Computers, Materials & Continua
DOI: 10.32604/cmc.2022.022989
Article

Tech Science Press

Computer-Vision Based Object Detection and Recognition for Service Robot in Indoor Environment

Kiran Jot Singh[1], Divneet Singh Kapoor[1,*], Khushal Thakur[1], Anshul Sharma[1] and Xiao-Zhi Gao[2]

[1]Embedded Systems & Robotics Research Group, Chandigarh University, Mohali, 140413, Punjab, India
[2]School of Computing, University of Eastern Finland, Yliopistonranta 1, FI-70210, Kuopio, Finland
*Corresponding Author: Divneet Singh Kapoor. Email: divneet.singh.kapoor@gmail.com
Received: 25 August 2021; Accepted: 12 October 2021

Abstract: The near future has been envisioned as a collaboration of humans with mobile robots to help in the day-to-day tasks. In this paper, we present a viable approach for a real-time computer vision based object detection and recognition for efficient indoor navigation of a mobile robot. The mobile robotic systems are utilized mainly for home assistance, emergency services and surveillance, in which critical action needs to be taken within a fraction of second or real-time. The object detection and recognition is enhanced with utilization of the proposed algorithm based on the modification of You Look Only Once (YOLO) algorithm, with lesser computational requirements and relatively smaller weight size of the network structure. The proposed computer-vision based algorithm has been compared with the other conventional object detection/recognition algorithms, in terms of mean Average Precision (mAP) score, mean inference time, weight size and false positive percentage. The presented framework also makes use of the result of efficient object detection/recognition, to aid the mobile robot navigate in an indoor environment with the utilization of the results produced by the proposed algorithm. The presented framework can be further utilized for a wide variety of applications involving indoor navigation robots for different services.

Keywords: Computer-vision; real-time computing; object detection; robot; robot navigation; localization; environment sensing; neural networks; YOLO

1 Introduction

Predicting the Future has always been difficult; estimating social change or future innovations is a risky affair. Yet, with the current developments in Artificial intelligence, it can be readily envisioned that robotic technology will rapidly advance in the coming decade, expanding its control over our lives. Industrial robots, which were once exclusive for huge factories, have already expanded into small businesses. Even with service robots, a 32% growth rate was witnessed in 2020 [1]. The trends reflect that by 2025, robots will be part of the ordinary landscape of the general population doing the most mundane household activities, sharing our house and workspace. This will allow them to grow bigger than the internet. Not only will they give access to information, but they will also enable everyone to

reach out and manipulate everything. However, manipulating the objects requires object detection and recognition in real-time while navigating in physical space, especially for time-critical services, such as surveillance, home assistance, emergency response, etc., that needs real-time data analysis.

The robot seamlessly navigating through the workspace requires accurate object identification without confusing that object with the other objects. Robots are equipped with sensors like a video camera to detect and recognise objects [2]. The majority of research in the field is focused on refining the existing algorithms for the analysis of the sensor data to obtain accurate information regarding the objects. Fortunately, object recognition is one of the most advanced areas of deep learning, which helps a system establish and train a model for identifying objects under multiple scenarios, making it useful for various applications.

Object detection and recognition are accomplished through computer vision-based algorithms. The CNN (Convolutional Neural Network) is the most common technique to extract features from an image. It was designed as an improvement to deep neural networks with the purpose of enhancing the processing of 1D information [3]. Various models have been developed based on CNN like YOLO (You Only Look Once) [4], RPN (Region Proposal Network) and Regions with CNN (R-CNN). Amongst these bounding box algorithms, YOLO maintains the right balance amongst increased precision of object detection & localisation in real-time while providing less inference time and retains the information. The framework consists of an efficient end to end pipeline for feeding the actual frames from the camera feed to the neural system and utilises the obtained outcomes to guide the robot with customisable activities which correspond to the detected class labels.

Once the objects are identified, the next major task of a mobile robot is to localise the position of the robot on the map of the unknown environment. SLAM (Simultaneous Localisation and Mapping) is one of the most widely used algorithms that use sensors such as ultrasonic sensors or laser scanners to map an unfamiliar environment while localising the position of the robot on the map [5–7]. With the advancements in sensor technology, the use of SLAM in emergencies like disaster management has increased in the past few years [8].

Keeping in view the requirements of a Service Robot navigating in an Indoor Environment. This article is focused upon:

- Designing a computer vision-based framework for a robot, navigating in an indoor environment.
- Proposing an improved navigation algorithm for robots, through the development of a novel YOLO architecture-based model for object detection and recognition.
- Evaluating the performance of the proposed model in contrast to the state of the art algorithms, through standardised parameters of mean Average Precision (mAP) Score, mean inference time, weight size and false-positive percentage.

The rest of the paper is organized as follows. Section 2 describes the related work in the field of object detection/recognition and navigation for a mobile robot. The computer-vision based object detection/recognition algorithm is proposed in Section 3, along with SLAM based indoor navigation. Section 4 illustrates the experiment design and the results of the experiment being conducted are described in Section 5. Finally, the concluding remarks and future scope are mentioned in Section 6.

2 Related Work

Many modern-day camera-based multimedia applications require the ability to identify different objects & their location in images, usually put in a bounding box. One of the most popular applications utilising this ability is the gesture-based selfie that can identify faces in the camera feed and track the gestures made by the user to trigger capturing of the image. This ability refers to object detection and is commonly based on either Region-based [9–11] or single shot [4,12] based techniques. Region-based techniques involve proposing the region (bounding box) containing any potential object in the scene and classifying the objects after that. A faster response is obtained from the region-based convolutional neural networks by utilising the entire network for the image instead of dedicating to regions. The authors in [13] confirm near real time performance on a graphical processing unit (GPU) running a frame rate of 5 frames per second (FPS). To reduce the delays associated with sequential division of object detection into region proposal & subsequent classification, the authors in [4] proposed YOLO, achieving comparable performance at a much higher frame rate of 30 FPS, owing to its simpler efficient architecture that unifies region proposal & classification. Furthermore, the authors in [14], extend the state-of-the-art real-time object recognition algorithm proposed in [4] to a faster, improved YOLOv2 algorithm, finding special applications in robotic platforms like in [15]. Neural Networks were tested for on board processing using a couple of Raspberry pi microprocessors, resulting in abysmal performance. Processing time reduced substantially when using NVIDIA's Graphical Processing Units (GPUs) (GTX750TI and 860 M); it took less than 0.5 s to process each picture on the GPU, whereas on the Intel i7 Central Processing Unit (CPU), the processing time was 9.45 s. The test demonstrates the need of great processing capabilities, in particular the impact of using a graphic card for real time object recognition applications.

The authors in [16] develop an application which solely depends on depth information. Microsoft Kinect returns the depth information about a pair of legs using YOLOv2 to develop an image. The authors established successful execution of YOLOv2 on NVIDIA Jetson TX2 with satisfactory detection efficiency, while subjecting the system to a varying (low to medium) traffic.

The authors in [17] incorporate developing a map of the surroundings, as well as the positions of items trained previously for identification by the neural network, for the robot to follow. The authors utilize YOLO algorithm was for the detection of objects, together with a 2D laser sensor, odometers, an RGB-D camera & furthermore, a camera having depth sensor that had a higher processing capacity than the Microsoft Kinect.

NAO humanoid robot developed by the authors in [18] utilized YOLO for object identification and tracking the neural network significantly assisted the robot in real-time object identification and tracking, according to certain testing results. In another instance, the YOLO algorithm demonstrated a real-time tennis ball recognition by a service bot developed by the authors in [19] for retrieval in a tennis court.

The authors in [20] used YOLO to compute correlation between humans & objects based on their spatial separation. YOLO perfectly detected whether or not a person in an image consisting of a person & a cup of coffee, is drinking coffee. Similarly the authors in [21] detect & classify household objects & furniture for localization & mapping using YOLO & SLAM running in a Robot Operating System (ROS) application.

Real-time object identification on resource-constrained systems has attracted several Neural network based solutions usually compressing a pre trained network or directly training a small network [22,23]. The reduced size & complexity result in reduced accuracy. The MobileNet [24] for example suffers significant loss in accuracy while employing depth-wise separable convolutions to

reduce computational size & complexity. Enabling real time object detection on resource-constrained systems therefore requires load resolution to cloud based computing solutions to avoid the inherent accuracy trade-off in built-in systems. The Application Programming Interfaces (APIs) in [25–27] provide machine learning based web solutions for object detection, but are limited to applications involving image analysis at a frame rate much lower than real time tasks. The authors in [28] analyse the performance of standard object detection algorithms for feed captured by drones, to confirm the feasibility of real time object tracking, although, the work remains devoid of real-world problems like impact of communication protocols (errors, power consumption & latencies), techniques like multi-threading to lower computational latencies. In a nutshell, the different parameters of efficient object detection/recognition are elucidated in Tab. 1, in terms of detection, learning and output.

The authors in [29] developed a robotic navigation system for environments like hospital & home. The authors in [30] developed a robotic obstacle avoiding navigation system using ultrasonic sensors. The authors in [31] suggest using multiple sensors to improve precision of navigation while utilising an RGB-D camera in their robot. The work in [32] utilises an object tracking system for dynamic path planning by predicting the future locations of the object. One of the notable works in robot mapping & navigation, SLAM, has been enhanced by the authors in [21] for household indoor environments. The work in [33] exploits sensor fusion of numerous odometer methods to develop a vision based localisation algorithm for curve tracking. The authors in [34] develop a low-cost autonomous mapping & navigation robot based on ROS.

The authors in [35] develop an easy & sophisticated adoption of the Potential fields' method, one of the most appreciated techniques for controlling mobile autonomous robot, for navigation. Similar performance was attained for theoretical & practical implementation of the proposed method with an exception for environmental ambiguity, where the performance would plummet. The work in [36] exploited Numerical Potential Field method to develop a superior robot navigation path planner by reducing the computational delays associated with the global path planning techniques. The authors in [37] develop & confirm the efficacy of a fuzzy logic based artificial potential field for mobile robot navigation through an omnidirectional mobile robot. The proposed work remains constrained to a limited obstacle environment. The authors in [38] model a multi-objective optimization problem targeting maximization of the distance travelled, reduction of distance to destination & maximization of distance to nearest obstacle, and test performance over ten diverse routes along with three different positions of obstacles.

Table 1: Taxonomy of existing methods for object detection/recognition

S. No.		Parameter	Components
1	Detection	Settings	Bounding box [39], Pixel mask [40]
		Paradigms	One stage [41],Two stage [41]
		Backbone Architecture	ResNet [42], DetNet [43], MobileNet [44], DenseNet [45]
		Proposal Generation Methods	Anchor based [46], Keypoint based [47], Computer vision based [48]
		Feature Representation	Multi Scale [49], Region [50], Contextual [51], Deformable [52]

(Continued)

Table 1: Continued

S. No.		Parameter	Components
2	Learning	Training	Data Augmentation [53], Imbalance sampling [54], Localization refinement [55], Cascade learning [56]
		Testing	Duplicate removal [57], Model acceleration [58]
3	Output	Application	Face detection [59], Object detection [60], Pedestrian detection [61]
		Benchmarks	KITTI [62], ETH [63], FDDB [64], Pascal VOC [65], MSCOCO [40]

A potential field technique-based robot for a dynamic environment with mobile targets & stationary obstacles, was introduced in [66]. The authors created a hybrid controller that combines potential fields with Mamdani fuzzy logic to define velocity and direction. Simulations were used to validate performance. The hybrid approach overcomes local minima in both static and dynamic environments. Similarly, prospective route planning capabilities for mobile robots were utilized in various environments by authors in [67]. Main disadvantage was local minima. By not considering the global minimum, the robot became trapped in a local minimum of the potential field function. The increase in attraction force to robots with distance implied a high risk of collision with the obstacles.

To aid physiotherapists with determination of posture-related issues, the authors in [68] used the Microsoft Kinect sensor to collect anthropometric data and the accompanying software programme to evaluate the body measurements with depth information. Microsoft Kinect suffers significant accuracy errors in the depth information although satisfactory results were obtained from mathematical models. The proposed work concentrates on finding posture related inconsistencies such as one shoulder being lower than the other in order to make it easier for experts in the field to work. The authors in [69] developed a MATLAB based control system in conjunction with Microsoft Kinect that identifies the objects in image & calculates the distance based on sensor data. Similarly the authors in [70,71] used Microsoft Kinect sensor for robotic applications.

3 Proposed Methodology

The framework designed for computer vision based navigation for indoor environments is shown in Fig. 1. The robot named MAI is equipped with various sensors for making efficient object detection and recognition, and actuators for navigating inside a closed space, while avoiding various obstacles to reach the destination through a planned path. The proximity sensor, RGB-D camera, and microphone provide environmental data to the robotic operational control unit to drive the computer vision algorithms for object detection and recognition. The information related to detected and recognized objects are passed on to the navigation block, which generates a path for robot navigation in the indoor environment. This also takes into account real-time data from the proximity sensor to avoid obstacles while navigating on the planned path. The actuators take the instructions from the robot operation control based on the inputs received from computer vision and navigation blocks to drive the robot in motion towards the destination. The detailed description of our proposed methodology is given in the following subsections.

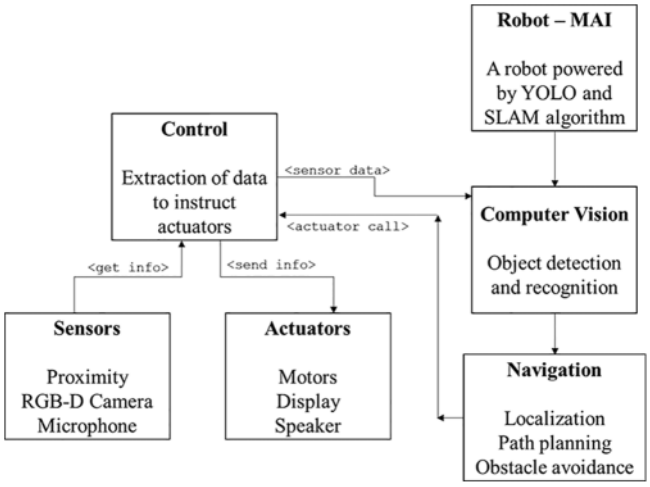

Figure 1: Navigation framework of MAI

3.1 Proposed Computer Vision Based Object Detection and Recognition

For an indoor mobile robot, there are many applications to object detection and recognition, such as obstacle detection and avoidance, staircase detection, edge detection, etc. The localization of the detected objects and its recognition/classification of the objects are the integral parts of the vision based object detection and recognition algorithm. The YOLO algorithm, developed by Redmon et al. [4], has evolved as a new approach for efficient object detection. YOLO models object detection in a frame as a regression problem. The input image is split in the form of an $n \times n$ grid. The cell of the grid containing the center of an object in input image is responsible for its detection. Thereafter, bounding boxes are predicted along with their respective class probabilities & confidence scores from grid cells to yield final detections. The confidence scores indicate the confidence of the algorithm over presence of object in the grid cell. Zero confidence score would imply absence of any object in the grid cell. Simultaneous prediction of multiple bounding boxes and their respective class probabilities through convolutional neural networks make YOLO extremely fast by avoiding the complex pipelines that limit the performance of traditional detection algorithms. As compared to the conventional two-step CNN-based object detection algorithms, YOLO provides good object detection results utilizing a single neural network to predict the bounding boxes, different classes, and the associated probabilities, with fast speed. The base YOLO algorithm includes a single neural network that uses full-scale pictures to predict bounding boxes and class probabilities in one cycle of assessment. The base YOLO algorithm is capable of handling the image processing with a speed of 45 FPS, quite faster compared to the industry standards. Furthermore, the base YOLO algorithm can be optimized directly on the object detection performance, as it utilizes only a single network.

For the mobile robot, which is navigating in an indoor environment, it needs to detect and localize the object, so as to further take the actions on the basis of label and location of object. In line with the aforementioned problem statement for the underlying system, the proposed algorithm takes the real-time video stream from the RGB-D camera mounted on the robot as input. The proposed algorithm outputs the class label of the detected object along with its location. The bounding boxes drawn over the detected objects are then utilized for drawing inferences from the robots' perspective. Further, these inferences are utilized by the robot to take certain actions based on the objects' classes. The YOLO algorithm extracts features from the input images (broken down from the video stream) by using the

convolutional neural networks, which are connected to the fully-connected neural network layers to predict the class probabilities and coordinates for the objects being detected.

To increase the speed of the base YOLO algorithm on the real-time video stream, the proposed algorithm utilizes smaller sizes of the filters of convolutional layers, with minimal loss of the overall accuracy. The modification of the base algorithm has been governed by two factors, that is, weight quantization and reduction in the number of filters of convolutional layers. Without a significant loss in the overall accuracy of the algorithm, weight size of the neural network being used can be reduced to mitigate the large memory consumption and longer loading time. This is accomplished by replacing floating-point computations to much faster integer computations, with a trade-off for reduction in the overall accuracy.

Also, the proposed algorithm utilizes only 16 convolutional layers with a *maxpool* layer of 2×2 of stride 2. This layer structure is then connected to 3 fully-connected neural network layers to return the final output. This proposed algorithm has been compared with other algorithms such as RFCN, YOLOv3 and Faster RCNN, in Section 4. The output of the proposed modified YOLO-based object detection algorithm is the bounding box and the class tag for the detected object. The proposed algorithm utilizes independent logistic classifiers to predict the likeliness of the detected object for a specific class. The resultant box of prediction can be given as:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \qquad & b_w &= p_w \exp(t_w) \\ b_y &= \sigma(t_y) + c_y \qquad & b_h &= p_h \exp(t_h) \end{aligned} \tag{1}$$

The prediction of multiple bounding boxes is performed by the YOLO algorithm per grid cell. In order to calculate the true positive for loss, the ground truth with the highest IoU (intersection over union) is selected. This strategy leads to specialism among prediction of the bounding boxes. The sum-squared error between the ground-truth and predictions is used by YOLO to calculate loss. The function of loss comprises of the classification loss, the localization loss which refers to the errors between the ground truth and predicted boundary boxes, and the confidence loss which refers to the box objectness, which are given as

$$\begin{aligned} Loss_{classification} &= \sum_{i=0}^{n} 1_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \\ Loss_{localization} &= \sum_{obj=0}^{n} 1^{obj} [(b_x - I_x)^2 + (b_y - I_y)^2 + (b_w - I_w)^2 + (b_h - I_h)^2] \end{aligned} \tag{2}$$

where, b_x, b_y is the predicted value of center coordinates while I_x, I_y is the real value, b_w, b_h is the width and height of predict bounding box, while I_w, I_h is the real value, $\hat{p}_i(c)$ denotes the conditional class probability for class c in cell i. The underlying algorithm's workflow is defined as the following:

Algorithm 1: Real-time object localization and recognition Algorithm

Input: Camera live/real-time video stream with minimum frame rate of 10 FPS
Output: Bounding box coordinates, values of confidence and object labels
1: Initialize a vacant *Queue*
2: Acquire feed from RGB-D camera
3: Calculate the frame rate *Fr*
4: Link the feed of video to the computer-vision based system

(Continued)

Algorithm 1: Continued

```
5: while capturing of frame is on do
6:    for every new frame Fn do
7:          if system is active then
8:                Save frame Fn in the Queue
9:          else
10:               Run the model for network with Fn as input for inferring
11:               Send values of confidence, coordinates for bounding boxes and output labels
12:               Send Fn overlapped with output of model
13:         end if
14:         if (Length_Queue ≥ max{5, Fr}) then
15:               Skip every frame in Queue
16:         end if
17:   end for
18:   end while
```

The algorithm utilizes neural networks which are fed by the frame-wise images extracted from the real-time video, to return the coordinate list in terms of x and y for the bounding boxes of bottom-right and top-left corners, in addition to the equivalent label of class for each of the objects detected. For high frame rate and/or longer computation time in inferencing, few frames are skipped to match with the real-time processing of the video, and mitigating the errors caused due to delayed detection results being relayed to the robot for action and indoor navigation.

3.2 Navigation in an Indoor Environment

The indoor navigation of the mobile robot is governed by simultaneous localization and mapping (SLAM) algorithm, which defines the navigation environment map. The data from the RGB-D camera and proximity sensors, after object detection/recognition is utilized by SLAM algorithm to plan the navigation path for the robot. The time progression for the robot navigation is defined as $t \in \{1, 2, \ldots, T\}$, where last time step of the robot is given by T. The pose function of the mobile robot is defined in the terms of speed, position, direction, and transmission range of robots, denoted as

$$\psi_t = f(\Upsilon(x, y), v, \theta, L) \tag{3}$$

where, $\Upsilon(x, y)$ denotes the position of the robot, v denotes the speed, θ denotes the direction and L denotes the transmission range of robot at discrete-time instance t. The area in which the robot has to navigate is further divided into a matrix of cells, given as $g \times h$, with g and h being the whole numbers. Each cell of the matrix so created can be illustrated as

$$\bar{U} = U_{g,h}z(g, h) \tag{4}$$

The advantage of SLAM is its high convergence and its ability to efficiently handle the uncertainty makes it useful for the map building applications [72]. In order to represent the map in terms of the finite vector, a graph-based SLAM approach is utilized, which records corresponding observations e_t from the on-board proximity sensors. The distance measurements (q_t) are performed at discrete-time steps t to find a new pose function ψ_{t+1} of the robot, which is denoted as

$$q_t = \left(\frac{\psi_t}{\psi_{t+1}}\right) \tag{5}$$

where, ψ_t and ψ_{t+1} denotes the before- and after-movement poses of the mobile robot navigating in indoor environment. At discrete-time instance t, the probabilistic form of the evaluated joint posterior over the map is expressed as [73]

$$\wp(\psi_{1:t}, b|e_{0:t}, q_{0:t-1}, \psi_0) \tag{6}$$

In order to store the overall data for the map in each iteration, the maximum-likelihood is re-evaluated while integrating each sensor data, which is expressed as

$$\wp(\psi_t, b|e_{0:t}, q_{0:t}, \psi_0) = \int\int\cdots\int \wp(\psi_{1:t}, b|e_{0:t}, q_{0:t-1}, \psi_0) d\psi_1 d\psi_2 \cdots d\psi_{t-1} \tag{7}$$

So, the graph-based SLAM is a two-step procedure for the map construction. The first step is the description and integration of the sensor-dependent constraints, depicted as front-end, and the second step is the abstract depiction of sensor-agnostic data, depicted as back-end [74,75].

4 Experiment

In order to test the implementation of the proposed framework with computer vision based navigation for indoor environments we deployed MAI Robot [76] in an indoor environment of Block 1, Chandigarh University (CU) which is a nonprofit educational organization located at Mohali, India. The ground truth images of the indoor environment at CU with map and robot navigation trail is shown in Fig. 2.

Figure 2: Ground truth images of indoor environment at CU with map and robot navigation trail

In order to avoid the experimental bias, we positioned some common furniture items of different shapes and sizes at the test scenarios. This experiment is designed for participants in an indoor environment scenario where they share the space with a service robot. The participants were made

aware about the test task before conducting the experiment. However, they did not possess any technical knowledge about programming and operating a robot. The whole experiment revolves around the theme of a future smart home where robots will be part of the ecosystem and will share common space with humans. These robots will perform the daily mundane jobs like answering door bells, serving guests etc. where real time object recognition and navigation will decide their effectiveness in that environment.

The robot designed for conducting the experiment is named as MAI as shown in Fig. 3 which is equipped with a single-board computer (Quad-core Cortex-A72 processor, 4 GB RAM, and 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless connectivity) for performing computations. MAI has proximity sensors, a Microsoft XBOX 360 Kinect RGB-D based camera along with RGB camera for detecting obstacles and conducting navigation.

Figure 3: MAI: Robot developed and used for conducting experiments

At the beginning of the experiment MAI self-located itself at the start of the main entrance of the corridor at CU. Based upon the target coordinates and topology semantics, MAI planned an optimum path based on the previous information of the map available which was developed using SLAM algorithm. The MAI navigated on its own without any intervention of manual control. In case when MAI encountered some obstacles like walking people, furniture or walls, it avoided those obstacles and re-planned its path in order to reach the destination.

The video stream from the camera is fed frame by frame to the neural network of YOLO algorithm in form of matrix which returns inference in terms of bounding boxes of different colors with labels for different objects as shown in Fig. 4. These labels are fed back to the MAI in order to take the programmable action to support the navigation as per the objects detected. In case, if the frame rate of video input feed is too high from the camera, the intermittent frames are dropped in pursuit of preserving the sanctity of navigation in real time.

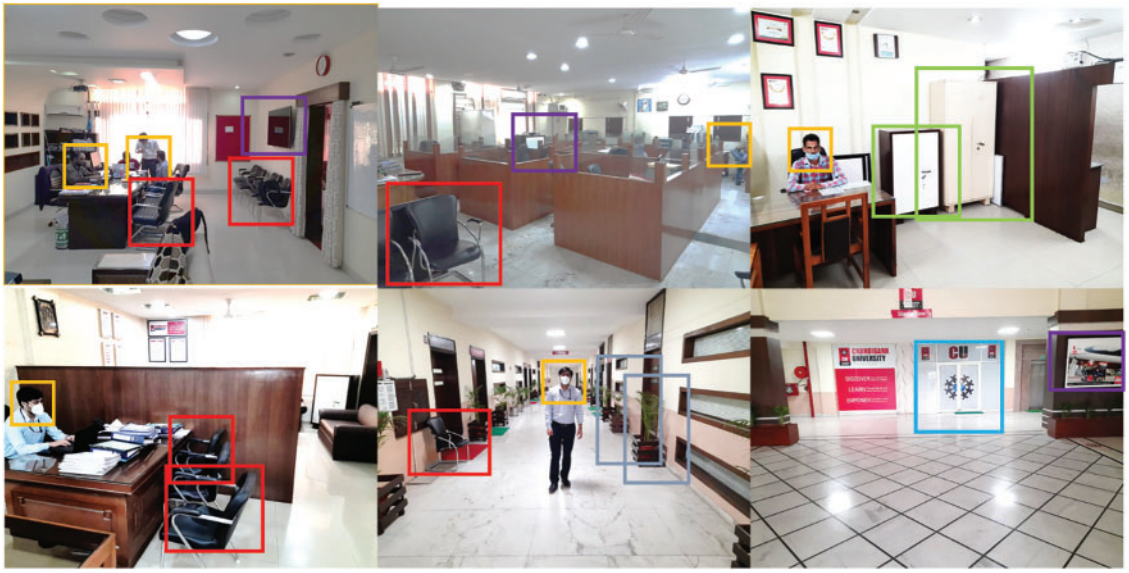

Figure 4: Results for object detection using YOLO in different rooms and corridor of CU

5 Result Discussion

The results for the experiment carried out in the indoor environment of CU with the proposed YOLO model have been presented from two points of view. Firstly, the proposed YOLO architecture with weight size 89.88 MB has been compared with state of the art algorithms named Faster RCNN [13], RFCN [77] and YOLOv3 [14] as depicted in Fig. 5. It can be observed from the results that proposed YOLO architecture performed considerably well in terms of mAP scores, mean inference time and weight size. Here, mAP score are the mean average precision score that compares the bounding box of ground truth image to the detected box and returns a score, where higher score represents better object detection. It can be connoted from Fig. 5 that the proposed computer-vision based modified YOLO algorithm illustrates 50% lesser mAP score. Mean inference time refers to the time taken by the algorithm to make the prediction where less the time better supports the real tile scenarios. The proposed algorithm takes 70% less time to compute inference. The weight size refers to the memory space and algorithm takes, which is 84% smaller for proposed algorithm as compared to Faster-RCNN and RFCN.

Secondly, we tested the proposed YOLO architecture for calculating the accuracy of the algorithm along with comparison of false positive percentage (which refers to how inaccurate the algorithm is in terms of detection) for other algorithms as well. The proposed algorithm very effectively detected different objects like chairs, doors, plants, TV screen and humans as shown in Tab. 2. It can be seen that output of the proposed algorithm is satisfactory for different objects except TV screens. Furthermore, Fig. 6 shows the comparison of the proposed algorithm with other algorithms in terms of false positive rate percentage, where less the percentage better the algorithm. It can be observed from the results that the proposed YOLO architecture performs considerably well in terms of false positive rate percentage. The proposed algorithm illustrates a false positive percentage of 4%, in comparison to 3.5% of RFCN algorithm. Considering the weight-size of the proposed algorithm which is approximately 7 times lesser than RFCN, the false positive percentage is quite acceptable for its implementation for various applications on low-computing devices. Furthermore, the mean

inference time of the proposed algorithm is minimum as compared to other algorithms, which makes it the best candidate for implementation on low-computing devices.

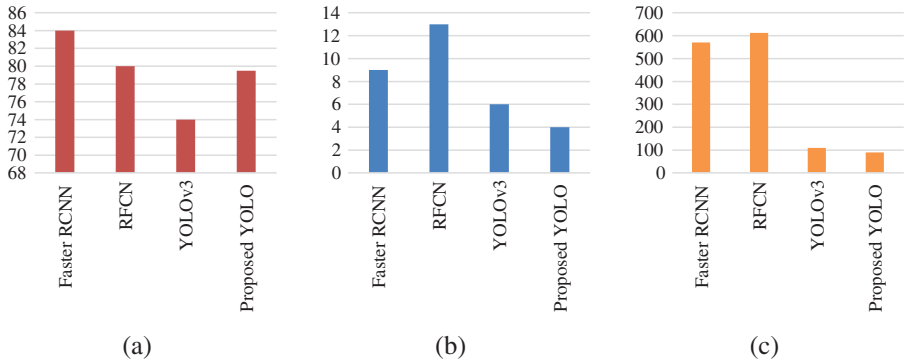

Figure 5: Comparison of proposed YOLO architecture with different algorithms in terms of (a) mAP Score (b) Mean inference time (c) Weight size

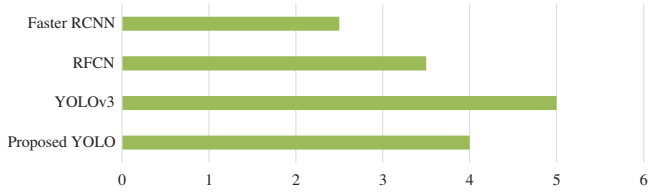

Figure 6: False positive percentage for different algorithms

Table 2: Verification results for different objects

Object	Verification (Number of times)	Successful verification (Number of times)	Success rate (%age)
Person	110	82	74.55
Amirah	110	75	68.18
Door	110	76	69.09
Plant	110	81	73.64
Chair	110	85	77.27
TV screen	110	68	61.82

6 Concluding Remarks

Service robots are going to be integrated into our daily lives and will share space with us. They will be part of our homes, shopping malls, government offices, schools and hospitals. In this paper a framework has been designed for computer vision based navigation for indoor environments to implement the functionalities of service robots. The robot named MAI makes use of SLAM for navigation and a YOLO based model has been proposed for computer vision based object detection and recognition. The proposed algorithm has been compared with state of the art algorithms named

Faster RCNN, RFCN and YOLOv3. The proposed algorithm takes least mean inference time and it has the smallest weight size as compared to other algorithms. Furthermore, its false positive percentage is comparable to state of the art algorithms. Our experimental results show that the proposed algorithm detects most of the obstacles with desired reliability. In future, we plan to test the MAI in public spaces with better proximity sensors to further enhance the navigation reliability as well.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. J. Singh, D. S. Kapoor and B. S. Sohi, "Selecting social robot by understanding human–Robot interaction," in *Int. Conf. on Innovative Computing and Communications*, New Delhi, India, 2021, pp. 203–213.

[2] M. J. Islam, J. Hong and J. Sattar, "Person-following by autonomous robots: A categorical overview," *International Journal of Robotics Research*, vol. 38, no. 14, pp. 1581–1618, 2019.

[3] L. Bondi, D. Güera, L. Baroffio, P. Bestagini, E. J. Delp *et al.,* "A preliminary study on convolutional neural networks for camera model identification," *Electronic Imaging*, vol. 2017, no. 7, pp. 67–76, 2017.

[4] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: unified, real-time object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 779–788, 2016.

[5] R. -J. Yan, J. Wu, S. -J. Lim, J. -Y. Lee and C. -S. Han, "Natural corners-based SLAM in unknown indoor environment," in *2012 9th Int. Conf. on Ubiquitous Robots and Ambient Intelligence (URAI)*, Daejeon, Korea, pp. 259–261, 2012.

[6] P. Yang, "Efficient particle filter algorithm for ultrasonic sensor-based 2D range-only simultaneous localisation and mapping application," *IET Wireless Sensor System*, vol. 2, no. 4, pp. 394–401, 2012.

[7] Z. Zhang, H. Guo, G. Nejat and P. Huang, "Finding disaster victims: A sensory system for robot-assisted 3D mapping of urban search and rescue environments," in *Proc. 2007 IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, pp. 3889–3894, 2007.

[8] R. Ventura and P. U. Lima, "Search and rescue robots: The civil protection teams of the future," in *2012 Third Int. Conf. on Emerging Security Technologies*, Lisbon, Portugal, pp. 12–19, 2012.

[9] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 580–587, 2014.

[10] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, pp. 1–9, 2015. [Online]. Available: https://arxiv.org/abs/1504.08083.

[11] P. Sachdeva and K. J. Singh, "Automatic segmentation and area calculation of optic disc in ophthalmic images," in *2015 2nd Int. Conf. on Recent Advances in Engineering & Computational Sciences (RAECS)*, Chandigarh, India, pp. 1–5, 2015.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed *et al.,* "SSD: Single shot multibox detector," in *European Conf. on Computer Vision*, Amsterdam, The Netherlands, pp. 21–37, 2016.

[13] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[14] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *CoRR*, vol. abs/1804.0.2767, pp. 1–6, 2018. [Online]. Available: https://arxiv.org/abs/1804.02767.

[15] R. Tenguria, S. Parkhedkar, N. Modak, R. Madan and A. Tondwalkar, "Design framework for general purpose object recognition on a robotic platform," in *2017 Int. Conf. on Communication and Signal Processing (ICCSP)*, Chennai, India, pp. 2157–2160, 2017.

[16] A. Lucian, A. Sandu, R. Orghidan and D. Moldovan, "Human leg detection from depth sensing," in *2018 IEEE Int. Conf. on Automation, Quality and Testing, Robotics (AQTR)*, Cluj-Napoca, Romania, pp. 1–5, 2018.

[17] D. Bersan, R. Martins, M. Campos and E. R. Nascimento, "Semantic map augmentation for robot navigation: A learning approach based on visual and depth data," in *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, Pessoa, Brazil, pp. 45–50, 2018.

[18] X. Zhao, H. Jia and Y. Ni, "A novel three-dimensional object detection with the modified You only look once method," *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, pp. 1–13, 2018.

[19] S. Gu, X. Chen, W. Zeng and X. Wang, "A deep learning tennis ball collection robot and the implementation on nvidia jetson tx1 board," in *2018 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM)*, Auckland, New Zealand, pp. 170–175, 2018.

[20] M. P. Zapf, A. Gupta, L. Y. M. Saiki and M. Kawanabe, "Data-driven, 3-D classification of person-object relationships and semantic context clustering for robotics and AI applications," in *2018 27th IEEE Int. Symposium on Robot and Human Interactive Communication (RO-MAN)*, Nanjing, China, pp. 180–187, 2018.

[21] P. Maolanon, K. Sukvichai, N. Chayopitak and A. Takahashi, "Indoor room identify and mapping with virtual based SLAM using furnitures and household objects relationship based on CNNs," in *2019 10th Int. Conf. of Information and Communication Technology for Embedded Systems (IC-ICTES)*, Bangkok, Thailand, pp. 1–6, 2019.

[22] M. Wang, B. Liu and H. Foroosh, "Factorized convolutional neural networks," in *Proc. of the IEEE Int. Conf. on Computer Vision Workshops*, Venice, Italy, pp. 545–553, 2017.

[23] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally *et al.,* "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," arXiv Prepr. arXiv1602.07360, 2016.

[24] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang *et al.,* "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv Prepr. arXiv1704.04861, 2017.

[25] A. Ravulavaru, in *Google Cloud AI Services Quick Start Guide: Build Intelligent Applications with Google Cloud AI Services*, Birmingham, UK: Packt Publishing Ltd, pp. 113–143, 2018.

[26] A. Koul, S. Ganju and M. Kasam, "Practical Deep Learning for Cloud," in *Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python, Keras & TensorFlow*, United States: O'Reilly Media, 2019.

[27] S. Mane and G. Shah, "Facial recognition, expression recognition, and gender identification," in *Data Management, Analytics and Innovation*," *Switzerland: Springer*, vol. 808, pp. 275–290, 2019.

[28] J. Lee, J. Wang, D. Crandall, S. Šabanović and G. Fox, "Real-time, cloud-based object detection for unmanned aerial vehicles," in *2017 First IEEE Int. Conf. on Robotic Computing (IRC)*, Taichung, Taiwan, pp. 36–43, 2017.

[29] G. Schmidt, U. D. Hanebeck and C. Fischer, "A mobile service robot for the hospital and home environment," in *Int. Advanced Robotics Programme (IARP 1997), Proc. of the Second Int. Workshop on Service and Personal Robots: Technologies and Applications*, Karlsruhe, Germany, pp. 1–8, 1997.

[30] T. Grami and A. S. Tlili, "Indoor mobile robot localization based on a particle filter approach," in *2019 19th Int. Conf. on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, Sousse, Tunisia, pp. 47–52, 2019.

[31] S. Diddeniya, A. Adikari, H. N. Gunasinghe, P. R. S. De Silva, N. C. Ganegoda *et al.,* "Vision based office assistant robot system for indoor office environment," in *2018 3rd Int. Conf. on Information Technology Research (ICITR)*, Moratuwa, Sri Lanka, pp. 1–6, 2018.

[32] A. Ess, K. Schindler, B. Leibe and L. Van Gool, "Object detection and tracking for autonomous navigation in dynamic environments," *International Journal of Robotics Research*, vol. 29, no. 14, pp. 1707–1725, 2010.

[33] D. Tick, A. C. Satici, J. Shen and N. Gans, "Tracking control of mobile robots localized via chained fusion of discrete and continuous epipolar geometry, IMU and odometry," *IEEE Transactions on Cybernetics*, vol. 43, no. 4, pp. 1237–1250, 2012.

[34] Y. Li and C. Shi, "Localization and navigation for indoor mobile robot based on ROS," in *2018 Chinese Automation Congress (CAC)*, Xi'an, China, pp. 1135–1139, 2018.
[35] Z. Laouici, M. A. Mami and M. F. Khelfi, "Hybrid method for the navigation of mobile robot using fuzzy logic and spiking neural networks," *Intelligent Systems and Applications*, vol. 6, no. 12, pp. 1–9, 2014.
[36] F. G. Rossomando, C. Soria and R. Carelli, "Adaptive neural dynamic compensator for mobile robots in trajectory tracking control," *IEEE Latin America Transactions*, vol. 9, no. 5, pp. 593–602, 2011.
[37] H. Eraqi, Y. EmadEldin and M. Moustafa, "Reactive collision avoidance using evolutionary neural networks," arXiv Prepr. arXiv1609.08414, 2016.
[38] O. Mohareri, "Mobile robot trajectory tracking using neural networks," M.S. Thesis, Department of Electrical Engineering, American University of Sharjah, Sharjah, UAE, 2009.
[39] Y. He, C. Zhu, J. Wang, M. Savvides and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 2888–2897, 2019.
[40] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec and K. Cho, "Augmentation for small object detection," arXiv Prepr. arXiv1902.07296, 2019.
[41] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang *et al.,* "Libra R-CNN: Towards balanced learning for object detection," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 821–830, 2019.
[42] X. Ou, P. Yan, Y. Zhang, B. Tu, G. Zhang *et al.,* "Moving object detection method via ResNet−18 with encoder–decoder structure in complex scenes," *IEEE Access*, vol. 7, pp. 108152–108160, 2019.
[43] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng *et al.,* "Detnet: design backbone for object detection," in *Proc. of the European Conf. on Computer Vision (ECCV)*, Munich, Germany, pp. 334–350, 2018.
[44] Y. -C. Chiu, C. -Y. Tsai, M. -D. Ruan, G. -Y. Shen and T. -T. Lee, "Mobilenet-SSDv2: An improved object detection model for embedded systems," in *2020 Int. Conf. on System Science and Engineering (ICSSE)*, Kagawa, Japan, pp. 1–5, 2020.
[45] S. Zhai, D. Shang, S. Wang and S. Dong, "DF-SSD: An improved SSD object detection algorithm based on DenseNet and feature fusion," *IEEE Access*, vol. 8, pp. 24344–24357, 2020.
[46] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li *et al.,* "Foveabox: Beyond anchor-based object detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 7389–7398, 2020.
[47] S. Hare, A. Saffari and P. H. S. Torr, "Efficient online structured output learning for keypoint-based object tracking," in *2012 IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, RI, USA, pp. 1894–1901, 2012.
[48] J. Lamer, D. Cymbalak and F. Jakab, "Computer vision based object recognition principles in education," in *2013 IEEE 11th Int. Conf. on Emerging eLearning Technologies and Applications (ICETA)*, Stara Lesna, Slovakia, pp. 253–257, 2013.
[49] H. Wang, Q. Wang, M. Gao, P. Li and W. Zuo, "Multi-scale location-aware kernel representation for object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 1248–1257, 2018.
[50] Y. Li, S. Wang, Q. Tian and X. Ding, "Feature representation for statistical-learning-based object detection: A review," *Pattern Recognition*, vol. 48, no. 11, pp. 3542–3559, 2015.
[51] A. Torralba, "Contextual priming for object detection," *International. Journal of Computer Vision*, vol. 53, no. 2, pp. 169–191, 2003.
[52] X. Zhu, W. Su, L. Lu, B. Li, X. Wang *et al.,* "Deformable DETR: Deformable transformers for end-to-end object detection," *CoRR*, vol. abs/2010.04159, pp. 1–16. 2020. [Online]. Available: https://arxiv.org/abs/2010.04159.
[53] B. Zoph, E. D. Cubuk, G. Ghiasi, T. -Y. Lin, J. Shlens *et al.,* "Learning data augmentation strategies for object detection," in *European Conf. on Computer Vision*, Glasgow, UK, pp. 566–583, 2020.
[54] Y. Li, T. Wang, B. Kang, S. Tang, C. Wang *et al.,* "Overcoming classifier imbalance for long-tail object detection with balanced group softmax," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 10991–11000, 2020.

[55] K. -W. Cheng, Y. -T. Chen and W. -H. Fang, "Improved object detection with iterative localization refinement in convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2261–2275, 2017.
[56] J. Li and Y. Zhang, "Learning surf cascade for fast and accurate object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Portland, OR, USA, pp. 3468–3475, 2013.
[57] L. Qi, S. Liu, J. Shi and J. Jia, "Sequential context encoding for duplicate removal," *CoRR*, vol. abs/1810.08770, pp. 1–11, 2015. [Online]. Available: http://arxiv.org/abs/1810.08770.
[58] X. Zhang, J. Zou, K. He and J. Sun, "Accelerating very deep convolutional networks for classification and detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 1943–1955, 2015.
[59] C. Li, R. Wang, J. Li and L. Fei, "Face detection based on YOLOv3," in *Recent Trends in Intelligent Computing, Communication and Devices*, Springer, Singapore. pp. 277–284, 2020.
[60] K. Li, G. Wan, G. Cheng, L. Meng and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, pp. 296–307, 2020.
[61] J. Cao, Y. Pang, J. Xie, F. S. Khan and L. Shao, "From handcrafted to deep features for pedestrian detection: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
[62] A. Geiger, P. Lenz, C. Stiller and R. Urtasun, "Vision meets robotics: the kitti dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
[63] S. Maji and J. Malik, "Object detection using a max-margin hough transform," in *2009 IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, FL, pp. 1038–1045, 2009.
[64] D. E. King, "Max-margin object detection," *CoRR*, vol. abs/1502.00046, pp. 1–8, 2015. [Online]. Available: https://arxiv.org/abs/1502.00046.
[65] S. Vicente, J. Carreira, L. Agapito and J. Batista, "Reconstructing pascal voc," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, Ohio, USA, pp. 41–48, 2014.
[66] J. Ghommam and M. Saad, "Backstepping-based cooperative and adaptive tracking control design for a group of underactuated AUVs in horizontal plan," *International Journal of Control*, vol. 87, no. 5, pp. 1076–1093, 2014.
[67] C. Wong, E. Yang, X. -T. Yan and D. Gu, "Adaptive and intelligent navigation of autonomous planetary rovers—A survey," in *2017 NASA/ESA Conf. on Adaptive Hardware and Systems (AHS)*, Pasadena, CA, pp. 237–244, 2017.
[68] S. A. Mantserov, L. O. Fedosova and M. A. Grishin, "Software development for automated system of recording the range of motion of the shoulder joint during rehabilitation," in *2020 Int. Conf. on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, Sochi, Russia, pp. 1–6, 2020.
[69] M. A. Tauwafak, A. S. Al Araji and B. R. Sadiq, "Development of aFree-navigation mobile robot system based on a digital image processing methodology," in *IOP Conf. Series: Materials Science and Engineering*, Bristol, vol. 433, no. 1, pp. 12059, 2018.
[70] Y. Wang, G. Song, G. Qiao, Y. Zhang, J. Zhang *et al.,* "Wheeled robot control based on gesture recognition using the kinect sensor," in *2013 IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, Shenzhen, China, pp. 378–383, 2013.
[71] H. Fadli, C. Machbub and E. Hidayat, "Human gesture imitation on NAO humanoid robot using kinect based on inverse kinematics method," in *2017 Int. Conf. on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA)*, Surabaya, Indonesia, pp. 116–120, 2017.
[72] A. R. Vidal, H. Rebecq, T. Horstschaefer and D. Scaramuzza, "Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 994–1001, 2018.
[73] M. Eck, "Simultaneous 2D Localization and 3D Mapping on a mobile Robot with Time-of-Flight Sensors," M.S. thesis, University of Applied Sciences, Koblenz, 2013.
[74] X. -S. Yang, M. Karamanoglu and X. He, "Flower pollination algorithm: A novel approach for multiobjective optimization," *Engineering Optimization*, vol. 46, no. 9, pp. 1222–1237, 2014.

[75] G. Grisetti, R. Kümmerle, C. Stachniss and W. Burgard, "A tutorial on graph-based SLAM," *Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.
[76] K. J. Singh, D. S. Kapoor and B. S. Sohi, "The MAI: A robot for/by everyone," in *Companion of the 2018 ACM/IEEE Int. Conf. on Human-Robot Interaction*, Chicago, USA, pp. 367–368, 2018.
[77] J. Dai, Y. Li, K. He and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks." *CoRR*, vol. abs/1605.06409, pp. 1–11, 2016. [Online]. Available: https://arxiv.org/abs/1605.06409.