

## TARGET DETECTION FOR VISUAL COLLISION AVOIDANCE SYSTEM

**Miro Petković, Danko Kezić,  
Igor Vujović, Ivan Pavić**  
*University of Split (Croatia)*

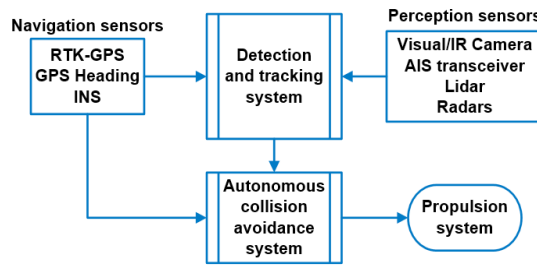
**Abstract.** Automatic Identification Systems (AIS) and Automatic Radar Plotting Aids (ARPA) are commonly used to detect targets for collision avoidance. However, AIS cannot detect targets without AIS transmitters and ARPA has limitations due to blind sector and small targets may not be detected. Advances in computer performance and video-based detection generated much interest in developing intelligent video surveillance systems to achieve autonomous navigation. To develop a reliable collision avoidance system, we propose the use of a visual camera for real-time object detection and target tracking. Moreover, the system should follow the International Regulations for Preventing Collisions at Sea (COLREGs) to avoid catastrophic accidents. In this paper only a part of the system is presented. For real-time object detection, the You Only Look Once (YOLO) ver. 3 convolutional neural network is used, and the target tracking filter based on a Kalman filter with built-in estimated relative position and velocity.

*Keywords:* collision avoidance; MASS COLREGs; YOLO; kalman filter

### Introduction

Improvements in computer performance and perception sensors have sparked interest in autonomous navigation research. Route planning, obstacle detection, guidance and control are of essential for safe operation. Moreover, autonomous navigation must follow the rules of COLREGs unless new rules are developed for Maritime Autonomous Surface Ships (MASS) (Kufalor et al. 2020; Zhuang et al. 2019) Rule 2. makes all marine vessels responsible for their actions in all circumstances (Jašić et al. 2011) and they are tasked to avoid collision with all means necessary. Also, Rule 5. demands an appropriate system for “proper lookout by sight and hearing as well as all available means appropriate in the prevailing circumstances and conditions so as to make appraisal of the situation and of the risk of collision” (Jašić et al. 2011). Ability to automatically detect and estimate motion of surrounding objects in changing maritime environment is therefore an important aspect for a collision avoidance system.

There are many types of collision avoidance equipment and navigation aids systems already used in maritime traffic environment. Among them, AIS provides information about surrounding vessels with great accuracy, but not all vessels are equipped with this device (Bošnjak et al. 2012). Ship radars with ARPA are also standard equipment onboard and can track a large area by using radar contacts. But they suffer from performance degradation when other unwanted reflection sources are present, have limitations in the blind sector and problems in detecting small targets (Heymann et al. 2014). Therefore, additional sensors are needed to improve situational awareness in autonomous navigation. In particular, sensors such as visual or thermo-visual cameras are already successfully used in robotic applications for close range obstacle detection. Visual cameras have high resolution and can improve object detection in the short to medium range. Since there is no perfect sensor, sensor fusion in the detection and tracking system is necessary for safe autonomous navigation.

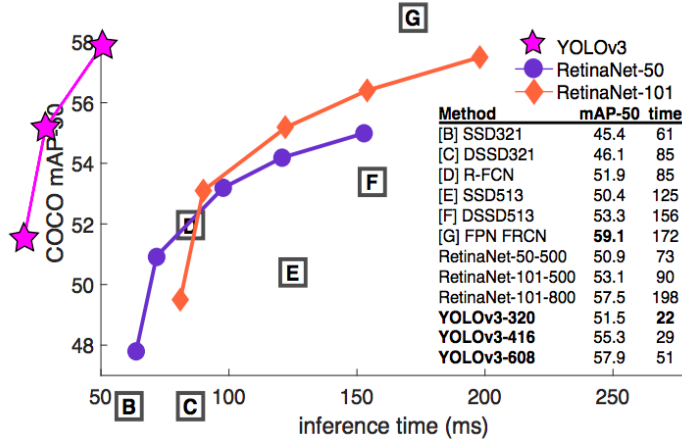


**Figure 1.** Illustration of autonomous collision avoidance system with multiple sensor fusion (source: Authors)

In this paper, we focus on part of the system illustrated in Figure 1. which is real time visual-based object detection based on deep learning computer vision. Deep learning computer vision methods are widely used for object detection. These convolutional neural network based algorithms can learn different object classes from thousands of training images to detect objects of interest (OOI) in new input images. For successful OOI tracking between frames, some kind of tracking algorithm must also be introduced.

### Object detection

In recent years, many object detection methods based on deep learning have been actively developed. Such as YOLO (Redmon et al. 2018), R-CNN (Schöller et al. 2019) and SSD (Liu et al. 2016) have tackled various challenging computer vision problems. A comparison of the speed versus accuracy of different methods can be seen in Figure 2.



**Figure 2.** Speed-accuracy tradeoff for metric mAP at 0.5 IOU.  
(Source: Redmon 2018)

YOLO version 3 is a single stage end-to-end object detection system that includes a backbone and a detection network (Redmon et al. 2018). The backbone network is Darknet-53, which consists of 52 fully convolutional layers, where input image is reduced to 416px\*416px. Then the image is down sampled 5 times for feature map extraction. Out of 52 layers in Darknet-53, 46 layers are divided into 23 residual units with 5 different sizes (Redmon et al. 2018). The output of the backbone network are feature maps of different sizes 13px\*13px, 26px\*26px and 52px\*52px, which are fed into three branches of the detection network to form a feature pyramid structure by up-sampling. Finally, the regression section performs the prediction of bounding box, object confidence and object category from the outputs of the feature pyramid network. According to (Redmon et al. 2018), the network predicts four coordinates for each bounding box (tx, ty, tw, th), if the cell is offset from the top left corner of the image by (cx, cy) and the bounding box prior has width and height (pw, ph), then the predictions correspond to:

$$b_x = \sigma(t_x) + c_x \quad (1)$$

$$b_y = \sigma(t_y) + c_y \quad (2)$$

$$b_w = p_w e^{t_w} \quad (3)$$

$$b_h = p_h e^{t_h} \quad (4)$$

where  $b_x, b_y$  are the x-y center coordinates of our prediction and  $b_w, b_h$  are the bounding box width and height.

Also, objectness score is predicted for each bounding box by using logistic regression. When objectness score is 1, it means that the bounding box is overlapping a ground truth object more than any other bounding box prior and is then selected as detection.

On COCO mAP 50 benchmark, YOLO version 3 performs equally well as other detectors but is significantly faster. Therefore, it is ideal for real-time applications. For our detector we used the official weight file obtained by training the network on the MSCOCO (Lin et al. 2014) dataset.

### Tracking algorithm

The main purpose of the tracking algorithm is the position and parameters estimation of the OOI in motion in a video sequence after its initial position is obtained by the object detector. Moreover, sometimes the object detector cannot detect the OOI in each successive frame, so the task of tracking algorithm is to fill the gaps between detections. The authors in (Han et al. 2020; Leykin et al. 2007; Bloisi 2009) proposed Kalman filtering (KF) for tracking for its real-time performance and good robustness. It is also pointed out that there are other promising tracking algorithms.

According to (Welch et al. 2006), KF consists of two main equations. The time update equation projects forward the current state and error covariance estimates to obtain the a priori estimates for the next time step. Also, the measurement update equations incorporates new measurements into the a priori estimate to obtain an improved a posteriori estimate.

To use KF as a tracking algorithm, the measurement vector  $M_k$  and the state vector  $q_k$  of the track within  $k$ -th frame are defined as follows:

$$M_k = [x_k \ y_k]^T \quad (5)$$

$$q_k = [x_k \ y_k \ V_{x_k} \ V_{y_k}]^T \quad (6)$$

where  $x_k, y_k$  are the center coordinates of the bounding box obtained from the object detector in  $k$ -th frame within the Cartesian coordinate system and  $V_{x_k}, V_{y_k}$  are  $x$  and  $y$  axis relative velocities. The control matrices  $A$  and  $C$  are initialized as follows:

$$A = \begin{bmatrix} 1 & 0 & dT & 0 \\ 0 & 1 & 0 & dT \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (8)$$

where  $dT$  is time between frames in seconds. The covariance matrix  $E_k$  is initialized as follows:

$$E_k = \begin{bmatrix} Ex_k & 0 & 0 & 0 \\ 0 & Ey_k & 0 & 0 \\ 0 & 0 & Evx_k & 0 \\ 0 & 0 & 0 & Evy_k \end{bmatrix} \quad (9)$$

where  $Ex_k$ ,  $Ey_k$  are the variance of  $x$  and  $y$  axis position and  $Evx_k$ ,  $Evy_k$  are the variance of  $x$  and  $y$  axis velocity. The noise matrix  $R_k$  is initialized as follows:

$$R_k = \begin{bmatrix} Nx_k & 0 \\ 0 & Ny_k \end{bmatrix} \quad (10)$$

where  $Nx_k, Ny_k$  are the variance of the  $x$ - and  $y$ -axis noise. At initialization, the matrices  $P_k$  and  $E_k$  are identical.

The time update equations are as follows:

$$\bar{q}_k = A \cdot q_{k-1} \quad (11)$$

$$\bar{P}_k = A \cdot P_{k-1} \cdot A^T + E_k \quad (12)$$

And measurement update equations are as follows:

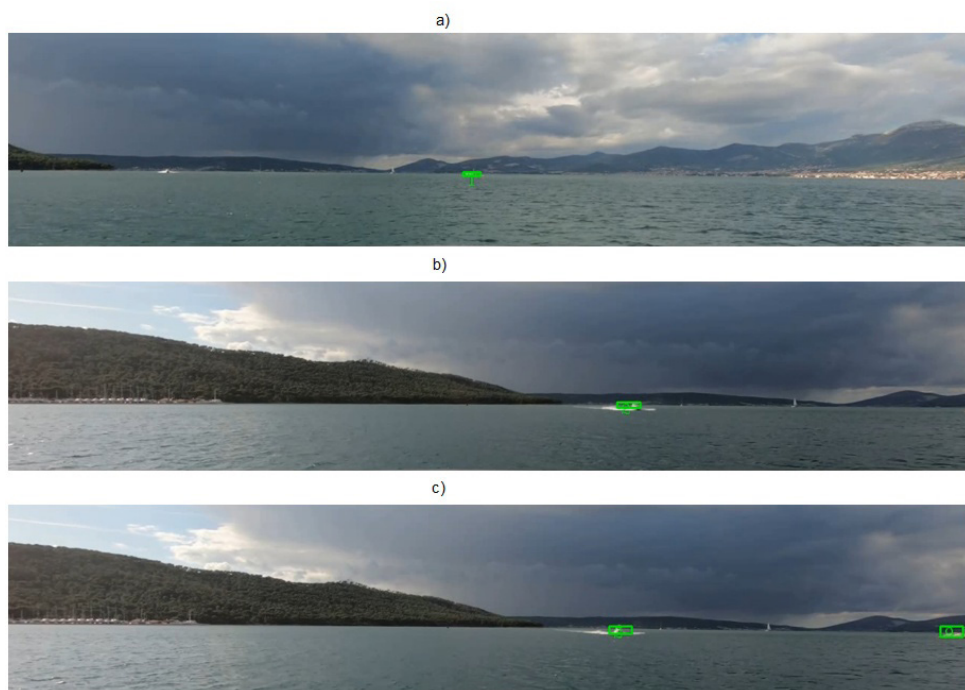
$$K_k = \bar{P}_k \cdot C^T (C \cdot \bar{P}_k \cdot C^T + R_k)^{-1} \quad (13)$$

$$q_k = \bar{q}_k + K_k (M_k - C \cdot \bar{q}_k) \quad (14)$$

where  $\bar{q}_k$  is a prior estimate of the state,  $q_k$  is a posterior estimate of the state,  $\bar{P}_k$  is a prior estimate of the error covariance, and  $K_k$  is the Kalman gain.

### Preliminary results

Object detection based on YOLO version 3 with KF tracking algorithm is implemented in C++. The system was tested on a personal computer with Intel i7-9700k 3.6 GHz processor, 16 GB RAM and Nvidia 2080 RTX GPU. The video sequence was captured by GoPro Hero 7 camera, mounted on a small ship during navigation, with a 2704px\*1520px resolution.



**Figure 3.** Preliminary results of the object detection and tracking system  
(source: authors)

In addition, the height of the video sequence was divided into two halves, as the lower half contained only the ship from which video sequence was filmed and was constantly being detected by detection network. Therefore, the lower half of the sequence was ignored for time being. YOLO version 3 proved to be fast, processing a single frame in 31 ms. Preliminary results can be seen in Figure 3, where on a) the system successfully detected and tracked the island with the buoy, but failed to detect the incoming motorboat, on the left of the image, when it was far away in line with land-sea line. It can also be seen that the sailboat in the middle of the image, which is in line with the land-sea line, was far away and was never successfully detected during the sequence. In Figure 3. b) and c), it can be seen that the motorboat was successfully detected and tracked after it approached the ship.

### **Conclusions**

YOLO version 3 proved to be fast, ready for real-time use. It detected all objects of interest at short to medium range from the ship, but had some problems when the objects were farther away near the land-sea line. It also did not detect sailboats

throughout the video sequence. These problems could be solved by training the network on a custom dataset with higher input resolution to the network (increasing from 413px\*413px to 608px\*608px). Also, the dataset should contain only annotated objects of interest, e.g. motorboats, sailboats, buoys, etc. The Kalman filter has been shown to be robust for single target tracking. Some sort of target association algorithm needs to be implemented in the Kalman filter to allow for multiple target tracking. Solving these problems and implementing the object detection and tracking system into a collision avoidance system will be the focus of our future work.

**Acknowledgments.** This paper is supported by the project „Establishment of reference database for studying the influence of weather conditions on marine video surveillance“ (Faculty of Maritime Studies, University of Split).

## REFERENCES

- Kufoalor, D.K.M. et al., 2020. Autonomous maritime collision avoidance: Field verification of autonomous surface vehicle behavior in challenging scenarios. *J. F. Robot.* **37**(3), 387 – 403. doi: 10.1002/rob.21919.
- Zhuang, J. et al., 2019. Collision avoidance for unmanned surface vehicles based on COLREGS. *ICTIS 2019 - 5th Int. Conf. Transp. Inf. Saf.* Wuhan, China, 1418–1425. doi: 10.1109/ICTIS.2019.8883829.
- Jašić, D. et al., 2011. *Međunarodna pravila o izbjegavanju sudara na moru*. Sveučilište u Zadru, Zrinski d.d. Čakovec.
- Bošnjak, R. et al., 2012. Automatic Identification System in Maritime Traffic and Error Analysis. *ToMS.* **1**(2), 77 – 84. doi: 10.7225/toms.v01.n02.002.
- Heymann, F. et al., 2014. Is ARPA suitable for automatic assessment of AIS targets? *Mar. Navig. Saf. Sea Transp. Adv. Mar. Navig.*, 223 – 232. doi: 10.1201/b14961-40.
- Redmon, J. & Farhadi, A., 2018. YOLOv3: An incremental improvement. *arXiv*.
- Schöller, F. E. T. et al., 2019. Assessing Deep-learning Methods for Object Detection at Sea from LWIR Images. *IFAC-Papers OnLine.* **52**(21), 64 – 71. doi: 10.1016/j.ifacol.2019.12.284.
- Liu, W. et al., 2016. SSD: Single shot multibox detector. *Lect. Notes Comput. Sci.* 9905 LNCS, 21–37. doi: 10.1007/978-3-319-46448-0\_2.
- Lin, T. Y. et al., 2014. Microsoft COCO: Common objects in context. *Lect. Notes Comput. Sci.* (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). **8693** LNCS(PART 5), 740 – 755. doi: 10.1007/978-3-319-10602-1\_48.

- Han, J. et al., 2020. Autonomous collision detection and avoidance for ARAGON USV: Development and field tests. *J. F. Robot.* **37**(6), 987 – 1002. doi: 10.1002/rob.21935.
- Leykin, A. et al., 2007. Thermal-visible video fusion for moving target tracking and pedestrian classification. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* doi: 10.1109/CVPR.2007.383444.
- Bloisi, D. & Iocchi, L., 2009. ARGOS – A video surveillance system for boat traffic monitoring in Venice. *Int. J. Pattern Recognit. Artif. Intell.* **23**(7), 1477 – 1502. doi: 10.1142/S0218001409007594.
- Welch, G. & Bishop, G., 2006. An Introduction to the Kalman Filter. *In Pract.* **7**(1), 1 – 16. doi: 10.1.1.117.6808.

✉ **Miro Petković**

<https://orcid.org/0000-0001-9365-959X>

Faculty of Maritime Studies

University of Split

Split, Croatia

E-mail: mpetkovic@pfst.hr

✉ **Danko Kezić**

<https://orcid.org/0000-0003-2055-8039>

Faculty of Maritime Studies

University of Split

Split, Croatia

E-mail: danko.kezic@pfst.hr

✉ **Igor Vujović**

<https://orcid.org/0000-0001-6649-1461>

Faculty of Maritime Studies

University of Split

Split, Croatia

✉ **Ivan Pavić**

Faculty of Maritime Studies

University of Split

Split, Croatia