Mediterranean Smart Cities Conference (MSCC 2024)
2-4 May, 2024, Martil, Morocco

New Approach: Intelligent System-Based Object Detection in Smart Parking

Mohamed hamza Boulaich
National High School for Computer Science and Analysis,
Mohamed V university in Rabat
Rabat, Morocco
Mohamedhamza_boulaich@um5.ac.ma

Said Ohamouddou
National High School for Computer Science and Analysis,
Mohamed V university in Rabat
Rabat,Morocco
said_ohamouddou@um5.ac.ma

Hibatallah Meliani
National High School for Computer Science and Analysis,
Mohamed V university in Rabat
Rabat,Morocco
hibatallah_meliani@um5.ac.ma

Mohammed Ali Ennasar
National High School for Computer Science and Analysis,
Mohamed V university in Rabat
Rabat, Morocco
mohammed-ali_ennasar@um5.ac.ma

Abdellatif El Afia
National High School for Computer Science and Analysis,
Mohamed V university in Rabat
Rabat, Morocco
abdellatif.elafia@ensias.um5.ac.ma

Abstract—**We propose a cutting-edge Internet of Things (IoT)-powered intelligent parking system and employs a Raspberry Pi model-based camera for accurate vehicle detection. A wireless sensor network is used to gather machine-to-machine (M2M) data from the Raspberry Pi and transmit it to the server using the Message Queuing Telemetry Transport (MQTT) communication protocol. The model has been trained using deep learning techniques.**

Keywords—roadside parking, vehicle detection, IoT, raspberry Pi. Wireless Sensor Network, YOLO v7.

I. Introduction

Roadside parking on public streets and Urban areas worldwide grapple with the significant issue of traffic congestion. Frequently, this results in significant disruptions to vital services such as emergency response units, ambulances, and fire services [1]. It has adverse effects on individuals, businesses, and the environment [2]. As urban areas expand and the quantity of vehicles on the streets rises, the negative consequences of traffic congestion are not only experienced in terms of time wasted during travel, but also in its influence on vital services, public safety, and the overall standard of urban living. Autonomous driving technology is a ground-breaking advancement in transportation that envisions a future when vehicles can navigate and function without human involvement. This shift goes beyond individual cars and includes a vision of future interconnected and intelligent urban ecosystems called smart cities. In traditional roadside parking there are many sorts of trucks halting in double-position on the street while load or unload operations, this scenario happens regularly on everyday especially on narrow street in downtown. To tackle these challenges, we need to employ a strong combination of hardware components such as microco-ntrollers, sensor, and communication modules. At Aspect Software, we need to build a target detection algorithm that is capable of detecting and differentiating between object. Figure 1 displays an actual situation demonstrating the issues provided by the loading and unloading activities of delivery trucks by the roadside. Deep learning offers a powerful way to detect different vehicles using a target detection algorithm which is normally used in one- and two-stages detectors [3]. two-stage approach uses a Region Proposal Network (RPN) to retrieve the selected area object locations, then refines them for accurate bounding boxes and class labels, including R-CNN algorithm, SPP-Net algorithm, fast R-CNN [4]-[5].

Figure 1: Real-world Example of Double-Parking at Roadside

The paper is constructed as follows: Section 2 presents a summary of key research publications concentrating on traffic congestion applying Deep learning techni-ques. Section 3 describes the system model, defines the proposed approach, and introduces the metrics determ-ining the accuracy of model identification. Section 4 includes results and discussion. Finally, part 5 concludes the paper.

II. Related Work and Contribution

In recent years, various researches have investigated into smart parking and traffic management. However, standard approaches fall short in correctly address the complicated optimization difficulties faced by severe traffic congestion. M. Veeramanickam et al, [11] designed smart parking system-based internet of thing (IoT) employing Arduino UNO's model in implementation, adding ultrasonic sensor for parking management. This proposed model assists users to facilitate identification of parking places, the system runs on a First come First serve (FCFS).

Amara Aditya et al, [13] designed intelligent parking systems employing IoT based sensor ultrasonic and radio frequency identification to detect the parking space and utilize mobile application that enables users to verify the availability of adjacent parking space. Hence, in the last several years traffic congestion has utilized artificial intelligent (AI) techniques such as fuzzy logic [10], and YOLO family based deep learning [12]. Ho, George to Sum, et al. [10] was employed to solve the fuzziness in vehicle parking reservations and estimate the period of stay of the vehicles. The proposed system can evaluate average space utilization, loading and unloading activities, and the average parking waiting times. In [12], YOLOv5 has been used to identify and target detection objects of interest in images or real-time video with high accuracy and speed, refine the feature extraction network of YOLOv5, add a small target detection head, and improve the ability of the backbone network to extract target features.

A self-adaptive Fuzzy Mamdani Multi-objective Simu-lated Annealing method has been used by several researchers. Many research has used fuzzy logic to control the settings of Simulated Annealing (SA). While [22] directly modified the annealing temperature, earlier research, including [21], concentrated on adjusting the acceptance probability and current temperature. Furthermore, [23] employed Mamdani fuzzy controllers to regulate SA's cooling rate, while [24] used the same controllers to regulate SA's neighborhood structure. In [25], type 2 fuzzy logic was used to control the acceptance probability and current temperature. The application of fuzzy logic extends to parameter control in population-based metaheuristics. For example, it was used in [26] to modify Ant Colony System (ACS) population size, and in [27] and [28][29] to modify local pheromone and pheromone parameters, respectively.

Techniques such as the Hidden Markov Model (HMM) have been utilized to regulate SA parameters, this was demonstrated in [30][35][36], where the cooling schedule was modified using the HMM. Moreover, HMM was used in [31] to regulate SA's neighborhood structure. As mentioned in [32][33][34], HMM has also been applied to other metaheuristics, such as ACS, where it adaptively controlled pheromone parameters, pheromone level exponent, and associated parameters.

Main contribution: This paper proposes a system to control roadside parking space utilization owing to load and unload activity in order to prevent traffic congestion in the city's downtown. We employ affordable tools, simple-to-install Internet of Things devices. Our YOLOv7 algorithm is executed at a high computing power. A working prototype is developed to evaluate the system's functionality and viability for use in smart parking scenarios. This study aims to address the requirements of smart parking, which include less traffic congestion, low-cost applications, and ease interface. Furthermore, in this work, we present the use of Fuzzy Mamdani controllers in the Multi-objective Simulated Annealing algorithm to dynamically tune the temperature and the acceptance probability.

III. THE PROPOSED APPROACH

This study offers a system built to run on an automated IoT platform, using sensor data analysis to increase operational efficiency. Hence, the YOLOv7 model underwent training, validation, and testing via Jupyter notebook platform within anaconda. We used, deep learning framework based pytorch that provides the building blocks for designing and training neural networks. It contains functions and classes for manipulating tensors, building computational graphs, and implementting various machine learning algorithms [15].

A. Methodology

The fundamental purpose of this initiative is to design a complete solution to solve smart parking management and decrease traffic congestion, especially to avoid traffic jams on narrow streets in big cities. As our proposed model is not simple for categorizing cars. It is proposed after thoroughly considering all aspects and restrictions of object detection. Figure 2 shows a less complicated system approach.

- First, before coming, every user must send a message by Module GSM, In order to verify that the place is empty or not. For instant ("check: Franklin Street N°1041"), the Module GSM transmit the letter into the server.
- Second, we implement an IoT model-based Raspberry Pi with an average-definition smart camera to check if the space is vacant or full.
- Third, deliver the data to the server via MQTT communication protocols.
- Finally, after treating the acquired data and assigning the object inside the image, it will change this information via Module GSM for an instant ("You can access the specific area") or negative form.

The workstation was utilized to conduct our experimentation, as detailed in Table 1.

Table 1: Workstation Environmental Configuration

Components	**Description**
GPU	NVIDIA RTX 3080
CPU	AMD Ryzen Serie 7 7700X
RAM	64 GB (32 GB X 2)
OS	Ubuntu 23.10 64 bit
CUDA	11.8
Pytorch	2.2.
Python	3.8.0

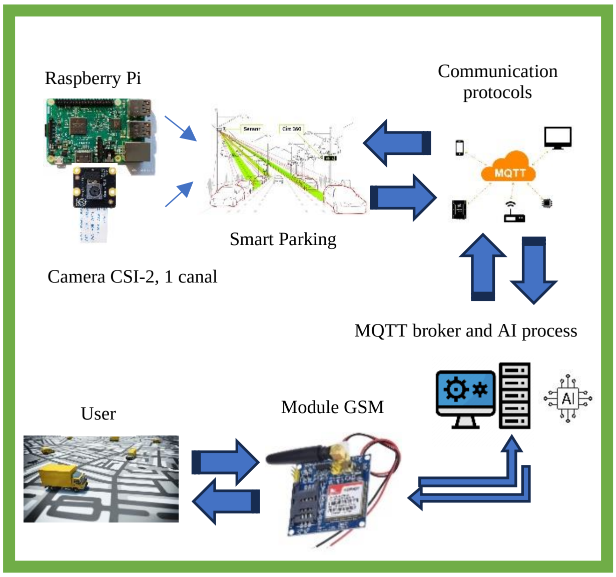

Figure 2: The architecture of the smart parking system

B. Yolo version 7 (YOLOv7)

We utilize the YOLOv7 algorithm [14], which contains High mean average precision (mAP) and comes through in recognizing small items better than previously used family YOLO. YOLOv7 target detector, whose important features will revolutionize in the area of computer vision in 2022, the official YOLOv7 stands out for its extraordinary speed and accuracy, exceeding its predecessors.

C. Performance Evaluation Metrics

Several assessment criteria are applied in YOLO and its derivation models for object detection. In this article, we propose precision (P), recall (R), and mean average precision (mAP) as the evaluation metrics to examine the accuracy of model identification.

- **Precision (P) :** This metric measures the proportion of successfully detected object out of all the objects the model has predicted as positive. It can be determined using the formula given in "Eq. (1)" [7].

$$P = \frac{TP}{TP + FP} \tag{1}$$

TP: True Positive, FP: False Positive

- **Recall (R):** This measure shows the fraction of true positives detected out of all actual items contained in the image. It provides an intuitive measure of the model's comprehensiveness. Recall can be determined using the formula presented in "Eq. (2)" [7].

$$R = \frac{TP}{TP + FN} \tag{2}$$

TP: True Positive, FN: False Negative

- **Mean Average Precision (mAP):** This metric determines the average precision (AP) over varying amounts of overlap between predicted and ground truth bounding boxes. A greater mAP indicates better overall object detection performance. The calculation for mAP is illustrated in "Eq. (3)" [7].

$$mAP = \frac{1}{N}\sum_{N}^{1} P_i \tag{3}$$

N: The number of classes

D. Dataset and Hyperparameter Tuning Algorithm

In this study, we employ a dataset obtained from the Roboflow site, namely Udacity self-driving car dataset for YOLOv7 Python [16]. The collection consists of 97,942 labels across 11 classes and 15,000 pictures. A downscaled version to 512x512 (with a download size of around ~580 MB) has been released, which is compatible with most common machine learning models, including YOLOv7. In this data, we are interested in the classes of cars and trucks that contain 64.399 and 3.623 labels, respectively.
Considering the problem of optimizing traffic congestion due to roadside activities, and with the purpose of maximizing the precision and recall of the model YOLOv7. We fine-tuned the hyperparameters, specifically adjusting the learning rate and momentum. Figure. 4 depicts the suggested technique for upgrading the model. In our proposed methodology, we optimized the YOLO model using hyperparameter tuning [17].

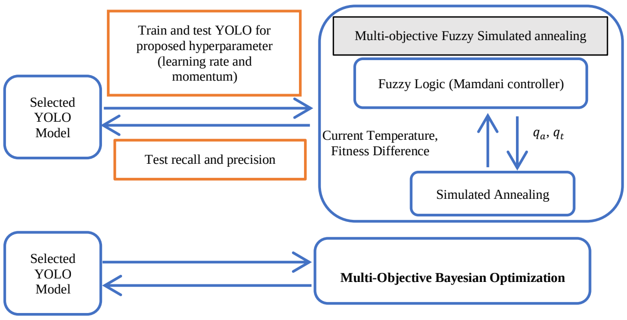

Figure 4: Proposed Method for Optimizing the YOLO Model via Tuning Learning Rate and Momentum Hyperparameters in the Adam Optimizer.

Multi-objective Fuzzy Simulated annealing (MOFSA): This algorithm takes both multi-objective simulated annealing (MOSA) and fuzzy logic (FL) concepts to a new level, specifically designed for optimization problems. In our work, we use fuzzy membership functions, which translate objective values into "degrees of satisfaction" between 0 and 1, representing how well a solution meets each objective (e.g., "highly efficient," "moderately fast"). And simulated annealing is the basic MOSA mechanism that directs the search process. New solutions are randomly produced and accepted based on a likelihood of being impacted by their fuzzy fitness and a "temperature" value (first high, then lowering). In order to achieve the best outcome of the model's YOLOv7, we contrasted MOFSA with Bayesian optimization.

Multi-Objective Bayesian Optimization (MOBO): is a sophisticated approach for solving optimization problems with multiple, often conflicting objectives such as Define Objectives, Prior Knowledge and Surrogate Model, Acquisition Function, Sample the Next Point, Update the Model and iterate.

Algorithm: Multi-Objective Fuzzy Simulated Annealing

Input: Objective functions f and g, Maximum number of iteration($MaxIter$), initial temperature T_init
Output: A set of best solutions for the objective functions f and g
Initialize parameters T_init, qa (acceptance parameter), qt (temperature parameter), A
$T \leftarrow T$_init
S ← generateSolution()
updateArchive(S)
for i ← 1 to $MaxIter$ do
    S' ← generateSolution()
    ΔE ← [(f(S') + g(S')) – (f(S) + g(S))]/2
    If S ≻ S'   // current dominates new solution
        If (min {1, [1 - (1 - qa)ΔE]$^{1/(1-qa)}$} > $random$[0,1]
            S ← S'
    else
        If S≻a, a ∈ A   // new solution dominates archive
            S ← S'
            updateArchive(S')
        elseif a ≻ S', a ∈ A
            // archive solutions dominate new solution
            a* ← selectRandomArchiveSolution()
            S ← select(a*, S', S)
        else
            //new solution doesn't dominate or is not dominated by any archive solution
            S ← S'
            updateArchive(S')
        end
    end
    qt, qa ←fuzzyLogicController(T, ΔE)
    T ← T_init * $\frac{2^{qt-1}-1}{(1+i)^{qt-1}-1}$
end

Algorithm: Bayesian Optimization

Assuming Goal is to maximize unknown function f(x) on data D:
For n loops D
    Select new x_{n+1} by optimization of α which is an aquisation function
    $x_{n+1} = max\ \ \alpha\ (\mathrm{x};\ D_n)$
    Get new observation y_{n+1} from objective function
    Augment data $D_{n+1} = \{D_n,\ (x_{n+1}, y_{n+1})\}$
    Update model
End for

IV. EXPERIMENTAL RESULTS

In this experiment, we did a comparative examination of three training models: YOLOv7, YOLOv7 with Optimization utilizing Multi-Objective Fuzzy Simulated Annealing, and YOLOv7 with Multi-Objective Bayesian Optimization. Figure 5. displays the test images recognized using YOLOv7 models. After training, we have acquired comprehensive results in this model. Figure. 6 illustrates the outcomes of training YOLOv7. We have lowered the dataset size for both training and testing to 8000 and 2000, respectively. The dataset now consists of only two classes: "car" and "truck." In the initial training step, we employed the whole dataset, including 11 classes, utilizing 100 epochs to train. Subsequently, we saved the file containing the best model, with the training process taking approximately 8 hours and 25 minutes. To expedite the training period, we decreased the dataset size for training and testing. Additionally, we cut down the designations, using only class 1 for vehicles and class 10 for trucks. Following this tweak, we trained our model for 20 epochs using the preserved best.pt file.
The algorithm of fuzzy multi-objective simulated annealing has not given us the ideal result in terms of both recall and precision. We have tried to increase the temperature and number of iterations to 5000 and 100,

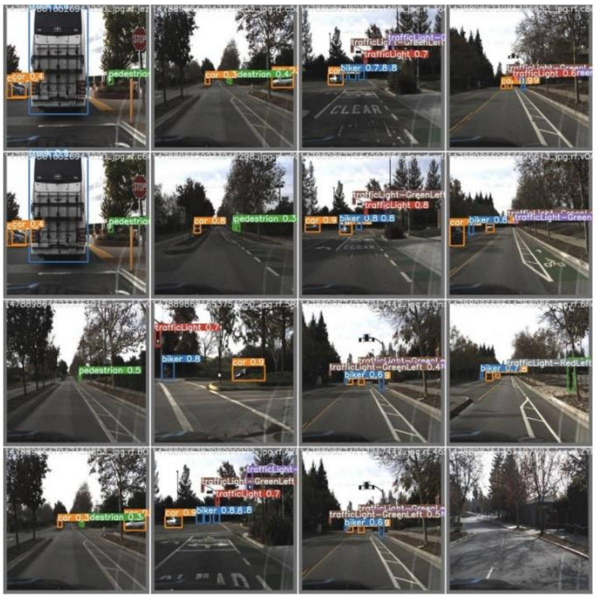

Figure 5: Detection outputs tested image using YOLOv7

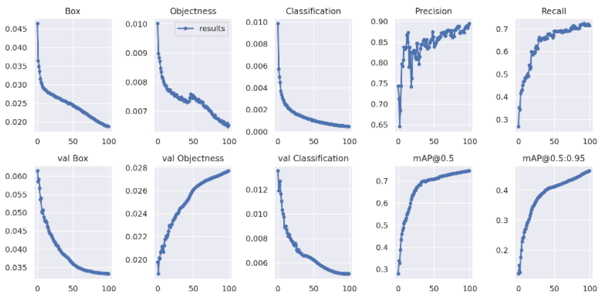

Figure 6: the results of training a YOLOv7 object detection model

respectively, but the results were unstable. Figure 7 demonstrates the performance of the MOFSA based on recall and precision (the x-axis denotes the number of iterations). The figures don't consist of a positive or negative slope for either precision or recall throughout all iterations. This shows that the algorithm might not have converged on a single optimal solution within the 100 iterations. The precision tends to fluctuate more considerably in the initial iterations (up to 50 iteration or so), and the recall also fluctuates during the iterations, but the variances seem more constant compared to precision.

We compared these results using Multi-Objective Bayesian Optimization (MOBO). Figure 8. depicts the evolution of model accuracy during MOBO for 35 iterations.

The accuracy tends to rise as the number of iterations progresses. However, there are occasional fluctuations. This shows that the Bayesian optimization process is identifying better hyperparameter combinations that lead to improved model accuracy.

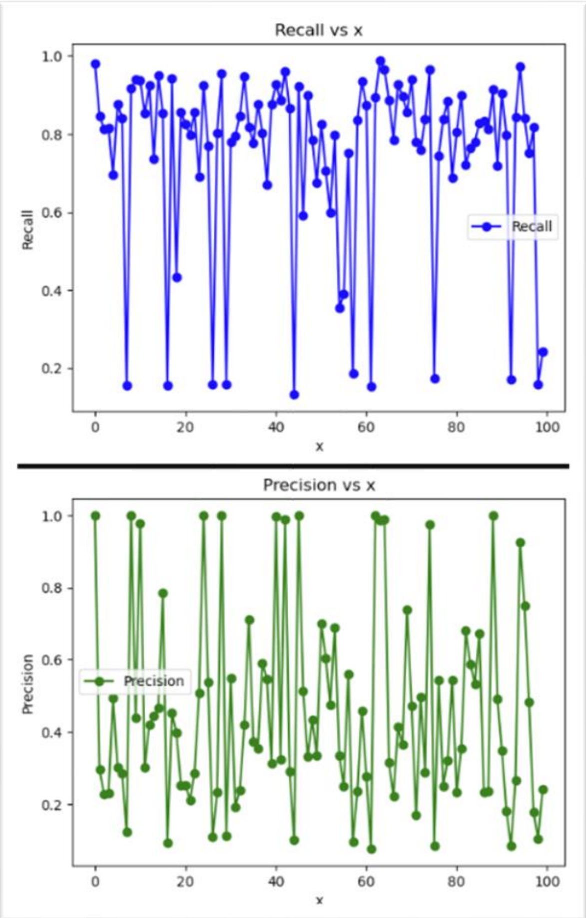

Figure 7: The precision-recall curve of YOLOv7-MOFSA

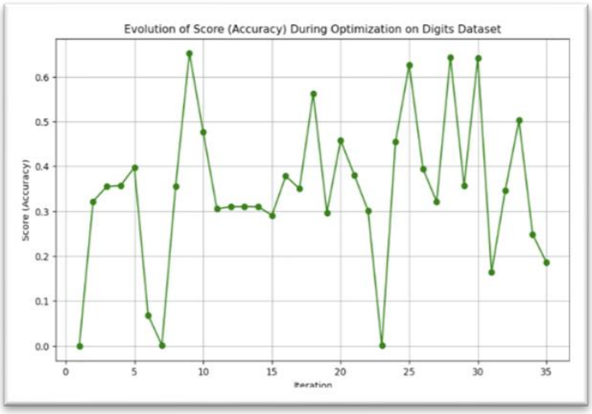

Figure 8: Improving model accuracy through multi-objective Bayesian optimization.

The overall positive slope is a favorable sign, indicating that the optimization is typically on the correct path. It's desirable in Bayesian optimization because it shows that the algorithm is identifying hyperparameter settings that boost model performance. Table 1. gives the performance of YOLO models trained on different algorithms of optimization. Compared to the other models, the YOLOv7 model offered high values of precision after 96 epochs, whereas the algorithm of optimization MOFSA gave 0.249 epochs of precision after 96 iterations, and the number of epochs was fixed at 20 epochs. The best model of optimization was YOLOv7 with MOBO, which gave 0.682 after nine iterations, and the number of epochs was fixed at 20 epochs.

Table 1. The performance of YOLO models trained on different optimization.

Model: YOLOv7				
Number of Epochs	Precision	Recall	mAP@0.5	F1-score
09/99	0.645	0.344	0.327	-
96/99	0.889	0.717	0.746	-
Model: YOLOv7 with MOFSA				
Number of iterations	Precision	Recall	mAP@0.5	F1-score
09/99	0.0015	0.08	0.000435	-
96/99	0.249	0.159	0.0699	-
Model: YOLOv7 with MOBO				
Number of iterations	Precision	Recall	mAP@0.5	F1-score
09/35	0.682	0.626	0.615	0.652
31/35	0.832	0.218	0.262	0.346

Furthermore, we conducted a comparative analysis between our proposal and another experimental approach, the majority of which utilized YOLOv5 to address the identical issue of detecting pedestrians and automobiles on roadsides. We selected three metrics, namely precision, recall, and mAP@0.5, for the purpose of comparison. Table 2 shows the comparison with three metrics.

Table 2. Comparison of different algorithms.

Algorithms	Metrics		
	P	R	mAP@0.5
YOLOv5[12]	82.9%	70.5%	74.9%
Our proposed (YOLOv7)	88.9%	71.7%	74.6%
Our proposed (YOLOv7 with MOBO)	83.2%	21.8%	26.2%

V. Conclusion and Future Work

The suggested intelligent system, based on an IoT prototype for object identification in smart parking, includes a Raspberry Pi, and MQTT communication protocol between machine-to-machine (M2M), and a

GSM SIM900 module for communication between the server and the user. The software component leverages YOLOv7 for object detection, paired with advanced optimization methods, particularly fuzzy multi-objective simulated annealing and Bayesian multi-objective optimization, for hyperparameter tuning. Upon thorough examination, it was revealed that Bayesian optimization multi-objective outperformed fuzzy multi-objective simulated annealing in terms of convergence efficiency and attained superior precision, recall, and F1-score values. The integration of YOLOv7 and these optimization techniques enhances the system's capability for precise and efficient object recognition in smart parking scenarios.

In our future work, we will investigate dynamic optimization strategies that can change the hyper-parameters of YOLOv7 in real-time depending on changing environmental conditions or varying object detection requirements. enhancing the user interface, maybe incorporating a mobile application, to give users real-time parking availability information, warnings, and user-friendly engagement.

REFERENCE

[1] Navarro-Espinoza, Alfonso, et al. "Traffic flow prediction for smart traffic lights using machine learning algorithms." Technologies 10. 1 (2022): 5.

[2] Ho, George To Sum, et al. "A computer vision-based roadside occupation surveillance system for intelligent transport in smart cities." Sensors 19.8 (2019): 1796.

[3] Ahmed, Muhammad, et al. "Survey and performance analysis of deep learning based object detection in challenging environments." Sensors 21.15 (2021): 5116.

[4] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[5] He, Kaiming, et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition." IEEE transactions on pattern analysis and machine intelligence 37.9 (2015): 1904-1916.

[6] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934 (2020).

[7] Sirisha, U., et al. "Statistical analysis of design aspects of various YOLO-based deep learning models for object detection." International Journal of Computational Intelligence Systems 16.1 (2023): 126.

[8] Joo, Hyunjin, Syed Hassan Ahmed, and Yujin Lim. "Traffic signal control for smart cities using reinforcement learning." Computer Communications 154 (2020): 324-330.

[9] Khanh, Quy Vu, et al. "Wireless communication technologies for IoT in 5G: Vision, applications, and challenges." Wireless Communications and Mobile Computing 2022 (2022): 1-12.

[10] Ho, George To Sum, et al. "A computer vision-based roadside occupation surveillance system for intelligent transport in smart cities." Sensors 19.8 (2019): 1796.

[11] Veeramanickam, M. R. M., et al. "IoT based smart parking model using Arduino UNO with FCFS priority scheduling." Measurement: Sensors 24 (2022): 100524.

[12] Li, Qirui, et al. "Object Detection of Roadside Pedestrians and Vehicles Based on Improved YOLO v5." Focus 608.608: 3.

[13] Aditya, Amara, et al. "An IoT assisted Intelligent Parking System (IPS) for Smart Cities." Procedia Computer Science 218 (2023): 1045-1054.

[14] Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

[15] Imambi, Sagar, Kolla Bhanu Prakash, and G. R. Kanagachidambaresan. "PyTorch." Programming with Ten-sorFlow: Solution for Edge Computing Applications (2021): 87-104.

[16] https://public.roboflow.com/object-detection/self-driving-car

[17] Isa, Iza Sazanita, et al. "Optimizing the hyperparameter tuning of YOLOv5 for underwater detection." IEEE Access 10 (2022): 52818-52831.

[18] Sharma, Himanshu, Ahteshamul Haque, and Frede Blaabjerg. "Machine learning in wireless sensor networks for smart cities: a survey." Electronics 10.9 (2021): 1012.

[19] Zhang, Shuo, et al. "Tiny YOLO optimization oriented bus passenger object detection." Chinese Journal of Electroni-cs 29.1 (2020): 132-138.

[20] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (20 14).

[21] Lalaoui, M., & El Afia, A. E. (2018). A Fuzzy generalized simulated annealing for a simple assembly line balancing problem. IFAC-PapersOnLine, 51(32), 600–605.

[22] Qin, L., Wang, J., Li, H., Sun, Y., & Li, S. (2017). An Approach to Improve the Performance of Simulated Annealing Algorithm Utilizing the Variable Universe Adaptive Fuzzy Logic System. IEEE Access, 5, 18155–18165.

[23] Lalaoui M., A. El Afia, R. Chiheb, (2017). Fuzzy logic controller for an adaptive Huang cooling of simulated annealing, The 2nd International Conference on Big Data, Cloud and Applications (CloudTech'17) IEEE Conference, Rabat, Morocco. https://doi.org/10.1145/3090354.3090420

[24] Lalaoui, M., El Afia, A., & Chiheb, R. (2018). Simulated Annealing with Adaptive Neighborhood using Fuzzy Logic Controller. Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications - LOPAL ’18

[25] Lalaoui M., A. El Afia, (2018), A Versatile Generalized Simulated Annealing Using Type-2 Fuzzy Controller for the Mixed-Model Assembly Line Balancing Problem, the 9th IFAC Conference on Manufacturing Modelling, Management and Control, Berlin, Germany, August 28-30, 2019. https://doi.org/10.1016/j.ifacol.2019.11.633.

[26] Bouzbita, S., El Afia, A., and Faizi, R. (2019, September). Adjusting Population Size of Ant Colony System Using Fuzzy Logic Controller. In International Conference on Computational Collective Intelligence (pp. 309-320). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-28374-2_27

[27]El Afia, A., Bouzbita, S., and Faizi, R. (2017). The effect of updating the local pheromone on acs performance using fuzzy logic. International Journal of Electrical and Computer Engineering, 7(4), 2161. DOI: 10.11591/ijece.v7i3

[28] Bouzbita, S., El Afia, A., and Faizi, R. (2020). The behaviour of ACS-TSP algorithm when adapting both pheromone parameters using fuzzy logic controller. International Journal of Electrical and Computer Engineering, 10(5), 5436. DOI: 10.11591/ijece.v10i5.

[29] Bouzbita, S., El Afia, A. (2022). A Fuzzy Meta Model for Adjusting Ant Colony System Parameters. In book: Proceedings of the 5th International Conference on Big Data and Internet of Things (pp.48-58). DOI:10.1007/978-3-031-07969-6_4

[30] Lalaoui M., A. El Afia, R. Chiheb.(2017). A Self-Tuned Simulated Annealing Algorithm Using Hidden Markov Model, International Journal of Electrical and Computer Engineering,(IJECE). http://doi.org/10.11591/ijece.v8i1.pp291-298

[31] Lalaoui M., A. El Afia and R. Chiheb, (2016). Hidden Markov Model for a Self-Learning of Simulated Annealing Cooling Law, the 5th International Conference on Multimedia Computing and Systems IEEE Conference, ICMCS'16, Marrakech,.Morocco..https://doi.org/10.1109/ICMCS.2016.7 905557

[32] Bouzbita, S., El Afia, A., and Faizi, R. (2018, May). Parameter adaptation for ant colony system algorithm using Hidden Markov Model for TSP problems. In Proceedings of the International Conference on Learning and Optimization

Algorithms: Theory and Applications,(pp.1-6). DOI:https://doi.org/10.1145/3230905.3230962

[33] Bouzbita, S., El Afia, A., Faizi, R., and Zbakh, M. (2016, May). Dynamic adaptation of the ACS-TSP local pheromone decay parameter based on the Hidden Markov Model. In 2016 2nd international conference on cloud computing technologies and applications (CloudTech) (pp. 344-349). IEEE. DOI: 10.1109/CloudTech.2016.7847719

[34] Bouzbita, S., El Afia, A., and Faizi, R. A New Hidden Markov Model Approach for Pheromone Level Exponent Adaptation in Ant Colony System. In Heuristics for Optimization and Learning (pp. 253-267). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-58930-1_17

[35] El Afia, Abdellatif, Mohamed Lalaoui, and Raddouane Chiheb. "A self controlled simulated annealing algorithm using hidden Markov model state classification." Procedia computer science 148 (2019): 512-521.

[36] Lalaoui, Mohamed, Abdellatif El Afia, and Raddouane Chiheb. "A self-adaptive very fast simulated annealing based on Hidden Markov model." 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech). IEEE, 2017.