Object Detection in Low Resolution Long Wavelength Infrared Images in Maritime Environment

F. M. Verburg

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
University of Twente
Overijssel, Enschede, the Netherlands
f.m.verburg@student.utwente.nl

***Abstract*—Autonomous sailing is still in its early stage, especially compared to autonomous driving. This research addresses the challenge of object detection in maritime environments using long-wavelength infrared (LWIR) images, a critical task for autonomous sailing. While significant progress has been made in object detection for the automotive industry, resulting in numerous datasets and benchmarks, the maritime domain lacks similar research and resources. The maritime environment differs a lot from the automotive environment. Cars have lights to illuminate the roads in the dark and vessels do not. This makes object detection in infrared images a crucial task for autonomous sailing. For detecting objects at large distances (500m +), other sensors like radar and AIS can be used. This research, however, focuses on maritime object detection at close range (0 to 500m) using a low-resolution sensor. The objective is to determine whether inexpensive sensors that produce low-resolution images, requiring minimal processing power, are sufficient for effectively performing this task. To enable this research we created a low-resolution LWIR maritime dataset in the inland area with approximately 5900 images, 6700 vessel labels and 320 buoy labels. Three state-of-the-art object detection models –YOLOv11, Faster R-CNN, and YOLO-FIRI– are evaluated on this dataset to check their ability to detect objects in low-resolution infrared images in maritime environment. Each model is trained on the raw, the colour inverted and the colour inverted + histogram equalized dataset. The results show that the best performing model in terms of recall is Faster R-CNN. The model with the highest mAP@0.5 is YOLOv11 in combination with inversion.**

***Index Terms*—Autonomous sailing, object detection, YOLOv11, Faster-RCNN, dataset, long wave infrared, low resolution**

I. Introduction

The detection of maritime objects plays an important role in autonomous sailing. While sailing on the wide, obstacle-free ocean, a ship can sail on autopilot using only GPS. When a ship enters inland waters or harbors, the amount of passive and active obstacles becomes a lot higher. In order to sail autonomously in these conditions, the autopilot needs to see what is happening around itself, in all weather and lighting conditions. Furthermore, the background on inland waters in much more complex, making the object detection harder. High-resolution visual cameras can be used to detect objects during daylight and most weather conditions, but in the dark their performance drops significantly. To see in the dark, infrared cameras can be used. There are two types of infrared sensors, active and passive. Active sensors require an infrared light source and capture the reflection. Passive infrared sensors rely solely on the infrared radiation emitted by objects. Infrared sensors capture a certain range of infrared wavelengths, Near Infrared (NIR), Short Wavelength Infrared (SWIR), Medium Wavelength Infrared (MWIR), Long Wavelength Infrared(LWIR) and Far Infrared (FIR). All these sensors exist in passive and active variants. Passive MWIR, LWIR and FIR sensors can capture object in the ambient temperature range. The ranges of wavelength each sensor captures, along with the corresponding temperature range, is shown in Table I. Flooding a river with infrared light for detection using an active sensor requires significant power, therefore a passive sensor is desired. The passive IR sensor with the best sensitivity at ambient temperature range is a LWIR sensor.

TABLE I: Infrared sensors and their wavelength and temperature ranges

Sensor	Wavelength [μm]	Temperatures [°C]
NIR	0.75 $\sim$ 1.4	400 $\sim$ 2600
SWIR	1.4 $\sim$ 3	50 $\sim$ 1600
MWIR	3 $\sim$ 8	0 $\sim$ 2500
LWIR	8 $\sim$ 14	0 $\sim$ 1000
FIR	14 $\sim$ 1000	-270 $\sim$ -80

For autonomous sailing, object detection should take place around the entire ship. Multiple infrared sensors are needed to capture a 360° view around the ship. High-resolution infrared sensors are expensive, especially compared to RGB cameras. A high quality thermal camera suited for maritime environment with a resolution of 640x480 costs more than €50.000 whereas a high-resolution, high quality RGB camera costs less than €1.000. Furthermore, processing the high-resolution images of multiple sensors is very resource intensive. For detecting objects at larger distances, other sensors like radar and AIS can be used. For detecting objects in close proximity to the ship, low-resolution LWIR sensors could be sufficient.

Large objects like ships will still be visible at a fair distance because they will take up a large number of pixels. Smaller object like buoys will only become visible when they are closer to the vessel. This is fine as a ship can sail closer to small objects like buoys than to other vessels.

State of the art machine learning models like the single-stage detector YOLO [1] and two-stage detection algorithm Faster R-CNN [2] are widely used in automotive for object detection tasks such as pedestrian detection [3], [4], driver distraction detection [5] and small object detection [6]. Furthermore, research has been done on maritime object detection in the visible light spectrum [7], [8], high-resolution infrared spectrum [9]–[14], and fusion of visible and high-resolution infrared [15], [16]. Preprocessing steps like inversion and histogram equalization show performance improvement in object detection in infrared images [12], [17].

The widespread use of object detection models in the automotive industry has led to a large number of open source datasets [18]–[23]. Since object detection is relatively new in the maritime industry there are not many datasets available. The datasets that are available contain RGB images and high-resolution NIR images [24] or high-resolution LWIR images [25]. There is no publicly available maritime dataset with low-resolution LWIR images captured on inland waters.

While several studies have successfully demonstrated the potential of object detection models on RGB images, high-resolution infrared images in maritime environments, and low-resolution infrared images in non-maritime settings, there remains a gap in applying these models to low-resolution infrared images in maritime environments. No suitable datasets are available to test the models in the mentioned setting and therefore one has to be created. One of the challenges in creating a low-resolution infrared dataset is labeling small objects. It is difficult to see by eye where the outline of the object is, especially when its temperature is close to ambient temperature.

To fill this gap in research, this study aims to achieve the following objectives:

Objective 1 Collect and label a low-resolution long wave infrared dataset in maritime environment.

Objective 2 Analyze the appearance of objects in infrared imagery, including the effects of environmental factors such as reflection and temperature on their visual characteristics.

Objective 3 Evaluate the effectiveness of current state-of-the-art object detection models in detecting vessels and buoys in low-resolution infrared images within maritime environments.

Objective 4 Adapt state-of-the-art object detection models to a different domain to improve object detection performance.

To address these objectives, this research provides the following key contributions:

A low-resolution (160x120px) long wave infrared dataset with 5931 images of maritime environment was created. To label these images a labeling tool was developed by calibrating the low-resolution infrared camera with a high-resolution RGB camera. This calibration is needed to be able to label the objects in the low-resolution infrared images using a high-resolution RGB image of the same scene. The resulting labeled dataset has 6744 ship labels and 321 buoy labels. Images of similar objects in different conditions are compared to see the effect of reflection and temperature on the images. The dataset was divided into five equally sized parts, with each part containing nearly the same number of labels, within a predefined percentage margin of variation. These subsets were used to create five different training, validation and testing splits.

Three object detection models were trained and tested on the five datasets. The models that were evaluated are YOLOv11, Faster-RCNN and YOLO-FIRI. All models are trained on each dataset three times, once without preprocessing, once with inversion, and once with inversion and histogram equalization.

The paper is structured as follows. Section II refers to work related to this research and is split into Object Detection Datasets, Sensor Calibration, Object Detection and Domain Adaptation. Section III explains how the dataset is collected and shows how the custom labeling tool is designed with the calibration between the RGB and IR sensor. How the data is split, preprocessed and trained is explained in section IV. The results are presented in section V followed by the discussion and conclusion in section VI and section VII.

II. Related Work

Object Detection Datasets

One of the first widely used benchmarks for object detection is the PASCAL Visual Object Classes dataset [26]. The first version was introduced in 2007 and it contains about 2500 RGB images for training and 2500 images for validation and testing. The dataset is labeled with 20 classes using bounding boxes. In total more than 12.000 object are labeled. A more recent object detection dataset is Microsoft COCO: common objects in context [27]. COCO is a large-scale object detection, segmentation, and captioning dataset. It has more than 330.000 images and 1.5 million object instances in 80 object categories. It is used to evaluate state-of-the-art object detection models.

The KITTI Vision Benchmark suite [18] contains multiple datasets from various sensors, including stereo visual cameras and Lidar. The 2D object detection benchmark contains 7.481 training and 7.518 test images with a total of 80.256 labeled objects. The KITTI Vision Benchmark is widely used to evaluate object detection models that are used in automotive for autonomous driving. Another dataset for autonomous driving is Nuscenes [19]. It contains data from LIDAR, Camera, IMU and GPS. It contains 1.2 million camera images with several types of human, object and vehicle labels. To test object detection models on infrared images in the automotive industry, FLIR released the Teledyne FLIR ADAS Dataset [20]. It contains 26.442 annotated infrared images with 520.000 bounding boxes across 15 different object categories. The resolution of the images is 640x512 and they are captured with a LWIR sensor.

Shao *et al.* collected a large-scale dataset of ships called SeaShips [28]. It contains 31.455 high-resolution (1920x1080) rgb images and has 40.077 labels of six common ship types. The background of the images is complex as they are recorded by surveillance cameras in inland and coastal environment. The MassMind dataset (Massachusetts Marine INfrared Dataset) [25] contains 2912 RGB and LWIR images in maritime environment. The resolution of the infrared images is 640x512. The dataset does not contain bounding boxes for object detection but each image is segmented with pixel level instance segmentation. Therefore this dataset is not directly usable for object detection. The Singapore Maritime Dataset [24] consists of RGB and NIR labeled images and has 10 classes containing multiple types of ships, buoys, persons and planes. The images are captured on the open water around Singapore and have a resolution of 1920x1080. They are labeled using bounding boxes. Schöller *et al.* created a LWIR dataset in maritime environment [9]. The dataset contains 21.322 images a 640x480 resolution. The images were recorded from a ferry sailing in open water. The VAIS dataset [29] and MARVEL dataset [30] both contain high resolution RGB and IR images, but do not have bounding box labels needed for object detection, as they are datasets for object classification.

Since a lot of research has been done on autonomous driving there are many datasets available for object detection in automotive. For object detection in maritime environment the number of available datasets is much smaller. Especially in inland waters, which have more complex backgrounds, the choice narrows down even further. In the case of low-resolution infrared images in this environment, there are no choices left.

Sensor Calibration

For labeling the low-resolution infrared images a calibration between the IR and RGB sensor is needed. Quite some research has been done on multisensor calibration.

Sher *et al.* [31] calibrated a RGB and infra-red camera using AprilTags made out of different materials glued onto backgrounds of other materials. They used the following combinations of materials: Cardboard-Acrylic, Wood-Vinyl and Metal-Vinyl. The best results were obtained using vinyl AprilTags on metal plates. The paper does not mention the resolution of the sensors but looking at the pictures it is clear that their resolution is much higher then 160x120 pixels. They report a pixel error of ±5 pixels in the y deriction and ±2 pixels in the x direction.

Shibata *at al.* [32] proposed a method for joint geometric camera calibration of visible and low-resolution far-infrared cameras. They designed a calibration target which consists of a low emissivity background and aluminium plates with a high emissivity and reflectivity. The two materials are separated by a thermal insulating layer. This increases the contrast in the thermal images.

Zhang *et al.* [33] proposed a method to find an extrinsic calibration between a LiDAR, RGB camera and thermal camera. The thermal camera has a resolution of 640x512. They use a heated target which is detected by each sensor automatically. For the cameras, the 3D position of the board is determined by decomposing the homograph matrix.

Fu *et al.* [34] developed an algorithm to calibrate stereo, thermal and laser sensors without the need of a target. The thermal sensor used by Fu *et al.* has a much higher resolution than the one used in this research.

The available research shows that a calibration between an infrared sensor and a RGB camera can be done quite accurately. The sensors used in the discussed research are different to the ones used in this research. The resolution of the infrared sensor is lower. With some adaptations to the methods in the discussed research a calibration between the low-resolution infrared sensor and the RGB camera can be made.

Object Detection

A comparative study [9] between Faster R-CNN, YOLOv3 and R-Net on LWIR images of maritime environment with a resolution of 640x480 pixels showed that Faster R-CNN achieved the highest average recall (0.9), while YOLOv3 had the highest precision (0.98). An interesting observation is that the recall of Faster R-CNN for small buoys was 10% higher than for YOLOv3. The better performance on detecting small objects could indicate that Faster R-CNN will perform better when the resolution of the images is lower. Although R-Net scored high precision on all classes, its recall was significantly lower than that of YOLOv3 and Faster-RCNN. The dataset that was used for the evaluation of the models contains images at open water, which have a less complex background than when taken on inland water.

Li *et al.* [14] proposed YOLO-FIRI, a method for infrared object detection based on YOLOv5. It is designed to detect small and weak objects quickly in infrared images. Its performance was tested on the KAIST infrared pedestrian dataset [23], which has 640x512 resolution infrared images. Compared to YOLOv4 they report a mAP increase of approximately 37%.

One challenge faced by object detection models is detecting small objects. Cao *et al.* [6] propose a method to improve the accuracy of detecting small objects in RGB images. They developed a new loss function for the Faster R-CNN network which improves the models capability in detecting small objects. This new loss function could be used to improve the object detection performance in low-resolution infrared images as well, as the object will also be small (few pixels).

Wang *et al.* [12] developed a lightweight ship detection method that is deployed on an embedded device. While their model is slightly outperformed by the YOLOv5 (the most recent YOLO model at the time), in terms of accuracy and recall, their model processes about 60 frames per second, compared to 30 frames per second for the YOLO model.

State-of-the-art object detection models like YOLO and Faster R-CNN show potential in detecting objects in low resolution infrared images, but very little research has been done in maritime environment. Especially when it comes to low-resolution infrared images on inland waters there is a clear gap in the available research.

Domain Adaptation

Due to the lack of available data for object detection in thermal imagery the robustness of the models in this domain is lower than those in the RGB domain. To improve the performance and robustness of the IR object detection models the domain could be adapted to become more similar to the RGB domain.

To improve the performance of object detection in low resolution infrared images, Wang *et al.* [35] applied Contrast Limited Adaptive Histogram Equalization (CLAHE [36]) to partially augment the training data. The model (based on YOLOv5) that was trained on this data outperformed the model that was trained on the non-augmented data.

In an effort to improve accuracy of driver distraction behavior, Liu *et al.* developed CEAM-YOLOv7 [5]. One of the features of this model is the data preprocessing step. They use inversion and CLAHE to adapt the infrared images before training. By only applying this preprocessing step and ignoring the other improvements in their proposed model, the mAP@0.5 increased from 0.612 for YOLOv7 to 0.698 for CEAM-YOLO.

Beyerer *et al.* [37] propose a strategy to use CNN-based object detection frameworks, which are pretrained on RGB images, by transforming IR images as close as possible to the RGB domain. They evaluate the performance of a person detection model trained on RGB data in detecting persons in thermal images, comparing the results across several tested preprocessing steps. The images processed with inversion and histogram equalization achieved the best results.

A method proposed by Guo *et al.* [38] uses domain adaptation to augment the training data for pedestrian detection models. They do this by adapting the widely available labeled RGB data to synthetic IR data and augmenting this new data to the RGB data. A pedestrian detection model is trained on the combined data. According to their results their method reduces the log-average miss rate by 12%.

The discussed research shows that domain adaptation methods can be used to improve the performance of object detection models in the IR domain. Several studies report that inversion and histogram equalization have a positive effect on the performance of the models.

III. Dataset Creation

To create a properly labeled, low resolution infrared dataset, several steps were taken. In Section III-A, the hardware and data collection is described. The tool that was developed to annotate the data is described in Section III-B. How the data was labeled is described in Section III-C

A. Hardware Setup and Data Collection

The data collection system consisted of two sensors, a FLIR Lepton 3.5 (IR) and a Sony IMX390 (RGB), mounted together on a stand. Both sensors were connected to a NVidia Jetson through USB and recorded using a custom ROS2 [39] system.

The FLIR Lepton 3.5 is a passive LWIR sensor which has a resolution of 160x120 pixels, a horizontal field of view of 57°, pixel size of $12\mu m$ and records at 9 frames per second. It can record in RGB and grayscale. This dataset is recorded in grayscale. Its spectral range is $8 \sim 14\ \mu m$. The Sony IMX390 has a resolution of 1920x1080, a horizontal field of view of 59.8° and records at 60 frames per second. The IR sensor was put in a waterproof box and the RGB camera was mounted to this box with a custom 3D print. The sensor combination was mounted to a stand which can easily be moved around and secured to a vessel. Both sensors captured at their maximum frame rate and later, the images were synchronized by matching the closest microsecond timestamp of the RGB images to each IR image.

The data acquisition was done over the course of two days. The first day, data was recorded at several locations from shore in Dordrecht, Papendrecht and Zwijndrecht. The exact locations are shown in Fig. 1. The recording locations were chosen in such a way that many different backgrounds and lighting conditions were captured. The recordings were made between 11:00 and 15:00 on the 4th of September 2024. The air temperature was 21°C and it was cloudy.

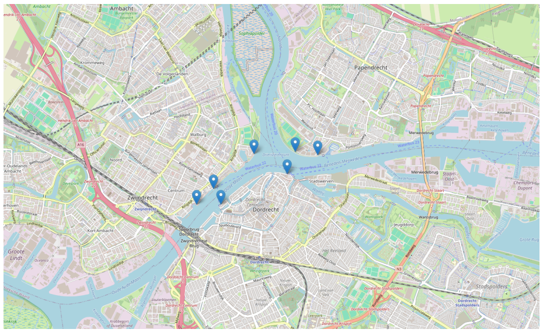


Fig. 1: Shore recording locations

The second day, the sensors were mounted on the bow of a vessel. The vessel sailed the route shown in Fig. 2. Several harbours were entered to collect data of docked vessels. Furthermore, vessels and buoys were recorded while approaching them from several angles. These recordings were made between 9:00 and 15:00 on the 18th of September 2024. During the first half of the recording it was cloudy and misty, during the second half of the recording it was sunny. The temperature started at 17 °C and went up to 23 °C at the end of the recording.

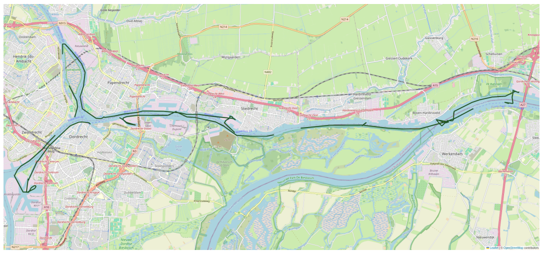
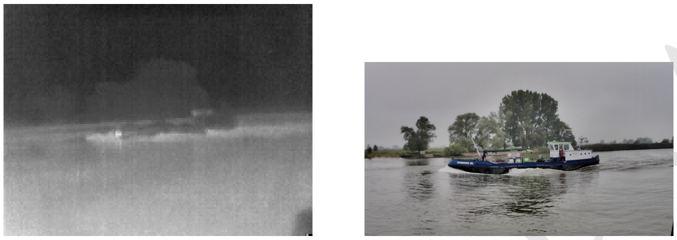


Fig. 2: Sailing route recording vessel

B. Data Annotation Tool

Annotating low resolution infrared images is challenging. When an object is close to the infrared sensor it can be easily distinguished, as is shown in Fig. 3a. The ship is clearly visible and can be labeled easily.

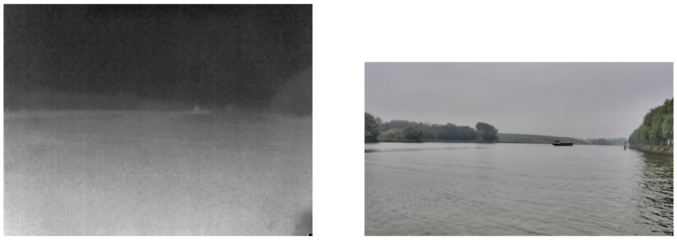


a. Object close to IR sensor

b. Object close to RGB sensor

Fig. 3: Example of IR and RGB image with object near sensors

When objects are further away from the sensor, it becomes very hard to see the object because of the very low resolution of the infrared sensor. In Fig. 4a an example of a ship further away from the sensor is shown. As you can see, it is very hard to detect the ship by only looking at the infrared image. It makes the labeling of these images impossible.



a. Object far from IR sensor

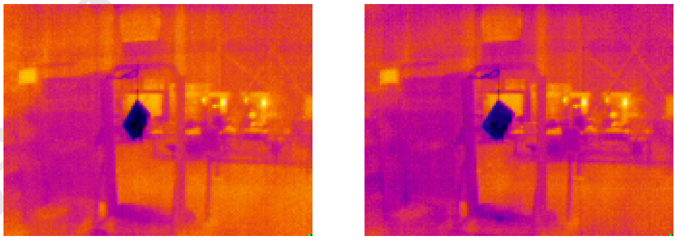
b. Object far from RGB sensor

Fig. 4: Example of IR and RGB image with object far from sensors

To be able to label the data of all recorded infrared images, we need to calibrate the RGB camera to the IR camera. When this is done, we can use the higher resolution RGB image to determine which pixels in the low resolution IR image belong to an object that needs to be labeled.

To calibrate the RGB camera to the thermal camera a calibration target was designed. This target needs to be visible by both sensors. For it to be visible by the thermal camera, it needs to be cooled or heated to have a different temperature than the ambient temperature. Furthermore, the material should have a low reflectivity so it does not reflect ambient radiation, and a high emissivity so it is clearly visible by the sensor. Acrylic plastic has a high emissivity and low reflectivity when coloured in a matt colour.

A test was conducted to test the visibility of cooled acrylic plastic. The acrylic was cooled in a pool with a water temperature of 17 °C for five minutes. Then is was suspended from a rope and infrared images were recorded every minute for fifteen minutes. The image after 1 minute can be seen in Fig. 5a and the image after 15 minutes can be seen in Fig. 5b. The acrylic is still clearly visible after 15 minutes, which is enough time to collect the data needed for the calibration of the two sensors.



a. Cooled acrylic after 1 min

b. Cooled acrylic after 15 min

Fig. 5: Visibility test of cooled acrylic

The shape of the target had to be chosen in such a way that its key features could be easily distinguishable with both sensors. A large diamond shaped board was chosen, each corner of the board can clearly be seen by both sensors. The board has four holes in a square pattern. In future work, these holes can be used to determine the 3D location of the target by using homograph matrix decomposition, as done by Zhang *et al.* [40].

Several recordings of the calibration board in various positions were captured with both sensors. For each pair of RGB and IR image the matching features of the calibration target were selected manually, as can be seen in Fig. 6.

By doing this for all pairs of images we get a list of matching sets containing RGB points and IR point. The RGB points are considered as source points eq. (1) that need to be transformed to the IR points, which we will consider as destination points eq. (2).

$$\text{Source points: } (x_i, y_i) \text{ for } i = 1, 2, \ldots, n \tag{1}$$

$$\text{Destination points: } (x'_j, y'_j) \text{ for } j = 1, 2, \ldots, n \tag{2}$$

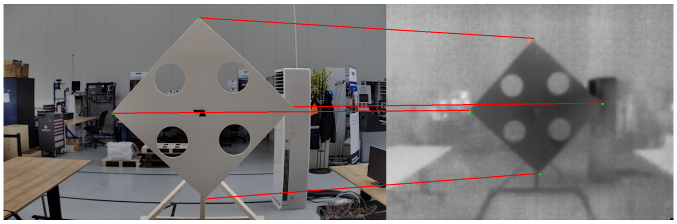


Fig. 6: Point Matching Example

The goal is to find the affine transformation matrix M eq. (3).

$$M = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \end{bmatrix} \tag{3}$$

When the affine transformation is applied to point (x_i, y_i) we get the transformed point (x'_i, y'_i) by using eq. (4).

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} m_{02} \\ m_{12} \end{bmatrix} \tag{4}$$

This tranformation matrix M is estimated by using the estimateAffine2D function of OpenCV [41]. This function finds the matrix by minimizing the sum of square differences between source and destination points. The function is shown in eq. (5).

$$E = \sum_{i=1}^{n} \begin{aligned} &(x'_i - (m_{00} \cdot x_i + m_{01} \cdot y_i + m_{02}))^2 \\ &+ (y'_i - (m_{10} \cdot x_i + m_{11} \cdot y_i + m_{12}))^2 \end{aligned} \tag{5}$$

C. Labeling

The transformed RGB images can now be used to label the low-resolution infrared images. A custom labeling tool was developed to make the task as efficient as possible. The labeling tool is based on work done by Cartucho *et al.* [42]. When a bounding box is drawn around an object in the transformed RGB image, the bounding box is copied to the IR image and vice versa. This way, the tool could also be used to label RGB images in low lighting conditions. The images are labeled with two classes: ship and buoy. All motorized vessels are labeled as ship. Small boats like rowing boats are not labeled. Many different types of buoys exist, in this dataset they are all labeled equally.

IV. Object Detection

This section describes how the object detection models are evaluated. In Section IV-A the splitting of the data is described. Section IV-C describes the used models in more detail. Section IV-D explains how the models were trained and evaluated.

A. Data Splitting

To assure reliable, generalizable and bias-free performance of the models, the dataset needs to be split correctly. This means that the class distribution should be consistent across the train, validation and test sets. Unfair splits may result in meaningless performance metrics or overfitting. To get a better sense of the true performance of the model, we want to do cross-validation. To achieve this, the dataset was divided into five equally sized parts with each part containing about the same number of labels within a predefined percentage margin of variation. This way we can train and test the models on 5 different combinations of the subsets.

The data was randomly split into 5 subsets until the amount of images, vessel and buoy-labels were within a certain percentage among all subsets. The amount of images in each subset lie within 10% difference of each other. For the vessel labels the percentage is 20% and for the the buoys the percentage is 40%.

B. Preprocessing

We apply two preprocessing steps that have shown performance improvement in object detection models on infrared datasets in previous research [5], [35].

The first preprocessing step is inversion which inverts greyscale of the IR image. The dark parts of the image become bright and vice versa. This results in an image that looks more similar to a greyscale visual image.

The second preprocessing step is the adaptive histogram equalization, more specifically, Contrast Limited Adaptive Histogram Equalization (CLAHE) [43]. Histogram equalization is a contrast enhancement method which helps making objects and features more distinguishable from the background. Infrared images can suffer from low contrast due to the limited intensity variations. This happens when objects have a similar temperature as their background. Histogram equalization redistributes the intensity values which makes subtle differences more visible. Traditional histogram equalization enhances the entire image uniformly. This can result in unwanted effects like noise amplification. To prevent this, CLAHE is used. This method has a contrast limited feature which prevents over-enhancement in areas with relatively uniform intensity.

C. Models

Three models are trained and tested: the newest YOLO (v11), Faster-RCNN and YOLO-FIRI. The first two have been selected because they show great results on the MS COCO dataset [27] and previous research on infrared object detection, and because they are different network types. YOLO is a one-stage detector, while Faster-RCNN is a more classical two-stage detector, which usually gives better accuracy and recall, but is a bit slower in performing the detection. However, as long as the model can process 9 frames per second it is good enough. Previous research has shown that Faster R-CNN is capable of processing 2 frames (640x480 pixels) per second on an outdated machine. Much higher processing speeds are expected with the lower resolution on newer machines. YOLO-FIRI was selected because it is customized specifically for infrared images. Each model is described in more detail later.

The models are evaluated on Precision (P), Recall (R), mean Average Precision at an IoU threshold of 0.5 (mAP@0.5), Average Precision for vessels (AP Vessel) and Average Precision for buoys (AP Buoy). For autonomous sailing a high recall is more important than high precision because it is more important to detect all objects than it is to correctly classify the objects. The Precision and Recall are calculated using the True Positives (TP), False Positives (FP) and False Negatives (FN). A prediction is classified as True positive when at least 50% of the bounding box overlaps with the annotation. If a prediction is made when no object is present it will be classified as False Positive. When a object is present and it is not predicted it is a False Negative. The Precision and Recall are calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

For each class we can calculate the Average Precision (AP), which is the area under the Precision-Recall (PR) curve. The Precision and Recall are computed at multiple confidence thresholds and this results in the PR-curve. For each class the AP is calculated with the formula shown in eq. (8).

$$AP = \int_0^1 \text{Precision}(\text{Recall})\, d\text{Recall} \tag{8}$$

Since we only have two classes we can easily calculate the mean Average Precision (mAP@0.5) with te equacation shown in eq. (9). The performance of the model is better when the mAP is higher.

$$mAP = \frac{AP_{\text{vessel}} + AP_{\text{buoy}}}{2} \tag{9}$$

YOLOv11 [44]
The first object detection model that is evaluated is the latest YOLO model. A mentioned before, YOLO is a one-stage detector, which means that every input image is passed through the network a single time. Version 11 is Ultralitics latest iteration of real-time object detectors. They claim that their newest iteration achieves a higher mAP on the COCO dataset while using fewer parameters than YOLOv8. Several model variant with different sizes are available: nano, small, medium, large and extra large. For this research, the large pre-trained model is chosen. It was trained on the COCO dataset which includes 80 pre-trained classes, including ”boat” which is also a class in this research.

Faster R-CNN
The second model that is tested is the two stage detector Faster R-CNN. Previous research has shown that it usually has a higher recall than other models. To train this model, detectron2 [45] is used. Similar to YOLO, for Faster R-CNN there are several different pre-trained models available to choose from. These models are also trained on the COCO dataset. There are three backbone combinations to choose from, FPN, C4 and DC5. FPN stands for Feature Pyramid Network and it is a feature extractor which allows the model to detect objects of various sizes more effectively. This is very helpful for this research since the dataset contains ships of all sizes. The FPN backbone combinations are recommended by Detectron2 because they it obtains the best speed/accuracy trade-off. For the training the R101-FPN is chosen. This model has 101 layers and should give the highest performance out of all models pre-trained on the COCO dataset.

YOLO-FIRI [14]
The last model that is evaluated is YOLO-FIRI, a model that is based on YOLOv5 and developed by Li *et al.* to perform better on infrared images. They expanded and iterated the cross-stage-partial connections module in the early layers to maximize the use of shallow features. Furthermore, they introduced an attention module that focuses on the objects and suppresses the background. The model was tested on the KAIST dataset and compared to YOLOv4. They report an increase in mAP from 81% to 98.3%.

D. *Hardware and Training*

These three object detection models are evaluated on the newly created dataset and the preprocessing variations. The models are trained on a laptop with a intel i7-12850HX processor, 32.0 GB Ram and NVIDIA RTX A4500 16 GB GPU.

V. Experiments & Results

A. *Dataset Creation*

In total, about 35.000 IR and 175.000 RGB images were collected during the two days of recording in about 250 individual recordings. Each recording has a different scenario which can be an empty river, a single ship or buoy, or a combination of buoys and ships. Out of each recording 25 random IR images were selected. For each IR image the RGB image with the timestamp closest to the timestamp of the IR image was selected. The matching images were passed to the labeling tool for annotation.

In order to use the labeling tool the IR and RGB camera are calibrated. To calibrate the two cameras 15 images of the calibration target are collected. Between each recording the calibration board was moved to a different position. For each matching pair of images the four corners of the target are manually selected. This results is two sets of pixel coordinates, each containing 60 coordinates.

$$\text{IR points: } \{(29, 59), (64, 20), (103, 56), ...\} \tag{10}$$

$$\text{RGB points: } \{(340, 542), (760, 66), (1242, 512), ...\} \tag{11}$$

By using the OpenCV function *EstimateAffine2D* we find the transformation matrix between the IR points and RGB points. The resulting transformation matrix between the IR and RGB sensor is shown in eq. (12).

$$M = \begin{bmatrix} 0.989 & 0.004 & 4.009 \\ -0.010 & 0.742 & 58.449 \end{bmatrix} \tag{12}$$

To check if the transformation matrix is accurate, a RGB image was transformed and laid over the corresponding IR image. The RGB images was transformed using the OpenCV function *WarpAffine*. The result can be found in Fig. 7.

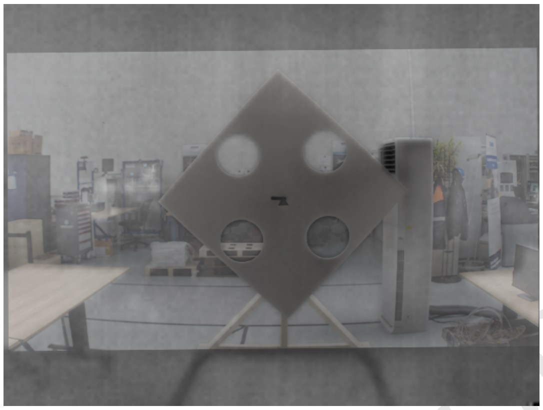
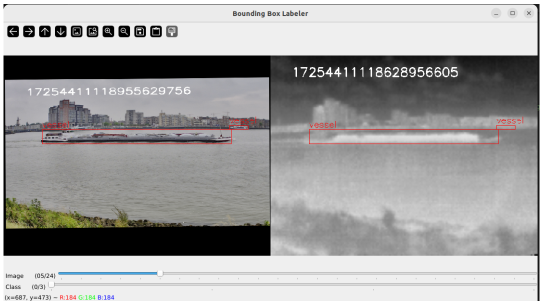


Fig. 7: Blended IR and transformed RGB image

The target in the transformed RGB image matches the target in the infrared image. This means that if the target would be labeled in the transformed RGB image, the same label could be used for the target in the infrared image. This is what is done in the Dual Labeling Tool. A screenshot of the tool is shown in Fig. 8. The red bounding box was drawn around the ship visible in the transformed RGB image and automatically copied to the IR image.



Bounding Box Labeler

Image (05/24)

Class (0/3)

(x=687, y=473) ~ R:184 G:184 B:184

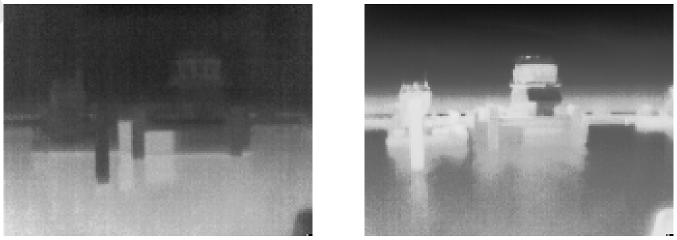
Fig. 8: Dual Labeling Tool

The tool is used to label a random selection from the recorded dataset. The resulting dataset has the following properties:

- Amount of images: 5931
- Amount of ship labels: 6744
- Amount of buoy labels: 321

B. Object Appearance in Different Conditions

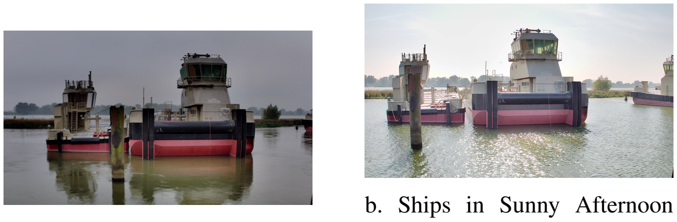
To see the effects of the ambient temperature and lighting on the visibility of object an experiment was conducted. Two docked ships were captured with the infrared camera in two different conditions. The first image was captured about an hour after sunrise on a cold, calm, misty morning. The ambient temperature was about 17 °C. The second image was captured in the afternoon while it was sunny and more windy than in the morning. The ambient temperature was about 23 °C. The recorded images can be seen in Fig. 9a and Fig. 9b, their corresponding rgb images can be found in Fig. 10a and Fig. 10b. The difference in environment conditions can clearly be seen in the rgb images. While the ambient temperature between the two images only differs 6 °C they look very different. The ships are more easily distinguishable in Fig. 9b because of the higher contrast created by the bigger differences in temperature between the water and ships, and the sky and the ships. Furthermore, some features of the ships are more visible. One of the negative effects visible on the sunny image is the reflection of the ships on the water created by the sun. It is difficult to see where the waterline of the ship is. In Fig. 9a you can clearly see this waterline.



a. Ships in Misty Morning IR

b. Ships in Sunny Afternoon IR

Fig. 9: Visibility Ships in Different Temperatures IR



a. Ships in Misty Morning RGB

b. Ships in Sunny Afternoon RGB

Fig. 10: Visibility Ships in Different Temperatures RGB

C. Object Detection

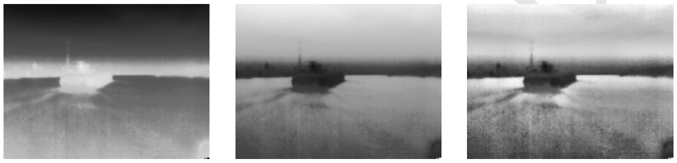
1) Data Splitting: The data is split into 5 random but comparable combinations. This is done by randomly splitting the 250 recordings into five sets and counting the amount of images, vessels and buoys in each set. If the amount of images in each set is with 10% difference of the other sets it is considered evenly split. The same check is done with the amount of vessel labels (20%) and buoy labels (40%). These five sets are combined in five different configuration where one set is reserved for validation, one for testing and the rest for training. The content of each resulting dataset split can be found in table II. Each set has similar amounts of images, vessels and buoys in their train, validation and test subset.

TABLE II: Specifications of Five Dataset Splits

Dataset	Nr. of Images			Nr. of Vessels			Nr. of Buoys		
	Train	Val	Test	Train	Val	Test	Train	Val	Test
Split 1	2707	885	891	3089	1079	950	153	58	44
Split 2	2687	911	885	3004	1035	1079	141	56	58
Split 3	2657	915	911	3034	1049	1035	144	55	56
Split 4	2687	881	915	3064	1005	1049	158	52	55
Split 5	2711	891	881	3163	950	1005	169	44	42

2) Preprocessing: The images in all resulting datasets are preprocessed using two methods. Firstly, the images are inverted by using the OpenCV function *bitwise_not*. After that, contrast limited adaptive histogram equalization is applied by using OpenCV function *createCLAHE*.

In Fig. 11 an image with each preprocessing step is shown. Fig. 11a shows the image captured by the infrared sensors, Fig. 11b shows the inverted image and Fig. 11c shows the image that is inverted and has histogram equalization applied to it (HE). All splits were preprocessed which results in three sets for each split. In total there are 15 sets that each model is trained and evaluated on.



a. Raw image

b. Inverted image

c. Inverted & HE

Fig. 11: Same image with each preprocessing step

3) Trained Models Performance: Each model is trained on the 15 sets. Since the YOLO-FIRI model is based on YOLOv5 it has similar parameters as the YOLOv11. The parameters are set equally to get a comparable result. The YOLOv11 and YOLO-FIRI models are trained starting with a pretrained model. For YOLOv11 the *yolov11l.pt* file is used as starting weights and for YOLO-FIRI *yolov5l.pt* is used. These files are chosen because they offer a good balance between performance and training time. The YOLO models are trained for 1500 epochs with a patience of 150 epochs. This means that the model will stop the training early if it does not notice any improvement for the last 150 epochs. The Faster R-CNN model is trained using Detectron2. When training the Faster R-CNN model you cannot specify the amount of epoch but you need to specify the amount of iterations. For this model 10000 iterations are chosen.

After training the models they are tested on the part of the dataset reserved for testing. Fig. 12 shows examples of the detection results from all models and in combination with the preprocessing steps. The first row of images are the raw images captured by the IR sensor. The images in the middle row are inverted. The images in the last row have been inverted and CLAHE has been applied. All models were able to detect the ship. Faster R-CNN has predicted a false positive in the raw image. YOLO-FIRI has predicted more false positives. On the image that is inverted and has histogram equalization applied, YOLOv11 has predicted the vessel with a bounding box that is too large.

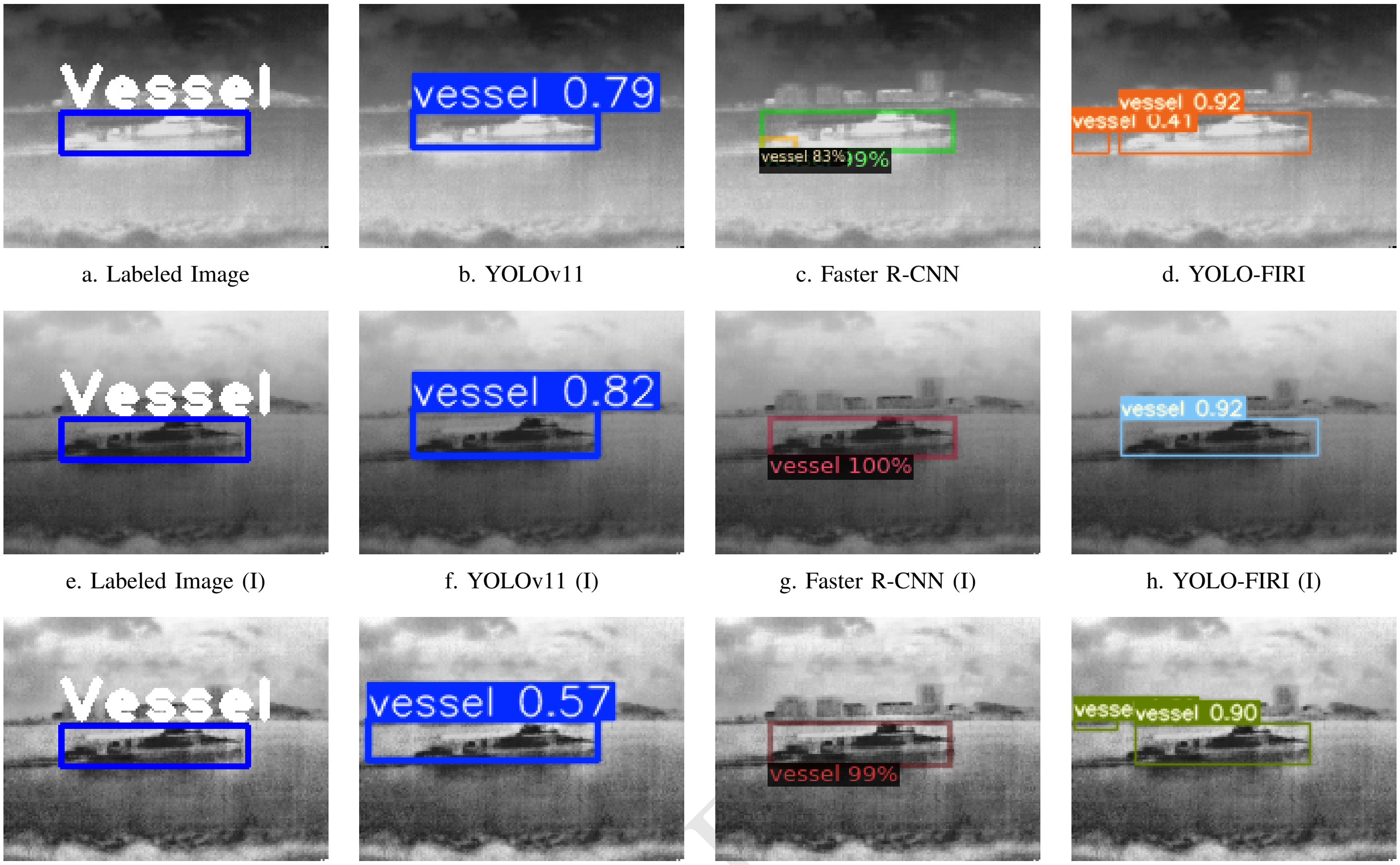
In table III we find the results of the trained models. The first column states the model, the second column states the preprocessing step that is applied to the the data, no preprocessing (-), inversion (I) or inversion and histogram equalization (I+HE). For each model the following evaluation metrics are shown: Precision (P), Recall (R), mean Average Precision at an IoU threshold of 0.5 (mAP@0.5), Average Precision for vessels (AP Vessel) and Average Precision for buoys (AP Buoy). Each model is trained on the five different splits. The average score for each metric is calculated, along with the standard deviation. The highest score for each metric is written in bold.

There are a couple of observations in this table that stand out. Firstly, all models struggle in getting high scores. The best performing model when it comes to mAP@0.5 is YOLOv11 in combination with inversion and it only scores 32.8%. It also has the highest Average Precision on both vessels and buoys over a range of confidence scores. When it comes to recall, which is a very important metric in autonomous sailing, Faster R-CNN comes out on top. This is in line with the results found in the research discussed in the related work (section II). The standard deviation across the different splits is lower for the Faster R-CNN model than for the YOLO models, especially when looking at the precision. Faster R-CNN clearly struggles in detecting the buoys which leads to the low precision. As mentioned in the related work (section II), previous work showed that Faster R-CNN outperformed YOLOv3 in detecting small buoys. This is clearly not the case in this research.

Inverting the infrared images has positive effect on the average precision for the YOLO models. It has a negative effect on the Faster R-CNN model. Inversion + Histogram Equalization has a negative effect on the performance of all models.

VI. Discussion

Although the dataset collected over the course of two days contains a lot of different kind of ships, it only contains a handful of buoys. It is clear that the object detection models struggle with detecting buoys. There are several possible explanations for this poor performance. Firstly, the most obvious reason, is that there is much less data available for the models



a. Labeled Image b. YOLOv11 c. Faster R-CNN d. YOLO-FIRI

e. Labeled Image (I) f. YOLOv11 (I) g. Faster R-CNN (I) h. YOLO-FIRI (I)

i. Labeled Image (I+HE) j. YOLOv11 (I+HE) k. Faster R-CNN (I+HE) l. YOLO-FIRI (I+HE)

Fig. 12: Vessel detection by YOLOv11, Faster R-CNN and YOLO-FIRI on image with different preprocessing steps

TABLE III: Performance of YOLOv11, Faster R-CNN and YOLO-FIRI on Custom Low Resolution Maritime Infrared Dataset with 3 Preprocessing Variations

Model	Preprocessing	P	R	mAP@0.5 (%)	AP Vessel (%)	AP Buoy (%)
YOLOv11	-	**0.610±0.230**	0.270±0.083	0.301±0.088	0.396±0.045	0.206±0.172
YOLOv11	I	0.520±0.135	0.292±0.064	**0.328±0.063**	**0.429±0.019**	**0.229±0.134**
YOLOv11	I + HE	0.581±0.242	0.251±0.079	0.270±0.067	0.397±0.046	0.143±0.159
Faster R-CNN	-	0.364±0.049	**0.386±0.021**	0.275±0.089	0.430±0.031	0.120±0.079
Faster R-CNN	I	0.362±0.038	0.383±0.032	0.236±0.035	0.404±0.034	0.068±0.038
Faster R-CNN	I + HE	0.381±0.035	0.367±0.016	0.244±0.042	0.409±0.034	0.078±0.041
YOLO-FIRI	-	0.314±0.099	0.329±0.093	0.229±0.040	0.396±0.011	0.063±0.088
YOLO-FIRI	I	0.363±0.202	0.330±0.049	0.268±0.091	0.403±0.031	0.132±0.191
YOLO-FIRI	I + HE	0.395±0.082	0.314±0.097	0.260±0.054	0.406±0.023	0.113±0.129

to learn from. A simple, cheap method to resolve this problem is to copy the data containing buoys several times such that the amount of buoy labels comes closer to the amount of vessel labels. However, this is not a great solution as the models will not get new data to learn from and the risk of overfitting might increase.

A second explanation for the poor performance of buoy detection is that buoys are much smaller than vessels. With the low resolution IR sensor it becomes very difficult to detect small objects. When a buoy is far away from the sensor it only takes up a couple of pixels in the image. To see if this hypothesis is correct a test could be performed where bounding boxes with an area below a certain threshold are discarded. This will remove mostly buoys labels, but it will probably have less trouble detecting the remaining buoys.

A third explanation is that buoys are passive objects which do not generate heat. Moving vessels with their engine running generate heat are more easily visible by the thermal camera.

Since the dataset was captured on only two different days it does not contain a lot of different weather and lighting conditions. To improve the performance of the models it is best to extend the dataset with images captured during different moments in the year for all weather conditions, and on different moments during the day and night to get more lighting conditions and thermal variations.

Previous work showed that Faster R-CNN outperformed YOLO (version 3) in detecting small objects. This is not what is observed in this research. The resolution of the images in this research is much smaller, which can be an explanation. A more logical explanation is that YOLO was developed further up to version 11 and it might now simply be better at detecting small objects than Faster R-CNN.

The preprocessing steps taken in an effort to improve the performance of the models did not have the expected effect. In previous work inversion and histogram equalization improved the performance of the model, but here we see another result. For YOLOv11 the inversion improved the performance by about 10% when it comes to Average Precision at a IoU of 0.5. For YOLO-FIRI the performance improves by about 15%. In this research the models were exclusively trained on one of the preprocessing steps. Training the models on a combination of all three datasets might improve the performance of the models.

VII. Conclusion

In this paper the performance of three state-of-the-art object detection models are evaluated on their ability to detect vessels and buoys on low-resolution long wavelength infrared images in maritime environment. To achieve this goal four objectives were established. Firstly a method to collect and label a low-resolution LWIR dataset in maritime environment is proposed. To evaluate the models a dataset was collected and labeled with the help of a custom labeling tool which relies on a calibration between a RGB and IR sensor. The resulting dataset contains 5900 images, 6700 vessel labels and 320 buoy labels. The effect of ambient temperature, lighting and reflection on the appearance of objects was observed in the collected data. Objects that look similar in the visual domain can look vastly different in the infrared domain when the temperature and water state differs. Three state of the art object detection models, YOLOv11, Faster R-CNN and YOLO-FIRI, are evaluated on the dataset with different preprocessing steps (raw, inverted and inverted + histogram equalization). Faster R-CNN achieved the highest recall (0.386), which is the most important performance metric in autonomous sailing. YOLOv11 has the highest mAP@0.50 when combined with inversion. Inversion has a positive effect on the performance of the YOLO models but a negative effect on Faster R-CNN.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December. IEEE Computer Society, 12 2016, pp. 779–788.

[2] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[3] X. Zhao, W. Li, Y. Zhang, A. Gulliver, S. Chang, and Z. Feng, "A faster rcnn-based pedestrian detection system," in *IEEE Vehicular Technology Conference*. Institute of Electrical and Electronics Engineers (IEEE), 7 2016.

[4] K. N. R. Chebrolu and K. Kumar, "Deep learning based pedestrian detection at all light conditions," in *IEEE International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 4 2019.

[5] S. Liu, Y. Wang, Q. Yu, H. Liu, and Z. Peng, "Ceam-yolov7: Improved yolov7 based on channel expansion and attention mechanism for driver distraction behavior detection," *IEEE Access*, vol. 10, pp. 129 116–129 124, 2022.

[6] C. Cao, B. Wang, W. Zhang, X. Zeng, X. Yan, Z. Feng, Y. Liu, and Z. Wu, "An improved faster r-cnn for small object detection," *IEEE Access*, vol. 7, pp. 106 838–106 846, 2019.

[7] J. H. Kim, N. Kim, Y. W. Park, and C. S. Won, "Object detection and classification based on yolo-v5 with improved maritime dataset," *Journal of Marine Science and Engineering*, vol. 10, 3 2022.

[8] J. Jiang, X. Fu, R. Qin, X. Wang, and Z. Ma, "High-speed lightweight ship detection algorithm based on yolo-v4 for three-channels rgb sar image," *Remote Sensing*, vol. 13, 5 2021.

[9] F. E. 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S240589631932169X

[10] M. Kristo, M. Ivasic-Kos, and M. Pobar, "Thermal object detection in difficult weather conditions using yolo," *IEEE Access*, vol. 8, pp. 125 459–125 476, 2020.

[11] J. Zhou, B. Zhang, X. Yuan, C. Lian, L. Ji, Q. Zhang, and J. Yue, "Yolo-cir: The network based on yolo and convnext for infrared object detection," *Infrared Physics and Technology*, vol. 131, 6 2023.

[12] L. Wang, Y. Dong, C. Fei, J. Liu, S. Fan, Y. Liu, Y. Li, Z. Liu, and X. Zhao, "A lightweight cnn for multi-source infrared ship detection from unmanned marine vehicles," *Heliyon*, vol. 10, 2 2024.

[13] S. Nirgudkar and P. Robinette, "Beyond visible light: Usage of long wave infrared for object detection in maritime environment," in *2021 20th International Conference on Advanced Robotics, ICAR 2021*. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1093–1100.

[14] S. Li, Y. Li, Y. Li, M. Li, and X. Xu, "Yolo-firi: Improved yolov5 for infrared image object detection," *IEEE Access*, vol. 9, pp. 141 861–141 875, 2021.

[15] S. Moosbauer, D. Konig, J. Jakel, and M. Teutsch, "A benchmark for deep learning based object detection in maritime environments," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, vol. 2019-June. IEEE Computer Society, 6 2019, pp. 916–925.

[16] F. Farahnakian and J. Heikkonen, "Deep learning based multi-modal fusion architectures for maritime vessel detection," *Remote Sensing*, vol. 12, 8 2020.

[17] C. Herrmann, M. Ruf, and J. Beyerer, "Cnn-based thermal infrared person detection by domain adaptation," in *Defense + Security*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:43927421

[18] A. Geiger, P. Lenz, and R. Utrasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 6 2012.

[19] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "Nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2020, pp. 11 618–11 628.

[20] Teledyne FLIR, "Adas dataset for automotive applications," https://www.flir.com/oem/adas/adas-dataset-form/, 2024, accessed: 2024-10-15.

[21] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[22] J. Mao, M. Niu, C. Jiang, H. Liang, X. Liang, Y. Li, C. Ye, W. Zhang, Z. Li, J. Yu, H. Xu, and C. Xu, "One million scenes for autonomous driving: ONCE dataset," *CoRR*, vol. abs/2106.11037, 2021. [Online]. Available: https://arxiv.org/abs/2106.11037

[23] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in *Conference*

on Computer Vision and Pattern Recognition, 2015. [Online]. Available: http://rcv.kaist.ac.kr/multispectral-pedestrian/

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

[25] S. Nirgudkar, M. DeFilippo, M. Sacarny, M. Benjamin, and P. Robinette, "MassMIND: Massachusetts Marine INfrared Dataset."

[26] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2009.

[27] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll'a r, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[28] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, "Seaships: A large-scale precisely annotated dataset for ship detection," *IEEE transactions on multimedia*, vol. 20, no. 10, pp. 2593–2604, 2018.

[29] M. M. Zhang, J. Choi, K. Daniilidis, M. T. Wolf, and C. Kanan, "Vais: A dataset for recognizing maritime imagery in the visible and infrared spectrums," *CVPR*, 2015.

[30] E. Gundogdu, B. Solmaz, V. Yucesoy, and A. Koc, "Marvel: A large-scale image dataset for maritime vessels," in *Computer Vision - ACCV*, 2016, pp. 165–180.

[31] B. Sher, X. Xu, G. Chen, and C. Feng, "Marker-based extrinsic calibration for thermal-rgb camera pair with different calibration board materials," in *International Symposium on Automation and Robotics in Construction*, 2023.

[32] T. Shibata, M. Tanaka, and M. Okutomi, "Accurate joint geometric camera calibration of visible and far-infrared cameras," in *IS and T International Symposium on Electronic Imaging Science and Technology*. Society for Imaging Science and Technology, 2017, pp. 7–13.

[33] J. Zhang, Y. Liu, M. Wen, Y. Yue, H. Zhang, and D. Wang, "L2v2t2calib: Automatic and unified extrinsic calibration toolbox for different 3d lidar, visual camera and thermal camera," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, 2023, pp. 1–7.

[34] T. Fu, H. Yu, W. Yang, Y. Hu, and S. Scherer, "Targetless extrinsic calibration of stereo, thermal, and laser sensors in structured environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, 2022.

[35] J. Wang, H. Wang, and A. Wu, "Enhanced small target recognition with lightweight yolov5 in low-res images," in *2024 16th International Conference on Advanced Computational Intelligence, ICACI 2024*. Institute of Electrical and Electronics Engineers Inc., 2024, pp. 9–12.

[36] S. Pizer, R. Johnston, J. Ericksen, B. Yankaskas, and K. Muller, "Contrast-limited adaptive histogram equalization: speed and effectiveness," in *[1990] Proceedings of the First Conference on Visualization in Biomedical Computing*, 1990, pp. 337–345.

[37] J. Beyerer, M. Ruf, and C. Herrmann, "Cnn-based thermal infrared person detection by domain adaptation." SPIE-Intl Soc Optical Eng, 5 2018, p. 8.

[38] T. Guo, C. P. Huynh, and M. Solh, "Domain-adaptive pedestrian detection in thermal images," in *International Conference on Image Processing*. IEEE, 2019, pp. 1660–1664.

[39] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074

[40] J. Zhang, R. Zhang, Y. Yue, C. Yang, M. Wen, and D. Wang, *SLAT-Calib: Extrinsic Calibration between a Sparse 3D LiDAR and a Limited-FOV Low-resolution Thermal Camera*. 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2019, this paper has information about the calibration board we are designing. It uses the method with four holes in the board. They use it to calibrate a Low resolution IR Camera (382x288) and a sparse 3D Lidar.

[41] Itseez, "Open source computer vision library," https://github.com/itseez/opencv, 2015.

[42] J. Cartucho, R. Ventura, and M. Veloso, "Robust object recognition through symbiotic deep learning in mobile robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2336–2341.

[43] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. T. H. Romeny, J. B. Zimmerman, A. Zuiderveld, and A. Ziekenhuis, "Adaptive histogram equalization and its variations," pp. 355–368, 1987.

[44] G. Jocher and J. Qiu, "Ultralytics yolo11," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[45] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.