboptimal for online inference. Tran et. al [23] assessed this using 3D channel-separated convolutions instead of conventional convolutional layers. This approach showed to yield a network with performance comparable to state-of-the-art methods while being 2-3 times more efficient. Kondratyuk et. al [24] also sought to reduce inference time by using a combination of Neural Architecture Search (NAS), stream buffers, and temporal ensembles providing an 80% reduction in computational needs compared to other state-of-the-art methods.

Ullah et. al [25] utilized a CNN-RNN structure, a bidirectional CNN-LSTM, for video analysis. Frame skipping was used to decrease processing time while keeping high accuracy. A bidirectional architecture requires a complete video being available during prediction. In contrast, our use-case calls for prediction during image acquisition. Furthermore, the required video length that contains all needed information is, in our case, not known a priori.

Recently, transformer models have shown to perform very well on sequential data [26]. Transformer models have also shown encouraging results in image classification tasks. An example of this is the Vision Transformer (ViT), in which an image was treated as a sequence of 16x16 patches. Arnab et. al extended on this idea to be able to process video data [27]. The Video Vision Transformer showed to surpass the previous CNN based state of the art.

Common to all the mentioned methods is that a whole video is needed as the input for prediction. The method proposed in this work differs by utilizing a decision algorithm enabling online inference during image acquisition. A predetermined frame count is therefore not needed, as the method can decide by itself whether or not enough frames have been processed.

While, research on sequence decoding for action recognition and video classification is highly relevant for this application, to the best of the authors' knowledge, little to no research has been done in the area of visual classification of buoy light patterns.

III. Buoy Light Patterns

Deep learning methods constitute powerful tools for solving tasks such as the one considered in this work, but their performance relies on the availability of large amounts of accurately annotated training data. While testing of the trained models was performed on real-life videos, training data was chosen to be generated synthetically. Thus, the amount of data needed for training deep learning models could be obtained in an efficient manner. This decision was motivated by previously reported successes of using synthetically generated data for training deep neural networks, e.g. in [28].

The flash pattern of a buoy is determined by a sequence code. Examples of sequences are illustrated in Figure 2. The code consists of three parts: pattern, color, and period, as described in [29]. The following characteristics are used to describe unique blink sequences:
F (Fixed), Fl (Flash light), L (Long), Q (Quick) one blink per second, VQ (Very Quick) two blinks per second, Oc Occulting (darkness interrupted by periods of light), Iso same period for light and dark. These can be combined, such that FFl(3) indicates a buoy with fixed light followed by 3 flashes.

TABLE I: Nomenclature for buoy light patterns

type	T	{ Fl, LFl, Iso ... }	
colour	C	{ white ( ), red (R), green (G) }	
period	p	xx	$[sec]$
light(1)	l_1	x.x	$[sec]$
dark(1)	d_1	x.x	$[sec]$
light(2)	l_1	x.x	$[sec]$
dark(2)	d_2	x.x	$[sec]$
flashes(1)	n_1	cardinality	
flashes(2)	n_2	cardinality	
morse	M	morse letter	

As an example the buoy $Fl(3)G.15s$ $(1.5 + 1.5 + 1.5 + 1.5 + 1.5 + 7.5)$ has a period of $15s$. It sends one green flash of duration $1.5s$, followed by darkness during $1.5s$. This is repeated three times. After the third green flash follows darkness for the remaining $7.5s$ signal. Hereafter the sequence is repeated. A light duration of 10% of the period is usually the case for light buoys. Using the abbreviations presented in Table I, a sequence with n flashes has the generic form $Fl(n).C.p = (l_1, d_1, ...l_n, d_2)$.

For this paper, the 10 most common red and green sequences were chosen for classification. Additional sequences could easily be incorporated.

For reasons of brevity, the methods presented in this paper, assume that a buoy is pre-detected by a detection algorithm, and a cropped video sample of the specific buoy is provided to the classification algorithm.

A. Synthetic Data Generation

In order to generate synthetic training data sequences, two images are needed: a background image and an image with a flashing light source. These images are then arranged in the specific order that makes the sequence. For a 25 fps video, the *Fl.G.10s* sequence would consist of 25 flash images followed by 225 background images. The sequence is looped in order to obtain a certain length, and a random segment of a chosen length is selected for the final sequence video. In this research, the generated videos have a length of 30 seconds to ensure we can accommodate the longer blink sequences. The background image is generated by randomly selecting a 32×32 pixel patch from a selection of background images. The background images were night-time cityscapes seen from a ship to include relevant noise and visual clutter.

The flash image is obtained by a blurred circle of random size is drawn on top of the background image at a random location. The color of the circle is chosen to appropriately match that of a buoy light.

Initial analysis of buoy pattern videos showed that the timing of the buoy patterns varies quite a lot from what is stated in the nominal requirement for the hardware, especially for the Fl.3s pattern, where some light flashes lasted up to 160% longer than what they were supposed to. While this does not affect human classification of the signals, it has a huge impact on how a computer sees the sequence. To accommodate for this and for timing variations imposed by camera frame rate, randomness is included in the timing of the generated sequences, uniformly varying the period with up to $\pm10\%$ and the light duration with up to $\pm20\%$. Specifically for Fl.3s patterns, the base light duration is chosen randomly to either 0.3, 0.5 or 0.7 seconds as these were observed in the gathered videos. Furthermore, Gaussian image noise with mean $\mu = 0$ and variance $\sigma^2 = R$, where R is a sequence-specific random variable following the uniform distribution between 1 and 12, is added to the generated sequence in order to emulate image noise. The temporal noise and image noise are superimposed separately. Two generated data examples can be seen in Figure 3.

To evaluate the method, testing was performed on real-life videos of buoy light patterns. The videos are captured both from the sea and from land overlooking the open sea. Test data includes examples both with and without city lights in the background. The test set has the following patterns: *Fl.3s, Fl.5s, Fl(2).5s, Fl(3).10s, Q, Iso.3s* as these were the patterns present during our data collection. The collection is

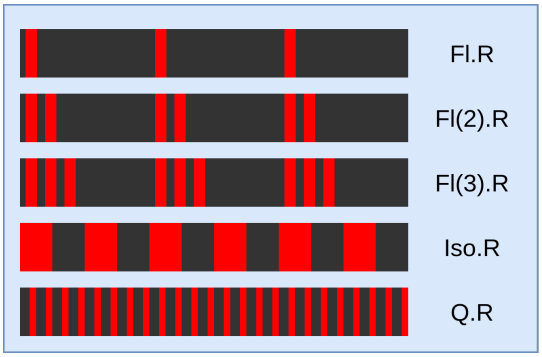
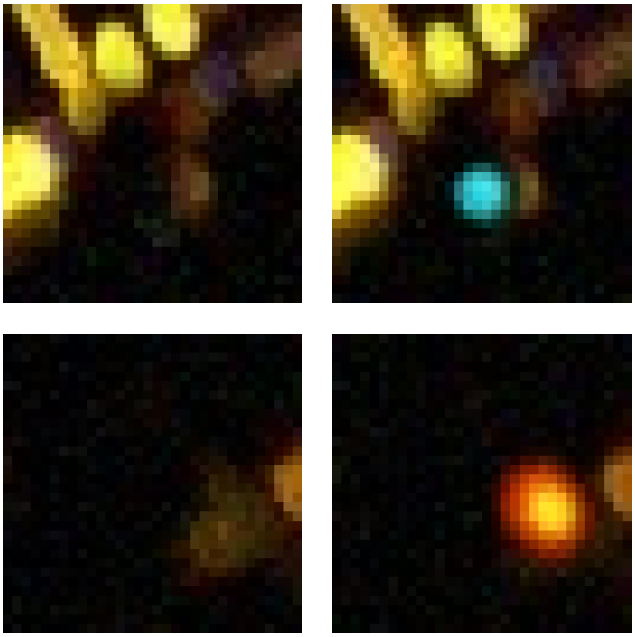
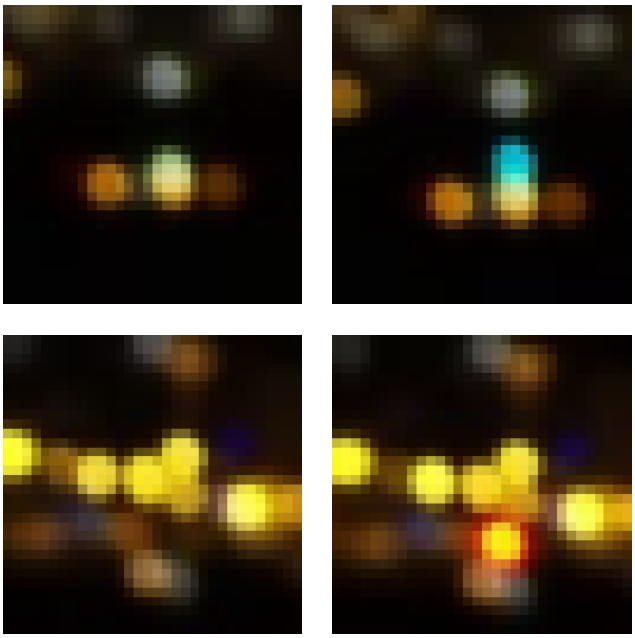


Fig. 2: Examples of common flash sequences for red lights. The same patterns exist for buoys with green flash lights. These are general illustrations of buoy light patterns as presented for navigators, where the period is omitted [30]. The exact definition of generic sequences is presented in Table I.



(a) Background images (b) Flash images

Fig. 3: Examples of generated training images



(a) Background images (b) Flash images

Fig. 4: Examples of real-world test data, taken from the port of Elsinore looking towards Sweden.

done by first recording a video and then manually cropping out the area in which the buoy light is contained. The crop is done with a large margin around the blink to accommodate vibrations and ship movement. While the blink videos were prepared manually, this could be done automatically for real-world implementation.

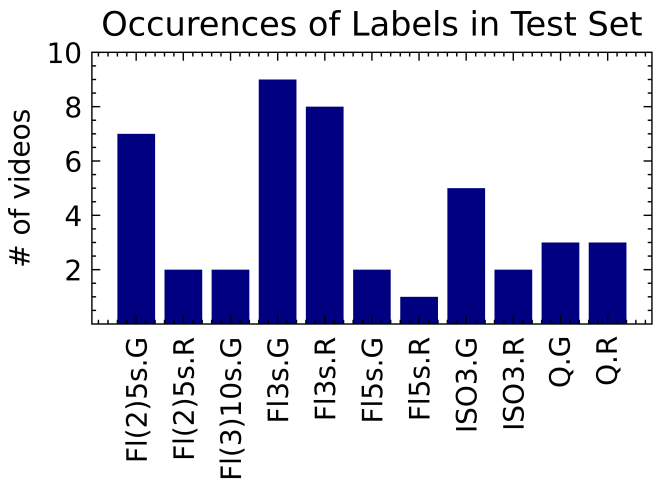


Fig. 5: Occurrences of the different buoys in the real-world videos, recorded at Oresund and the South Funen archipelago.

Such an approach could be based on a combination of detection and tracking of the subject. Subject detection could be based on deep learning methods such as object detection, which is a widely researched subject for marine environments, [31], [32], [33]. Detection could also be based on obstacle segmentation methods such as investigated by [34]. Tracking of the subject could be based on subsequent detection [35] or by using optical flow [36]. While recording the buoy lights, the camera was often set out of focus on purpose, as it was found to give better color reproduction. In total, 44 different buoy videos were collected for testing, an example of which can be seen on Figure 4. The occurrence of the different labels in the test set is illustrated in Figure 5.

IV. Baseline Method

In this section, a baseline method based on classical image analysis and detection theory is presented.

A. Detection Theory Based Method

Given the set of all possible blink sequences $\boldsymbol{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2 \dots \boldsymbol{s}_N\}$, and an observed sequence $\boldsymbol{z}$.

A hypothesis is $\mathcal{H}_i$ if the observed sequence $\boldsymbol{z}$ comprises the sequence $\boldsymbol{s}_i$, i.e.

$$\mathcal{H}_i : z(k) = s_i(k - k_{0i}) + w(k), \quad k = 0, 1, \dots, K-1 \quad (1)$$

where k_{0i} is the unknown delay of s_i and $w(k)$ is white Gaussian noise. The unknown delay k_{0i} of sequence $\boldsymbol{s}_i$ is estimated by the maximum likelihood estimate over all possible values of k_0,

$$\hat{k}_{0i} = \max_{k_0} \sum_{k=k_0}^{k_0+K-1} z(k) s_i(k - k_0). \quad (2)$$

With an observed sequence $\boldsymbol{z}$, we should choose $\mathcal{H}_i$ for which $p(\boldsymbol{z}|\mathcal{H}_i)$ is maximum. This is equivalent to choosing the $\boldsymbol{s}_i$ that gives minimum relative distance $\mathcal{D}_i$ between $\boldsymbol{z}(k - \hat{k}_{0i})$ and $\boldsymbol{s}_i$:

$$\mathcal{D}_i^2 = \frac{\sum_{k=0}^{K-1} (z(k + \hat{k}_{0i}) - s_i(k))^2}{\sum_{k=0}^{K-1} (s_i(k))^2} \quad (3)$$

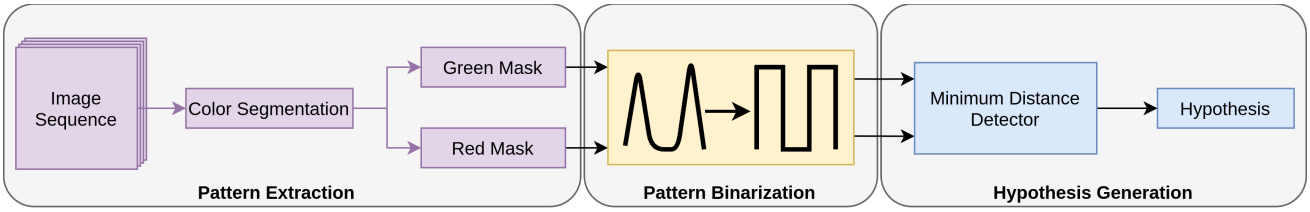


Fig. 6: Block diagram illustrating the detection theory-based baseline method. A light pattern is extracted from the image sequence by generating a green and red color mask using color thresholding. The color masks are binarized using Otsu's method. A hypothesis is then made based on the correlation between the binarized patterns and a bank of possible buoy patterns.

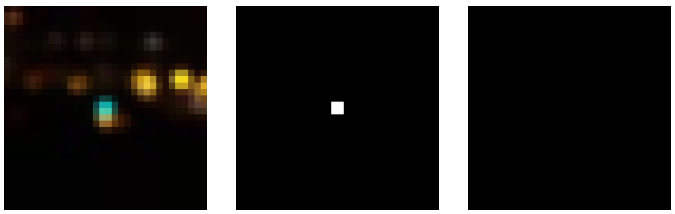
When choosing among N blink sequences, N can be reduced by involving a prior probability that a sequence exists in the area A of interest, $P(\boldsymbol{s}_i|ENC(A))$, where $ENC(A)$ is sea chart information about buoys for a specified area, usually the visible range given the altitude of the observer.

Hence, $\mathcal{H}_i$ is chosen as follows,

$$\mathcal{H}_i = \arg\max_i \left((1 - \mathcal{D}_i) \, P(\boldsymbol{s}_i|ENC(lat, lon)) \right) \quad (4)$$

B. Signal Conditioning

The detection theory based method is used together with classical image analysis as a baseline detector. A block diagram illustrating the data flow of the baseline algorithm is shown in Figure 6. First, a template bank is made, consisting of the N target sequences $\boldsymbol{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2 \dots \boldsymbol{s}_N\}$ to populate the vocabulary of sequences. To detect blink sequences from a video stream, the video is first converted to the HSV color space. The Hue (color) channel of each frame is then thresholded to exclusively include the color range matching that of a buoy light, with options $C = \{green, red\}$. The thresholded image is now a mask indicating whether or not a pixel is in the given color range. The mask processing for red and green are illustrated in Figure 7.



(a) Original image (b) Green mask (c) Red mask

Fig. 7: Example of the color separation for the baseline detection model showing no red light in the original image and some green light

For each frame in the video, the mean intensity of the thresholded image is added to the respective sequence. The result is, for each image, a sample $z_c(k)$ for each channel in C. The values of the proposal sequences are then thresholded to obtain a binary sequence of whether or not a blink is active in a given frame. The threshold level is based on first and second-order statistics using *Otsu's method* [37]. A step-by-step example of the extraction of a binary sequence is illustrated in Figure 10.

The proposed sequences $\boldsymbol{z}_c$ are then compared to each sequence $\boldsymbol{s}_i$ in the template bank $\boldsymbol{S}$, using Equations 2 to 4 for detection. This results in finding the template sequence $\boldsymbol{s}_i$ that has the closest correlation with the observed pattern, given the geographic area of interest. This includes the color of the sequence. For testing purposes, $P(\boldsymbol{s}_i|ENC(lat, lon))$ is set uniformly for each sequence type.

In real life, blink sequences have uncertainty, in particular in the duration of the short flashes in a sequence. The detection theory method achieved an accuracy of 64.0% on synthetically generated data where uncertainty in flash duration was modeled and 70.5% on the real videos available from Danish waters.

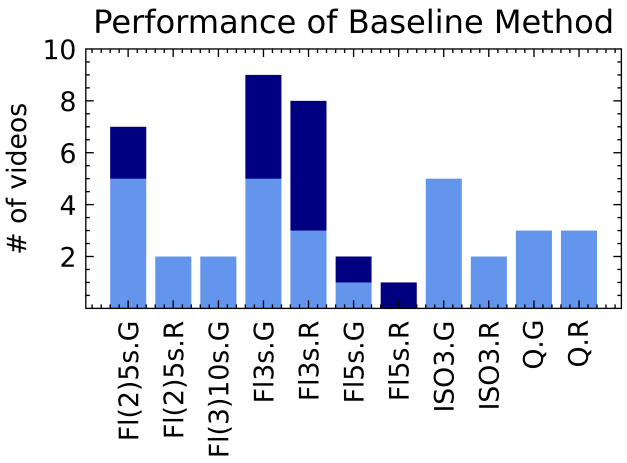


Fig. 8: Histogram showing the number of real-world videos correctly classified by the baseline method. The light blue bars are the number of correct classifications of a given sequence whereas the dark blue bars are the total number of the given label in the test set.

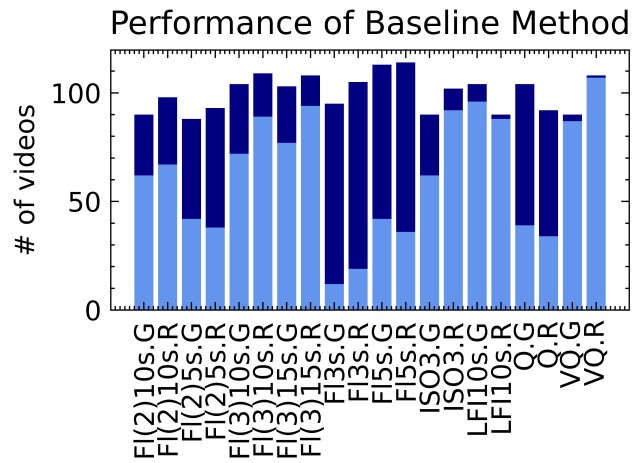


Fig. 9: Histogram showing the number of generated videos correctly classified by the baseline method. The light blue bars are the number of correct classifications of a given sequence whereas the dark blue bars are the total number of the given label generated.

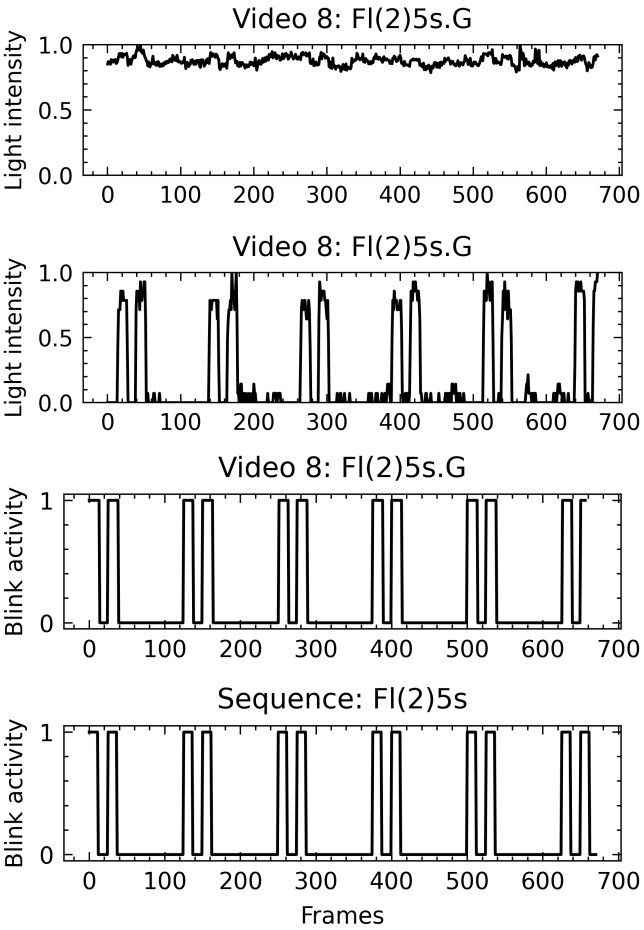


Fig. 10: Raw color channel (top), Color thresholded signal (second), Otsu thresholded signal (third), template sequence with the highest correlation (bottom).

V. Proposed Method

This section describes the components of the proposed method, as well as the structure of the model architecture.

A. Gated Recurrent Unit

The Gated Recurrent Unit (GRU) [38] is a type of RNN. The GRU was motivated by the Long short-term memory (LSTM) [39], which addresses the vanishing gradients problem of the classical RNN's by using gates to control the flow of information. Unlike a classical feed-forward neural network, a GRU is able to learn temporal information by keeping the current network state throughout a sequence. The GRU has two gates: The reset gate r_t, which determines what part of the previous information should be dropped before the update, and the update gate z_t, which controls how much information from the previous hidden state will carry over to the current hidden state. The update of the GRU can be described as:

$$z_t = \sigma_g \left(W_z x_t + U_z h_{t-1} + b_z\right) \tag{5}$$

$$r_t = \sigma_g \left(W_r x_t + U_r h_{t-1} + b_r\right) \tag{6}$$

$$\hat{h}_t = \phi_h \left(W_h x_t + U_h \left(r_t \odot h_{t-1}\right) + b_h\right) \tag{7}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t, \tag{8}$$

Where x_t is the input vector, h_t is the output hidden state, W and U are weight matrices, b_z, b_r and b_h denote bias components, σ_g is the sigmoid activation function and ϕ_h is the hyperbolic tangent activation function. For each frame in the input video, the GRU is updated using the CNN features of the current frame, after which the hidden state h_t will contain the encoded information about the video so far. When a new video is to be classified, the GRU has to be reset by setting $h_t = 0$.

B. Attention

After the GRU has encoded the temporal information in the video features, we are left with a final hidden state. This final hidden state carries the burden of encoding the entire meaning of the video into a single vector of limited size. Attention mechanisms try to accommodate this by making the classifier focus on the important parts of the input through the hidden state of each time step instead of relying on a single vector. For our method, the global attention method proposed by Luong et al. [40] has been implemented. given a context vector, c_t consisting of the hidden states from each time step and the encoder hidden state, h_t, the attentional hidden state is computed as

$$\tilde{h}_t = \phi_h \left(W_c \left[c_t; h_t\right]\right) \tag{9}$$

the final attention vector will then be

$$att = \text{softmax}\left(W_s \tilde{h}_t\right) \tag{10}$$

For global Luong attention [40], c_t is computed as a weighted average of the input encoder hidden state vector. The weighting, a_t is found by comparing the score of the

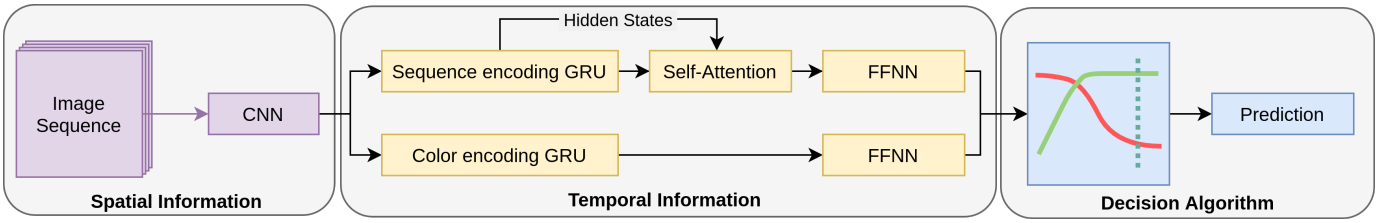


Fig. 11: Block diagram showing the data flow of the proposed method. The image features of a live video are sent to two different subnets in order to determine buoy sequence and color separately. The confidence in each buoy type is analyzed using a novel decision algorithm to determine when a prediction is final and the image acquisition can stop.

final hidden state with the score of the hidden state $\overline{h}_s$ from each step using *softmax*:

$$a_t(s) = \frac{\exp\left(\text{score}\left(h_t, \overline{h}_s\right)\right)}{\sum_{s'} \exp\left(\text{score}\left(h_t, \overline{h}_{s'}\right)\right)} \tag{11}$$

, where the score function can be defined using the following three methods:

$$\text{score}\left(h_t, \overline{h}_s\right) = \begin{cases} h_t^\top \overline{h}_s & \text{dot} \\ h_t^\top W_a \overline{h}_s & \text{general} \\ v_a^\top \tanh\left(W_a \left[h_t; \overline{h}_s\right]\right) & \text{concat} \end{cases} \tag{12}$$

For this paper, the *general* attention method is used as is proved to generalize better during testing compared to the *dot* method.

C. Architecture

In order to classify the generated videos, a CNN-RNN network is used, specifically a CNN-GRU configuration. The network consists of three parts: first a feature extractor, then a feature encoder, and lastly a classifier. The overall structure of the proposed CNN-GRU network is shown in Fig. 11.

The feature extractor consists of a residual CNN. The residual CNN is composed of residual blocks [20], described in Table II. Here, each convolutional layer is followed by batch normalization and leaky ReLU activation. The *Residual* layer in Table II performs the downsizing of the input to the proper size using a *conv2d* layer with a stride of 2 and adding the result to the output of the previous layer. The parameter values for the CNN feature extractor were chosen following the recommendations presented in [20] with a focus on creating a small network to accommodate the 32×32 pixel size of the video frames.

TABLE II:
Structure of residual block used in CNN feature extractor

Layer	Kernel size	stride
conv2d	3	1
conv2d	3	1
Residual	3	2

The CNN feature extractor is built using multiple such residual blocks, as seen in Table III. The pooling layer used at the output of the CNN is an *Adaptive average pooling* layer. This type of pooling returns the average response of each channel in the input and thereby returns a vector. This is done in order to remove all spatial information from the image, as the position of the buoy within the image should not affect the network predictions.

TABLE III: Structure of CNN feature extractor

Layer	output size
Input	(3,32,32)
conv2d	(8,32,32)
Residual block	(8,16,16)
Residual block	(16,8,8)
Residual block	(32,4,4)
Residual block	(64,2,2)
pool	(64,1,1)

After the CNN has extracted relevant features from the video, the temporal features are encoded into a vector representation. As the buoy light patterns are consisting of a sequence definition and a color definition, the proposed model is likewise split into two subnets: one for sequence encodement and one for color encodement. This is done as sequence and color are strictly decoupled. Both subnets consist of a GRU and a feedforward neural network (FFNN) classifier. Self-attention is added to the sequence encoding subnet as it has been found to improve performance on longer sequences [26]. The final GRU hidden state, h_t is sent to the self-attention together with a concatenation of all previous hidden states h_o. The self-attention output is then fed to the FFNN classifier for sequence estimation. The color subnet does not utilize self-attention as the GRU only has to remember the color of the latest blink. The GRUs are single-layer and uni-directional.

TABLE IV: Structure of the classification network

Layer	hidden size
Input	h_s
Linear	256
BN	256
Dropout	256
ReLU	256
Linear	N

Table IV shows the structure of the classification subnetworks used for classifying the encoded sequences. Here, h_s is the number of hidden units in the GRU and N is the output size. The GRU in the sequence subnet has a hidden size of $h_s = 128$ while the GRU in the color subnet has $h_s = 8$. Furthermore $N = 10$ for the sequence classification subnetwork as 10 different patterns are considered. The FFNN

classifier of the color subnet has $N = 2$ as two colors are considered. The hidden size of 256 neurons was manually chosen during initial testing as it showed to yield good results. The two outputs from each network strand, i.e. predicted sequence and color, are optimized using cross-entropy loss with equal weighting.

D. Decision Algorithm

During run time, the network analyzes a single image at a time, as a video of a buoy pattern is collected sequentially. The GRU will therefore be set in a state-full configuration, meaning that the hidden state obtained from analyzing the previous image is kept and used when analyzing the current image.

With images collected sequentially, it is desired to know whether the current video has been adequately processed or if image acquisition should continue. A novel algorithm for determining this has been implemented. We call this the *Decision Algorithm*. The hybrid approach of having an RNN model output predictions and a deterministic algorithm decide on the appropriate prediction has, to our knowledge, not been explored before. The decision is made based on the model confidence as a function of time. Simply deciding on a prediction when model confidence alone is above a given threshold is not enough, as it might take some time before the model settles on a prediction, before which, large fluctuations in confidence can occur. The decision algorithm, therefore, includes multiple measures:

- Running average of model confidence.
- Running standard deviation of model confidence.
- Running average slope of model confidence.
- Time threshold.

A running average of the confidences is used to choose the prediction with adequate confidence. The running standard deviation of the model confidences is used to estimate whether the model has converged to a prediction. Time thresholding is used such that the model can not make a prediction before having seen enough footage of a specific pattern. The time threshold is a factor of the period of the current prediction. This means that, if the time threshold is e.g. 2, the model will only be able to decide upon a pattern with a 5s period after having seen 10s of footage. Running average slope of model confidence is used such that a prediction is not chosen if the confidence is currently decreasing too much. The hyperparameter values for the decision algorithm were found with the *TPE algorithm* for hyperparameter optimization [41]. Using generated data, the parameters were optimized for both accuracy and average decision time. Optimization yielded the values provided in Table V.

TABLE V: Decision Algorithm's hyperparameter values

Parameter	Value
Confidence, pattern	0.66
Confidence, color	0.79
Slope, pattern	-0.13
Slope, color	-0.08
Mean std	0.017
Time threshold	1.9
Roll	68

This configuration yielded a mean time to decide of 2.31 periods during optimization.

VI. Results

This section presents the results found when testing the proposed method. The method is primarily tested on real buoy videos obtained in a marine environment, both on-board a moving vessel and from land overlooking the coast. The proposed method is provided with a single image frame at a time, to emulate video acquisition in a real-time scenario. Prediction confidence is obtained by calculating the *softmax* of the network output activation. A prediction is deemed final when the decision algorithm is triggered. The performance metric used to evaluate the method is accuracy, i.e. the amount of correctly classified videos divided by the total number of videos. The decision algorithm is configured with the hyper-parameters shown in Table V, and model dimensions are as described in Section V-C. A video showcasing the performance on real buoy patterns can be found at https://youtu.be/KEi8qNnKV2w.

Evaluating the proposed method on the 44 buoy videos yielded a test accuracy of 100.0% as the model was able to correctly classify all the videos. As the videos were collected in marine environments both during sea tests and in harbor, this shows that the method has good potential for eventual deployment. While a statistically significant assessment of accuracy requires a higher number of videos from sea tests at night, the results show that the method is able to classify real-world buoy light patterns, even though it was trained using purely generated data.

A. Comparision with a R3D-18 3D CNN architecture

As a comparison with other deep learning methods, the R3D-18 3D CNN architecture proposed in [21] was trained and evaluated. This specific architecture was chosen for comparison as it obtained close to state-of-the-art results while having a reasonable parameter count. The R3D-18 model showed to have an accuracy of 95.45% when tested on real-world buoy videos. The videos that were not classified correctly were predicted as being of a different buoy pattern. In a real-world scenario, a misclassification could lead to an autonomous vessel miscalculating its position and in the worst case cause a grounding. However, it should be noted that the 3D CNN is designed to examine complete video clips and not one frame at a time. This means that in order to classify a buoy pattern in a deployed situation, it would take the 3D CNN 30 seconds to do a single classification, as it is desired to have footage of at least two periods of the pattern. As the longest period in the patterns investigated is 15 seconds,

this corresponds to a 30 second detection time. The method proposed in this work can classify a signal in as low as six seconds for a *FL3s* buoy. A summary of the performance of the different methods is presented in Table VI.

TABLE VI: Performance of different methods

Method	Test Accuracy	Approximate Detection Time
Baseline	70.5%	$30s$
R3D-18 [21]	95.45%	$30s$
Proposed	**100%**	$6s$ to $30s$, depending on pattern

Some of the more interesting testing videos are shown in Figure 12. Here, frames with and without the blink are shown together with a graph presenting the blink pattern. The blink pattern graph is derived using the method described in Section IV.

Figure 12 shows example frames for four different videos from the test set along with the evolution over time of the model confidence for the various hypotheses for the pattern and color. The confidence plot ends where the decision algorithm is triggered, with the final prediction stated on top of the plot. Looking at the confidence plot of Figure 12b, the importance of the decision algorithm can be seen. Here, the confidence in the wrong label *Iso3s.R* started out being above the decision threshold, though the algorithm did not trigger due to high standard deviation. Later, the right label *Fl3s.R* gained confidence, and the decision algorithm triggered a correct prediction. Similarly, for Figure 12d, it can be seen that the model settles on the hypothesis *FL(3)10s* after ~ 200 frames of video. Here, the decision algorithm does not trigger due to the time threshold which dictates that 1.9 periods of video have to be processed before a decision can be made. In this example, 1.9 periods correspond to 475 frames for a *FL(3)10s* pattern. Later in the example, the model gains confidence in the *FL(2)5s* hypothesis, which triggers once the confidence is high enough as this specific pattern needs 238 processed frames to trigger. Similar behavior can be seen in Figure 12c.

To test the performance on all of the implemented patterns, the method is evaluated on 2000 sequences of new generated data. While this is not directly comparable to real data, it can address whether the network has difficulties with certain patterns. The new sequences are generated in the manner described in Section III and are not included in the training data. This test yielded an accuracy of 98%, classifying correctly 1960 sequences. The labels of the remaining 40 misclassified sequences can be seen in the histogram of Figure 13. There does not seem to be a significant correlation between the complexity of the patterns and misclassification. A relatively complex pattern such as *Fl(2)10s* has roughly the probability for misclassification as a simple pattern such as *Fl5s*.

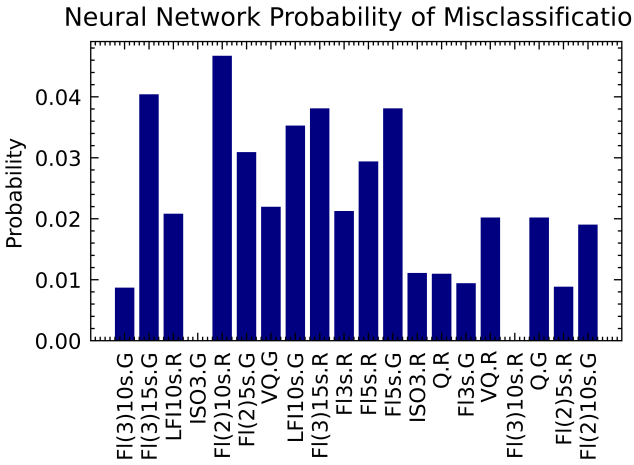


Fig. 13: Histogram showing the probability of misclassification for a given class during a test on synthetically generated data.

B. Robustness test

To test the robustness of the two methods, the test data is augmented using various methods as seen in Table VII. Using the same seed for both methods, the test set is augmented 5 times using the augmentations listed. It is clearly seen that the CNN-GRU approach results in a more robust method with a maximum accuracy degradation of 4.1% points when augmenting with motion blur, while the baseline has a maximum degradation of 10.5 % points for both white balance shifts and frame drops. Especially frame dropping is interesting as it was found that the blink timing of buoys varied quite a lot from the specified timing, making frame dropping representative augmentation. Augmentations were performed using the *imgaug* python library [42].

We could have included the augmentations listed in Table VII during training and we considered the possibility that the model might have been even more robust to these individual augmentations. Initial testing showed, however, that training with heavy augmentations, like motion blur, made the generated data less representative, which resulted in degraded generalization. Thus, it was decided not to consider augmentation during training.

TABLE VII: Result of robustness test on buoy videos

Augmentation	Baseline Accuracy	Network Accuracy
No Augmentation	70.5%	100%
Motion blur	61.4%	95.9%
Gaussian blur	65.5%	99.5%
White balance shift	60.0%	98.2%
Mean shift	65.9%	100.0%
Frame drop	60.0%	99.1%
Mean Accuracy	63.9%	98.8%
Mean Acc. Reduction	7.9%	**1.5%**

VII. Conclusions

This paper proposed a novel method for classifying buoy light patterns. The method consisted of a convolutional neural network with gated recurrent units (CNN-GRU). A novel

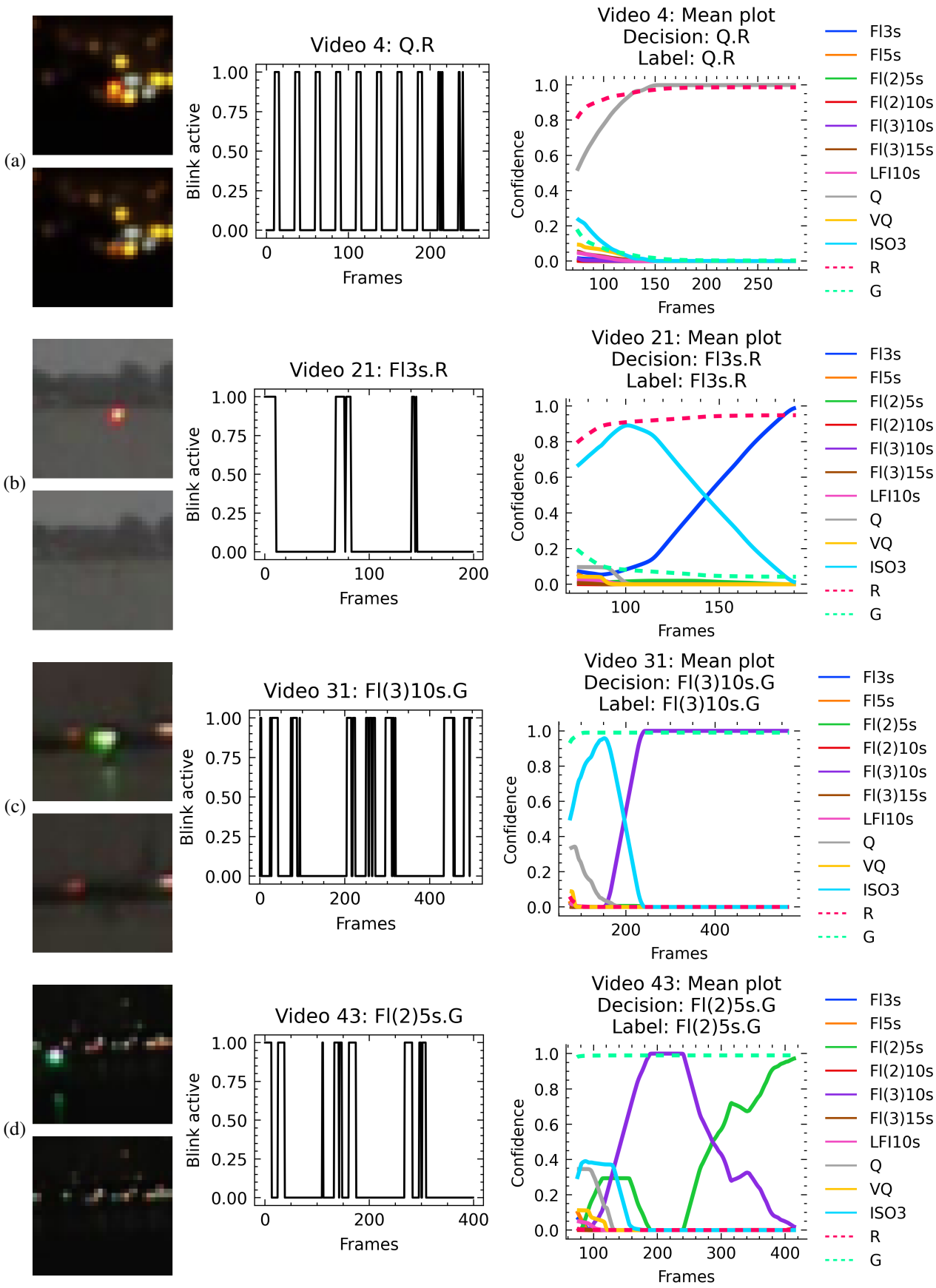


Fig. 12: Example frames from test videos (left) along with the derived blinking pattern (center) and the model response (right)

architecture was suggested consisting of parallel paths dealing with temporal and color information, respectively. The network was trained on synthetic data and validated on video streams from the sea. Training on synthetic data provided a network that was able to generalize and correctly classify the light patterns from 100,0% of buoys present in the 44 real-world videos we had available. This was achieved with the algorithm running in real-time. It was observed that timing in light sequences from buoys often deviates significantly from specifications. The algorithm was robust to these real-world deviations in flash duration. The performance of the proposed CNN-GRU network clearly surpasses that of the classical minimum distance detector that is based on correlation and multiple-models hypothesis tests. The paper demonstrated that the deep neural network approach provides a robust night-time classification of flash sequences from buoys at sea.

REFERENCES

[1] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human factors*, vol. 37, no. 1, pp. 32–64, 1995.

[2] M. Blanke, S. Hansen, J. D. Stets, T. Koester, J. E. Brøsted, A. Llopart Maurin, N. Nykvist, and J. Bang, "Outlook for navigation – comparing human performance with a robotic solution," in *Proc. of ICMASS'2018*. SINTEF Academic Press, May 2019. [Online]. Available: http://hdl.handle.net/11250/2599009

[3] T. Porathe, J. Prison, and Y. Man, "Situation awareness in remote control centres for unmanned ships," in *Proceedings of Human Factors in Ship Design & Operation, 26-27 February 2014, London, UK p. 93-101*, 2014.

[4] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 1993–2016, 2017.

[5] X. Mou, X. Chen, J. Guan, B. Chen, and Y. Dong, "Marine target detection based on improved faster r-cnn for navigation radar ppi images," in *2019 International Conference on Control, Automation and Information Sciences (ICCAIS)*, 2019, pp. 1–5.

[6] J. D. Stets, F. E. T. Schöller, M. K. Plenge-Feidenhans'l, R. H. Andersen, S. Hansen, and M. Blanke, "Comparing spectral bands for object detection at sea using convolutional neural networks," *Journal of Physics: Conference Series*, vol. 1357, p. 012036, oct 2019.

[7] J. B. Becktor, F. E. T. Schöller, E. Boukas, M. Blanke, and L. Nalpantidis, "Lipschitz constrained neural networks for robust object detection at sea," in *International Conference on Maritime Autonomous Surface Ship (ICMASS)*, Ulsan, Korea, 2020.

[8] J. Becktor, E. Boukas, M. Blanke, and L. Nalpantidis, "Reweighting neural network examples for robust object detection at sea," *Electronics Letters*, 2021.

[9] J. Bhatti and T. E. Humphreys, "Hostile control of ships via false GPS signals: Demonstration and detection," *NAVIGATION*, vol. 64, no. 1, pp. 51–66, 2017.

[10] ——, "Hostile control of ships via false gps signals: Demonstration and detection," *NAVIGATION: Journal of the Institute of Navigation*, vol. 64, no. 1, pp. 51–66, 2017.

[11] N. Engel, S. Hoermann, M. Horn, V. Belagiannis, and K. Dietmayer, "Deeplocalization: Landmark-based self-localization with deep neural networks," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 926–933.

[12] M. C. Nissov, D. Dagdilelis, R. Galeazzi, and M. Blanke, "Analyzing cyber-resiliency of a marine navigation system from behavioral relations," *IEEE Xplore, Proc. European Control Conference 2021*, June 2021.

[13] K. Dittmann, P. N. Hansen, D. Papageorgiou, S. Jensen, M. Lützen, and M. Blanke, "Autonomous surface vessel with remote human on the loop: System design for stcw compliance," *IFACpapers-Online, Proc. IFAC CAMS'2021*, 2021.

[14] C. Lee, T. Shen, W. Lee, and K. Weng, "A novel electronic lock using optical morse code based on the internet of things," in *2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE)*, 2016, pp. 585–588.

[15] S. S. Thomas, S. Gupta, and V. K. Subramanian, "Event detection on roads using perceptual video summarization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2944–2954, 2018.

[16] P. Kumar, S. Ranganath, Huang Weimin, and K. Sengupta, "Framework for real-time behavior interpretation from traffic video," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 43–53, 2005.

[17] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1933–1941.

[18] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal residual networks for video action recognition," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16, 2016, p. 3476–3484.

[19] K. Hara, H. Kataoka, and Y. Satoh, "Learning spatio-temporal features with 3d residual networks for action recognition," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 3154–3160.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[21] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[22] Y. Fan, X. Lu, D. Li, and Y. Liu, "Video-based emotion recognition using cnn-rnn and c3d hybrid networks," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, 2016, p. 445–450.

[23] D. Tran, H. Wang, L. Torresani, and M. Feiszli, "Video classification with channel-separated convolutional networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5552–5561.

[24] D. Kondratyuk, L. Yuan, Y. Li, L. Zhang, M. Tan, M. Brown, and B. Gong, "Movinets: Mobile video networks for efficient video recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 020–16 030.

[25] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional lstm with cnn features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[27] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," 2021.

[28] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018, pp. 969–977.

[29] U. S. N. G.-I. Agency, *List of Lights, publication 116*, 2020. [Online]. Available: https://msi.nga.mil/Publications/NGALOL

[30] IALA, "R0110 rhythmic characters of lights on marine aids to navigation edition 5.0," International Association of Marine Aids to Navigation and Lighthouse Authorities, urn:mrn:iala:pub:r0110, 2021.

[31] F. Schöller, M. K. Plenge-Feidenhans'l, J. D. Stets, and M. Blanke, "Assessing deep-learning methods for object detection at sea from LWIR images," *IFAC-PapersOnLine*, vol. 52, no. 21, pp. 64 – 71, 2019, 12th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2019.

[32] R. Spraul, L. Sommer, and A. Schumann, "A comprehensive analysis of modern object detection methods for maritime vessel detection," vol. 11543. SPIE, 2020.

[33] F. Farahnakian and J. Heikkonen, "Deep learning based multi-modal fusion architectures for maritime vessel detection," *Remote Sensing*, vol. 12, p. 2509, 08 2020.

[34] H. Kim, J. Koo, D. Kim, B. Park, Y. Jo, H. Myung, and D. Lee, "Vision-based real-time obstacle segmentation algorithm for autonomous surface vehicle," *IEEE Access*, vol. 7, pp. 179 420–179 428, 2019.

[35] F. Schöller, M. Blanke, M. K. Plenge-Feidenhans'l, and L. Nalpantidis, "Vision-based object tracking in marine environments using features from neural network detections," *submitted to IFAC-PapersOnLine*, 2020, iFAC World Congress 2020.

[36] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th Interna-*

tional Joint Conference on Artificial Intelligence - Volume 2. Morgan Kaufmann Publishers Inc., 1981, p. 674–679.

[37] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[38] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, 2014.

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[40] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[41] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, 2011, pp. 2546–2554.

[42] A. B. Jung, K. Wada, J. Crall, S. Tanaka, J. Graving, C. Reinders, S. Yadav, J. Banerjee, G. Vecsei, A. Kraft, Z. Rui, J. Borovec, C. Vallentin, S. Zhydenko, K. Pfeiffer, B. Cook, I. Fernández, F.-M. De Rainville, C.-H. Weng, A. Ayala-Acevedo, R. Meudec, M. Laporte *et al.*, "imgaug," https://github.com/aleju/imgaug, 2020, online; accessed 01-Feb-2020.

Frederik E. T. Schöller is pursuing his Ph.D. degree at the Technical University of Denmark, DTU within the area of Object Detection and Deep Learning methods related to autonomous navigation of marine vessels. He graduated as MScEE in 2019 with honors and has already published several conference papers. He received the Young Author Best Paper Award at the IFAC CAMS'2019 conference in Korea.

Lazaros Nalpantidis is Associate Professor of Autonomous systems and Robotics in the Department of Electrical Engineering, Technical University of Denmark (DTU). Before joining DTU, he was an Associate Professor of Cognitive Robotics at Aalborg University Copenhagen, Denmark, where he also served as Head of Section for Sustainable Production within the Department for Materials and Production. He has been a post-doctoral researcher at the Centre for Autonomous Systems (CAS), Computer Vision and Active Perception Lab. (CVAP) of the Royal Institute of Technology (KTH), Sweden. He received his Ph.D. (2010) in Robotic Vision from Democritus University of Thrace, Greece. He holds a M.Sc. (2005) (with Honors) in Electronic Engineering and a B.Sc. (2003) in Physics from Aristotle University of Thessaloniki, Greece.

Mogens Blanke is Professor in Automation and Control at the Technical University of Denmark, DTU. He received the MScEE degree in 1974 and the PhD in 1982 from DTU, was Systems Analyst at the European Space Agency 1975-76, had tenure track positions at DTU 1977-84, was Head of Division at Lyngsø Marine 1985-89 and Professor at Aalborg University 1990–99. He took his present position at DTU in 2000 and served, in addition, as Adjunct Professor at NTNU in Trondheim 2005-2017. His areas of special focus are diagnosis, prognosis, fault tolerant control and autonomous systems. Prof. Blanke served as Technical Editor for IEEE Transactions of Aerospace and Electronic Systems (2006-2016), is currently Associate Editor for Control Engineering Practice and Deputy Editor for Ocean Engineering. He received various international recognitions, latest the ASME DSCD 2018 Rudolf Kalman Best Paper Award and IFAC 2020 Control Engineering Practice Paper Prize. Research leadership included PI for the control system for the Ørsted satellite (1992-99) and PI for the national Danish research effort on surface ship autonomy (since 2017).