Design and implementation monitoring robotic system based on you only look once model using deep learning technique

Maad Issa Al-Tameemi[1], Ammar A. Hasan[1], Bashra Kadhim Oleiwi[2]
[1]Department of Computer Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq
[2]Department of Control and Systems Engineering, University of Technology-Iraq, Baghdad, Iraq

Article Info

Article history:

Received Jan 1, 2022
Revised Aug 25, 2022
Accepted Sep 23, 2022

Keywords:

Deep learning
Internet of things
Raspberry pi 4
Real-time objects detection
Robot
Webcam
YOLOv5

ABSTRACT

The need for robotics systems has become an urgent necessity in various fields, especially in video surveillance and live broadcasting systems. This work is aimed to design and implement a robotic system which is based mainly on raspberry pi 4 model B to control this overall system and display a live video by using a webcam (USB camera) as well as using (YOLOv5) you only look once algorithm-version five a deep learning-based object detector to detect, recognize and display objects in real-time. This deep learning algorithm is highly accurate and fast and is implemented by Python, OpenCV, PyTorch codes and the Context Object Detection Task (COCO) 2020 dataset. This robot can move in all directions and in different places especially in undesirable places to transmit live video with a moving camera and process it by the YOLOv5 model. Also, the robot system can receive images, videos, or YouTube links and process them with YOLOv5. Raspberry Pi is controlled remotely by connecting to the network through Wi-Fi locally or publicly using the internet with a remote desktop connection application. The results were very satisfactory and proved the high-performance efficiency of the robot.

Corresponding Author:

Maad Issa Al-Tameemi
Department of Computer Engineering, College of Engineering, University of Baghdad
Baghdad, Iraq
Email: maadesaa@gmail.com

1. INTRODUCTION

Recently, the use of mobile robots has become very necessary today because of their capabilities that can be used in several areas, including the field of surveillance, especially monitoring the places to be seen from a distance and determining the objects detection before entering these places. The authors designed an embedded system based on Rapsberry Pi to display live video on a web browser and make face detection without monitoring reactions [1], [2]. The Raspberry pi 3 controller and radio frequency identification are used to control the electronic lock door (E-Door) [3]. Many devices of the internet of things (IoT) are connected over the Internet to be controlled and managed by the users and thus get advantages of their data [4], [5].

YOLOv5 algorithm is used to detect Wheat Spike in unmanned aerial vehicle images [6], apples in orchards images as well as with YOLOv3 [7], Apple stem/calyx recognition in real-time [8], Face mask recognition [9], the mold on the food surface [10], the Ship in images of optical sensing [11]. Chest abnormality is detected based on YOLOv5 model by using ResNet50 controller [12]. A vehicle is designed to float on the different water bodies for collecting garbage floating automated by using you only look once model. This work is dependent mainly on Raspberry Pi 3 model B in order to control the overall vehicle [13]. Various technologies use one you look family (YOLO) algorithms based on machine and deep learning to detect smoke early in forests satellite imagery [14]. Object recognition, speech processing, and image classification technologies have been implemented using the Jetson Nano Board to detect obstacles for blind people [15].

The system is built using a Raspberry Pi 3 controller with an attached camera for object detection and recognition based on the YOLO algorithm in different places [16]. The authors have proposed an online assistive blind system for detecting the different objects based on YOLO algorithm and implementing by Raspberry Pi 3 Model B [17].

- The process of rover robot system

The main goal of making a live video with real-time object detection is to find a suitable object detection model in terms of accuracy and high speed as well as using a controller that implements this model in real-time without having to send data to the last place, let it be a high specification computer for processing data. Redmon *et al.* [18] a great development appeared in object detection, which is the creation of the YOLO algorithm. YOLOv2 YOLO9000 are then introduced to be better, faster, and stronger than YOLO in detecting objects, where YOLO9000 was the second "YOLOv2" trained and used to detect more than 9000 object categories [19]. YOLOv3 is made by updating the YOLOv2 to be faster than previous YOLO versions [20]. YOLOv4 is then created to improve YOLOv3 of real time object detection by achieving optimal speed and accuracy [21].

YOLOv5 is the last version of YOLO and used for detecting the objects with fast detection speed and exact precision, which gets and gives 72% AP 0.5 for the Common Objects in Context val2017 dataset [22]. Besides, YOLOv5 contains multi versions the minimum model size is YOLOv5s with 14 megabytes, which is convenient for deployment [23]. YOLOv4 has been proposed and compared with many object detectors, including EfficientDet and YOLOv3 where the proposed algorithm has been proven to have better performance than others and is 2 times faster than EfficientDet, as well as a significant improvement in YOLOv3 in terms of Average Precision and frames per second by 10% and 12% respectively [21].

Fang *et al.* [24] the authors are proposed a method based YOLOv5 model for the detection of surface knots on sawn timber. YOLOv3 spatial pyramid pool (SPP) and faster region-based convolutional neural network (R-CNN) were implemented on two datasets and compared with the YOLOv5 model. The experimental results showed that the YOLOv5 model has the best performance in terms of accuracy and speed [24].

A comparison was made between YOLOv5l, YOLOv4, and YOLOv3 algorithms for effective landing area detection, where it was found that YOLOv5l outperforms the rest of the algorithms in terms of accuracy and speed [25]. Thus, five versions of the YOLO algorithm appeared, while the last version was the best choice to perform real time object detection according to the results and comparisons with mentioned references. A mobile robotic surveillance system has been designed and implemented in both Java and python language based on Raspberry Pi 3 controller [26]. This system needs to create a client-server in the pc and a server-client in the controller of robot to transfer the data between them and make image processing in the pc because this controller cannot process data in real-time Due to its limited capabilities, which makes the system very slow. The system performs live streaming videos through two cameras and transfers the video frames to the computer to be processed by using the Harr cascade algorithm to detect the objects, and these processed frames with objects detection are then displayed directly on the screen of the pc [26].

The main contributions in this work, a remote-controlled mobile system proposes and distinguishes from the rest of the mentioned systems by the ability to broadcast live video of different places with a moving camera for detecting the objects in real-time, processing and displaying by Raspberry Pi model B itself without the need to send data to the computer with a high specification for processing. In addition, this system can detect objects in stored images, videos, or YouTube video links.

2. THE PROPOSED SYSTEM

The rover robotic monitoring system has been proposed, designed, and implemented to be able for moving long distances and be controlled locally and remotely to show live streaming monitoring video with object detection in real-time based on the YOLOv5 model using deep learning technique. Figure 1 illustrates overall design of robot system. The main robot system consists of three sub-systems, the first is the rover robotic sub-system used to move the robot in all directions, the second is the Raspberry pi 4 controller that is responsible for controlling all components associated with the main system, and the last one is monitoring subsystem based on YOLOv5 to make life streaming video with deep learning. Figure 2 shows the system components and architecture.

2.1. Rover robotic sub-system

A framework for building the rover robot consists of two parts that are responsible for managing the robot. This robot receives all the orders and performs all the functions through the work of these parts. The parts are listed as follows:

a) Hardware components

- Chassis and Actuators: used to contain the robot components.
- Direct current (DC) motors: the robot use four DC motors to be able to travel in all directions.
- L298N dual high bright: used to directly control the DC motors at the same time.
- Samsung batteries: used to supply the four DC motors with power.
- Webcam: used to capture the information of live video, images, and videos.
- Servo motor: used to raise the USB camera up and down.
- Stepper motor: used to rotate the camera left and right.
- ULN2003 driver board: is the interface between the Raspberry pi controller and stepper motor.
- Raspberry pi 4: used to control overall rover robotic monitoring system.
- General-purpose input/output (GPIO) interface: GPIO is used to connect external devices with the Raspberry Pi, and manage and control them.
- Remax power bank: used to power up the Raspberry Pi.

b) Software components

- Raspbian: Raspbian Linux operating system is used for the Raspberry Pi 4 controller.
- Python language version 3.9: used to develop the overall robot system because it is a good language and many available resources support it and used to implement the YOLOv5 model to make object detection in real-life streaming images and videos.
- Remote Desktop Connection is a technology that permits a computer to connect, gain access and take control of a raspberry pi over a network.

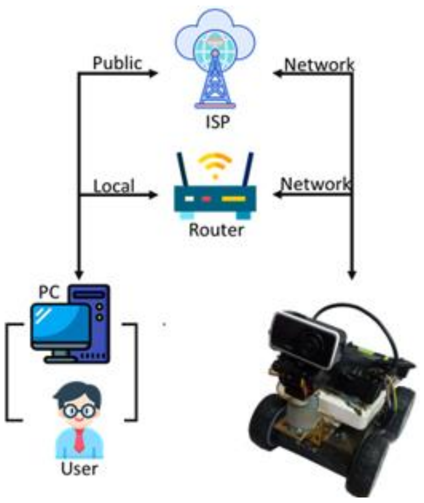


Figure 1. The overall design of robot system

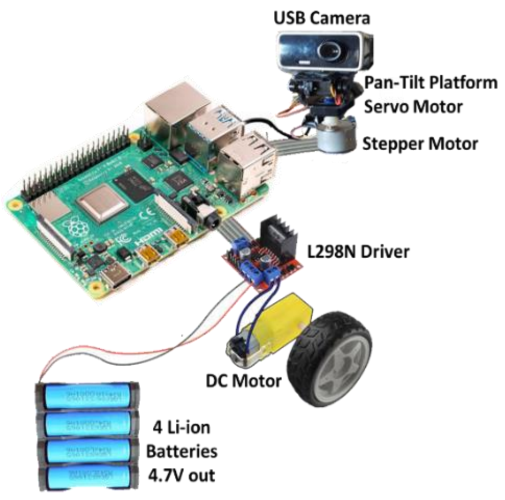


Figure 2. Hardware architecture

2.2. The raspberry pi 4 controller

Raspberry pi is an embedded computer that considers the heart of the pulse system. This manages and controls the entire robot with all its sub-systems and operations by receiving instructions from the user. The raspberry pi 4 have many important features and can be specified in the following list and shown in Figure 3: i) CPU 64-bit ARM Cortex-A72 (4×1.5 GHz), ii) GPU Broadcom Video Core VI, iii) RAM 4 GB LPDDR4 SDRAM (plus options for 2 GB and 8 GB), iv) Wireless local area LAN 2.4 GHz and 5 GHz IEEE 802.11b/g/n/ac wireless LAN, v) Bluetooth 5.0, vi) Ethernet Gigabit Ethernet, vii) 2× USB-A 3.0, viii) 2× USB-A 2.0, ix) GPIO Standard 40-pin GPIO header (fully backwards-compatible with previous boards), and x) Video 2× micro-HDMI ports (up to 4Kp60 supported).

DC motors, a stepper motor, and a servo motor are connected to the GPIO pins to perform the functions of the rover robotic system and the rotation of the camera horizontally and vertically with the surveillance system. A camera is installed with the controller to make real-time surveillance with object detection based on YOLOv5 model using deep learning technique. The YOLOv5 algorithm can also deal with and process the stored images and videos for detecting the objects. The used language to create the robot system is Python 3.9.

The robot side and user side connect each other by using either the local area network (LAN) or the public network (internet) using the remote desktop connection application. In the local network, both sides must be on the same network and their local Internet Protocol must be known to establish connection. In the public network, the public address should be known and recieved by internet service providers (ISPs) and anyone can access it via the internet.

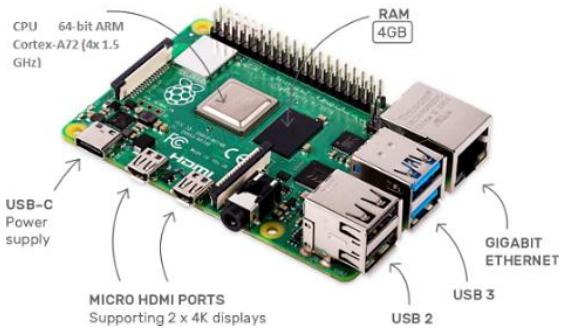

CPU 64-bit ARM Cortex-A72 (4x 1.5 GHz)
RAM 4GB
USB-C Power supply
GIGABIT ETHERNET
USB 3
MICRO HDMI PORTS Supporting 2 x 4K displays
USB 2

Figure 3. Raspberry Pi 4 model B

The terminal and status monitoring for the Raspberry Pi controller can be used remotely by remote desktop connection application, as well as be tunneled to any network services running on user Raspberry Pi (such as hypertext transfer protocol, secure shell protocol) and thus can be accessed worldwide over the internet. Begin by opening the remote desktop connection application on the windows computer and entering Raspberry Pi's local IP address to "Computer:" and clicking the "Connect" button. The PC will be connected to Raspberry Pi by receiving a screen from the xrdp software, which must be pre-installed in Raspberry Pi to allow remote connection. Then, the "username" and "password" of the account are entered which exist on your Raspberry Pi. After that, the Raspberry Pi screen will appear and the user will be able to manage and control the Raspberry Pi and all the devices connected with it.

2.3. Monitoring sub-system

The monitoring subsystem is designed and implemented in Raspberry Pi controller by using Python language. This subsystem can deal with either images, videos, and You Tube videos links or with live USB video camera associated with the controller to detect the objects. The USB camera is transmitting color video and scaled down to (320×320) resolution, in order to achieve low latency video streaming. This camera is installed on the robot by using a pan-tilt platform allowing the robot to rotate the camera horizontally at an angle of 360 degrees as well as vertically at an angle of 180 degrees to perform a full view. The user can control the camera remotely by connecting with the robot locally or publicly.

2.3.1. Object detection with YOLO

The monitoring subsystem uses the OpenCV for image and video processing by extracting useful information (frames) and then the YOLOv5 model for detecting objects in real-time. YOLOv5 is an ultralytics open-source research that pretrained the object detection architectures and models on the context object detection task (COCO) 2020 object detection task dataset [27] to take advantage of future vision AI strategies [22]. The structure of the YOLOv5 network contains three parts, which are the backbone, the neck, and the output as illustrated in Figure 4. In the first part, the input, which is an image with a resolution of 640×640×12, is received and sent to the focused structure. The splicing process is then used to become a 320×320×12 features map, and 32 convolution kernels are then applied to the convolution process to eventually become a 320×320×24 features map. The module Conv2D, Batch Norm, and leaky rectified linear unit (CBL) is considered as the basic convolution model. The module of BottleneckCSP is the second part and is designed to extract useful information on the feature map from the image.

This module structure is distinguished from others in that it reduces repetitive information in the optimization process of convolutional neural networks. By determining the length and width of the bottleneckCSP module, one can give four different versions, which are: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The YOLOv5 can add and combine a bottom-up feature pyramid structure based on the feature pyramid network (FPN) structure with the FPN layer which is from top to bottom features, and this allows the network to be able to detect the targets of different scales efficiently. Finally, the classification results and object detection will be obtained in the output as shown in Figure 4 [28].

The basic concept of the monitoring subsystem is to use YOLOv5s deep learning algorithm and implemented it on the live video connected with a USB camera or with the stored information of images, videos, or links of YouTube whether it is public or private for detecting the objects [22]. Figure 5 shows the main steps to implement the monitoring subsystem with objects detection in real-time. First, all packages, libraries, and frameworks necessary are imported by phyton language to implement the overall system. YOLO v5 model is then loaded by using the PyTorch framework which is an open-source machine learning for detecting the objects. The live video with a connected camera on the Raspberry Pi or stored images or videos is processed by using the OpenCV library and then sending to the YOLOv5s model. OpenCV gives a video capture object which handles everything identified with opening and closing the webcam. Creating the object is all we need to do and save all the frames from it. After that, the webcam is opened for captureing the frames and scale them, a loop is then used, and keep reading frames until press exit "q" key from the keyboard. Thus, the YOLOv5s model will make objects detection on these frames in real-time and then show them in a window or save it in the Raspberry Pi.

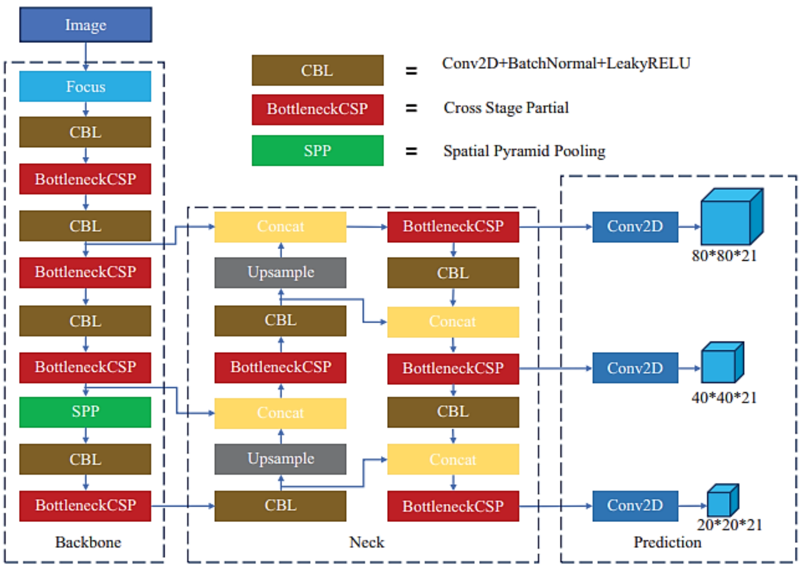


Image
Focus
CBL
BottleneckCSP
SPP
Backbone
Concat
Upsample
Neck
Conv2D
80*80*21
40*40*21
20*20*21
Prediction
CBL = Conv2D+BatchNormal+LeakyRELU
BottleneckCSP = Cross Stage Partial
SPP = Spatial Pyramid Pooling

Figure 4. The structure of the YOLOv5 network, which consists of the backbone, neck, and the output

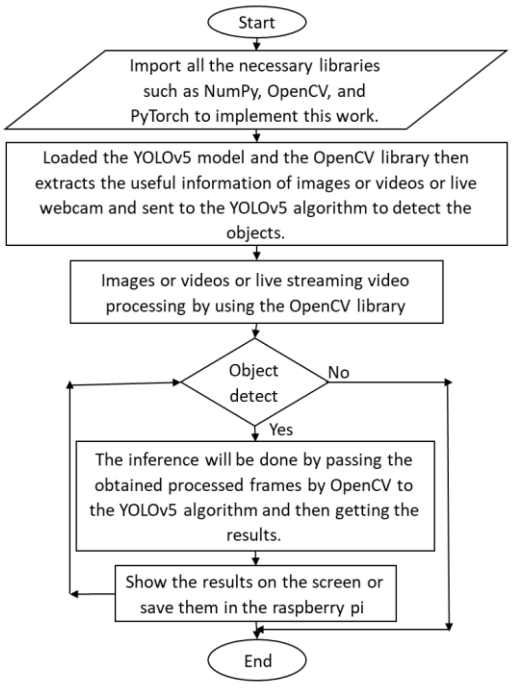

Start
Import all the necessary libraries such as NumPy, OpenCV, and PyTorch to implement this work.
Loaded the YOLOv5 model and the OpenCV library then extracts the useful information of images or videos or live webcam and sent to the YOLOv5 algorithm to detect the objects.
Images or videos or live streaming video processing by using the OpenCV library
Object detect
No
Yes
The inference will be done by passing the obtained processed frames by OpenCV to the YOLOv5 algorithm and then getting the results.
Show the results on the screen or save them in the raspberry pi
End

Figure 5. The flowchart of monitoring subsystem

3. RESULTS AND DISCUSSION

According to the previous explanations mentioned in the previous section, the robot consists of mobile camera fixed on a pan-tilt to perform a full view and a chassis based on four wheels, and these wheels are connected to four DC motors. The Raspberry Pi controls these motors allowing the robot to move forward and backward and rotate to the right and left by receiving commands from the user as Figure 6. The robotic system is created in python language and controlled remotely through a Wi-Fi 2.4 GHz wireless connection with TCP/IP protocol locally or publicly by using a remote desktop connection application.

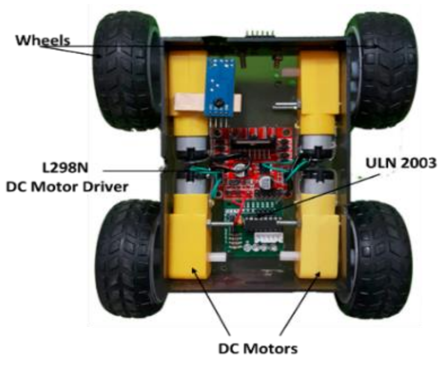


Figure 6. System structure

The robotic system has the ability to move in all directions, forward, backward, right, and left, as well as the ability to rotate the camera to the right, left, up, and down, in addition to the other functions that it detects the objects privatly or publicaly in images, videos, and live connected camera as shown in Figure 7. The keyboard control keys are used to perform the functions mentioned above, first, select the key ↑, ↓, ←, → to move the robot forward, backward, to the right or left and the keys R, F, D, G are used to control the pan-tilt and rotate the webcam up, down, to the left or right. Finally, the space key is pressed to stop the robot. The robot system is designed with several features to provide live video with object detection, recording abilities, or image and video processing with object detection. This system is based on the YOLOv5 model for detecting the objects in real-time by using one of the most used frameworks in the field of deep learning named PyTorch. The overall functions of the robot are controlled by Raspberry pi 4.

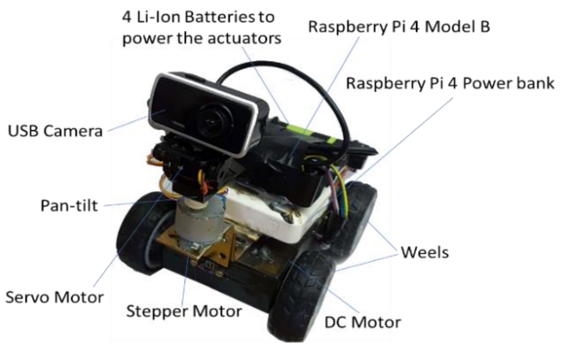
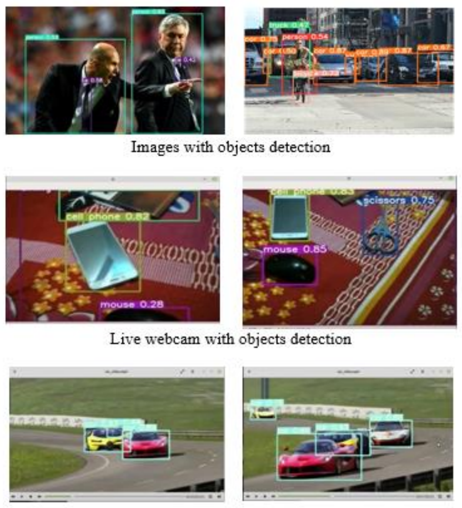


Figure 7. Main rover monitoring system

Despite the limited capabilities of the Raspberry Pi, compared to computers with high specifications of memory, graphics processing unit (GPU), and central processing unit (CPU), and in terms of the difficulty of applying deep learning algorithms in the Raspberry Pi, however, the application of the 5th generation of YOLO, YOLOv5 has become possible and has proven its efficiency in real time. YOLOv5s is implemented in this system by Python 3.9.0 with all requirements installed including PyTorch 1.7, and the models can be downloaded automatically from the YOLOv5 release. The system can detect the objects of a webcam in real time, image, video, directory, YouTube link privately or publicly [29]. The obtained results demonstrate the efficiency of this robot and the ability in detecting, classifying, and recognizing various objects in the images, videos, or live video in real-time with acceptable accuracies such as cars, persons, and cellphones by recording confidence scores of 0.85 for close objects and 0.55 for far objects as shown in the results in Figure 8.


Images with objects detection
cell phone 0.82
mouse 0.28
scissors 0.75
mouse 0.85
Live webcam with objects detection

Video with objects detection

Figure 8. Live video, images, video processing with objects detection

4. CONCLUSION

This work is proposed a rover robotic monitoring system by using Raspberry Pi 4 controlling with connected USB webcam and utilizing deep learning based YOLOv5s, OpenCV, PyTorch, and COCO 2020 object detection task dataset to detect the objects. The obtained results show the high efficiency in detecting and identifying the various object in the images, videos, or live video in real-time with acceptable accuracies. In conclusion, the overall work was implemented in python language by making a relatively low-price robot with a lot of features and functions which can perform more than one task at the same time including motion the robot with the detection of objects with live video. Further study, as extended of this work, a comparisons study can be applied with other datasets, different weather conditions, or algorithms.

REFERENCES

[1] S. Singh, P. Anap, Y. Bhaigade, and P. J. P. Chavan, "IP Camera Video surveillance using Raspberry Pi," *IJARCCE*, pp. 326–328, Feb. 2015, doi: 10.17148/IJARCCE.2015.4272.

[2] B. K. Oleiwi, "Scouting and controlling for mobile robot based Raspberry Pi 3," *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 1, pp. 79–83, Jan. 2019, doi: 10.1166/jctn.2019.7701.

[3] M. I. Younis, M. I. Al-Tameemi, and M. S. Hussein, "MPAES: a multiple-privileges access e-door system based on passive RFID technology," 2017.

[4] S. Saha and A. Majumdar, "Data centre temperature monitoring with ESP8266 based wireless sensor network and cloud based dashboard with real time alert system," in *2017 Devices for Integrated Circuit (DevIC)*, Mar. 2017, pp. 307–310, doi: 10.1109/DEVIC.2017.8073958.

[5] P. Singh and S. Saikia, "Arduino-based smart irrigation using water flow sensor, soil moisture sensor, temperature sensor and ESP8266 WiFi module," in *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Dec. 2016, pp. 1–4, doi: 10.1109/R10-HTC.2016.7906792.

[6] J. Zhao *et al.*, "A wheat spike detection method in UAV images based on improved YOLOv5," *Remote Sensing*, vol. 13, no. 16, p. 3095, Aug. 2021, doi: 10.3390/rs13163095.

[7] A. Kuznetsova, T. Maleva, and V. Soloviev, "Detecting apples in orchards using YOLOv3 and YOLOv5 in general and close-up images," 2020, pp. 233–243.

[8] Z. Wang, L. Jin, S. Wang, and H. Xu, "Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system," *Postharvest Biology and Technology*, vol. 185, p. 111808, Mar. 2022, doi: 10.1016/j.postharvbio.2021.111808.

[9] G. Yang *et al.*, "Face mask recognition system with YOLOV5 based on image recognition," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, Dec. 2020, pp. 1398–1404, doi: 10.1109/ICCC51575.2020.9345042.

[10] F. Jubayer *et al.*, "Detection of mold on the food surface using YOLOv5," *Current Research in Food Science*, vol. 4, pp. 724–728, 2021, doi: 10.1016/j.crfs.2021.10.003.

[11] Y. Chen, C. Zhang, T. Qiao, J. Xiong, and B. Liu, "Ship detection in optical sensing images based on YOLOv5," in *Twelfth International Conference on Graphics and Image Processing (ICGIP 2020)*, Jan. 2021, p. 61, doi: 10.1117/12.2589395.

[12] Y. Luo, Y. Zhang, X. Sun, H. Dai, and X. Chen, "Intelligent solutions in chest abnormality detection based on YOLOv5 and ResNet50," *Journal of Healthcare Engineering*, vol. 2021, pp. 1–11, Oct. 2021, doi: 10.1155/2021/2267635.

[13] C. Patil, S. Tanpure, A. Lohiya, S. Pawar, and P. Mohite, "Autonomous amphibious vehicle for monitoring and collecting marine debris," in *2020 5th International Conference on Robotics and Automation Engineering (ICRAE)*, Nov. 2020, pp. 163–168, doi: 10.1109/ICRAE50850.2020.9310888.
[14] C.-L. C. Huang and T. Munasinghe, "Exploring various applicable techniques to detect smoke on the satellite images," in *2020 IEEE International Conference on Big Data (Big Data)*, Dec. 2020, pp. 5703–5705, doi: 10.1109/BigData50022.2020.9378466.
[15] R. Joshi, M. Tripathi, A. Kumar, and M. S. Gaur, "Object recognition and classification system for visually impaired," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, Jul. 2020, pp. 1568–1572, doi: 10.1109/ICCSP48568.2020.9182077.
[16] H. Gupta, R. S. Yadav, S. M. S. Kumar, and M. J. Leo, "A novel trespassing detection system using deep networks," 2021, pp. 633–645.
[17] M. M. Abdul, F. Alkhalid, and B. K. Oleiwi, "Online blind assistive system using object recognition," *International Research Journal of Innovations in Engineering and Technology*, vol. 3, no. 12, pp. 47–51, 2019.
[18] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015.
[19] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.
[20] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, doi: 10.48550/ARXIV.1804.02767.
[21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: optimal speed and accuracy of object detection," *arXiv*, 2020, [Online]. Available: http://arxiv.org/abs/2004.10934.
[22] "Ultralytics-Yolov5," *Githup*. Accessed: Jan. 01, 2021. [Online]. Available: https://github.com/ultralytics/yolov5.
[23] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, p. 217, Feb. 2021, doi: 10.3390/f12020217.
[24] Y. Fang, X. Guo, K. Chen, Z. Zhou, and Q. Ye, "Accurate and automated detection of surface knots on sawn timbers using YOLO-V5 model," *BioResources*, vol. 16, no. 3, pp. 5390–5406, 2021.
[25] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs," *Sensors*, vol. 22, no. 2, p. 464, Jan. 2022, doi: 10.3390/s22020464.
[26] M. I. AL-TAMEEMI, "RMSRS: Rover multi-purpose surveillance robotic system," *Baghdad Science Journal*, vol. 17, no. 3(Suppl.), p. 1049, Sep. 2020, doi: 10.21123/bsj.2020.17.3(Suppl.).1049.
[27] T.-Y. Lin *et al.*, "Microsoft COCO: common objects in context," in *In: European conference on computer vision*, May 2014, pp. 740–755, [Online]. Available: http://arxiv.org/abs/1405.0312.
[28] F. Zhou, H. Zhao, and Z. Nie, "Safety helmet detection based on YOLOv5," in *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, Jan. 2021, pp. 6–11, doi: 10.1109/ICPECA51329.2021.9362711.
[29] G. Jocher, "ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration," 2021, doi: 10.5281/zenodo.4418161.

BIOGRAPHIES OF AUTHORS

Maad Issa Al-Tameemi is a faculty member - lecturer at the University of Baghdad since 2015. His research interests are systems and networks engineering, intelligent systems, mobile robotics, robotics, internet of things, wireless sensor networks, computer networks, routing protocols, and transport protocols. Furthermore, he has published many research papers and served as a reviewer in highly reputable conferences and journals. He is the managing editor of the Association of Arab Universities Journal of Engineering Sciences as well as the Director of the Cisco Academy for Computer Networks at the College of Engineering, University of Baghdad. He can be contacted at email: maadesaa@gmail.com.

Ammar A. Hasan born in Baghdad, He is currently working as a lecturer in Computer Engineering Department at University of Baghdad. His research interests are control and computer, mobile robotics, digital system and technology, evolutionary algorithms and embedded systems. He has published a lot of articles and served as a reviewer in many journals. He can be contacted at email: mr.ammaradel@coeng.uobaghdad.edu.iq.

Bashra Kadhim Oleiwi born in Baghdad, she is currently working as a lecturer in control and systems Engineering Department at University of Technology. Her research interests are mechatronics, robotics, systems dynamics, and control and instrumentation. She has published a lot of articles in many journals. She can be contacted at email: 60010@uotechnology.edu.iq.