INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 19, Issue: 6, Month: December, Year: 2024
Article Number: 6854, https://doi.org/10.15837/ijccc.2024.6.6854

CCC Publications

Integrating deep learning in target tracking applications, as enabler of control systems

V.O. Mihalca, O. Moldovan, I. Țarcă, D. Anton and D. Noje*

Vlad Ovidiu Mihalca
PhD School of Engineering Sciences
University of Oradea, Romania
Str. Universității nr. 1, Oradea, 410087, Bihor, Romania
ovidiu@vmihalca.ro

Ovidiu Moldovan
Department of Mechatronics
University of Oradea, Romania
Str. Universității nr. 1, Oradea, 410087, Bihor, Romania
omoldovan@uoradea.ro

Ianina Țarcă
Università Bocconi, Milano, Italy
Via Sarfatti, 25 Milano - PI 03628350153
ianina.tarca@studbocconi.it

Daniel Anton
PhD School of Engineering Sciences
University of Oradea, Romania
Str. Universității nr. 1, Oradea, 410087, Bihor, Romania
danielanton0227@gmail.com

Dan Noje*
RDI Programs and Projects Department
University of Oradea, Romania
Str. Universității nr. 1, Oradea, 410087, Bihor, Romania
*Corresponding author: dan.noje@gmail.com

Abstract

Target tracking is a key component of control systems with applications in various domains. Several examples of commercial applications may be given, which benefit from this technology: the development of car collision avoidance systems which must detect and track potential obstacles and hazards, or the development of UAVs that can track and record the evolution of athletes. For the purpose of our research, this technology was developed and tested as part of a less complex application. The current work describes a simple yet practical implementation of vision-based control for a mobile robot system. In the experiment, we used a mobile robot as target and a similar robot as follower. To achieve the tracking task, the used strategy involves the detection of a specific visual object mounted on the target, by extracting its features which are then used in issuing control commands within a remotely-closed loop. A Deep Learning approach is used for object detection, incorporating a detector model into the strategy while preserving an explicit controller in the overall scheme. The carried experiment has proven that this new approach provides the expected results, which make it a suitable tool for development of larger scale applications of control systems.

Keywords: vision-based control, visual servoing, object detector, transfer learning, deep learning

1 Introduction

Recent increase in computational capabilities of processing units brought with it a renewed, surging interest in Computer Vision and its applications - which can be observed in both the scientific community and industry. Machine Learning and Deep Learning have almost become synonymous with Artificial Intelligence due to the focus given to these sub-fields, as they have delivered good results in solving particular classes of problems such as object classification, object detection, texture analysis[1] or image segmentation. Many related fields benefit from the advances in Computer Vision, examples being Robotics, Security & Surveillance, Medicine, Automotive and others.

Vision-based control in Robotics refers to using information of a visual nature, such as images captured by a camera sensor, to drive a mobile robot or control an end-effector such as an industrial robotic arm. This is in contrast with using other simpler numeric information such as values read by one or more sensors (ultrasonic, infrared, wheel encoders). Visual information needs to be processed and interpreted relying on procedures and methods from Computer Vision, then used in conjuction with controller implementations to achieve vision-based control, employing additional processing algorithms as needed to avoid data loss issues[2, 3]. Such issues can also be avoided by relying upon data validation modules[4].

There are numerous application domains for vision-based control used with mobile robots. To name a few, it is used in power and infrastructure maintenance[5, 9] or laboratory equipment inspection[6], sports assistance or arbitration (athlete follower UAVs) as well as more complex applications such as automotive development of autonomous cars (collision avoidance subsystems and autonomous driving[7]). Mobile robots, particularly ground and air (UAVs), are recently deployed on a large scale in applications regarding automation of tasks in agriculture[12] as well as cleaning tasks[10]. Important mentions are also health applications[8] such as UV-C disinfection[11] which is of crucial importance, as highlighted by the recent SARS-CoV2 crisis.

Larger groups of mobile robots for search & rescue missions are described in [13], a paper which surveys swarm robotics principles and methods. A key aspect to highlight in interactions from mobile robot swarms is that of simpler, composable tasks, which can be simulated[14] in order to incrementally build more complex control strategies with confidence, in an iterative fashion. Not only for passively testing controller formulations over time, simulation software can also serve as a highly visual[15] user interface for interacting with a prototype model, for example in UAV pilot training[16]. A next step from interacting with a simulation model is an interactive interface with physical prototypes.

The work presented in this paper is part of a larger research project, having as goal the design and implementation of a visual control scheme based on a specific target object, using a remote software control application. Inspired by ideas and concepts presented by P. Corke[17], this project continues the study of vision-based control in the tradition of [18] with a novel system, control strategy and approach to visual information extraction, as well as simulations for a differential-drive kinematic model. It is comprised of a mobile robot group and a prototyping platform[21] receiving information from the robots and closing the control loop by sending remote commands. This system architecture draws inspiration from projects with educational value such as remote control lab experiments[19] or robot remote control AR applications[20].

The focus of this research paper is on both the vision-based control scheme as well as the Deep Learning model used for the target detection component of the control strategy.

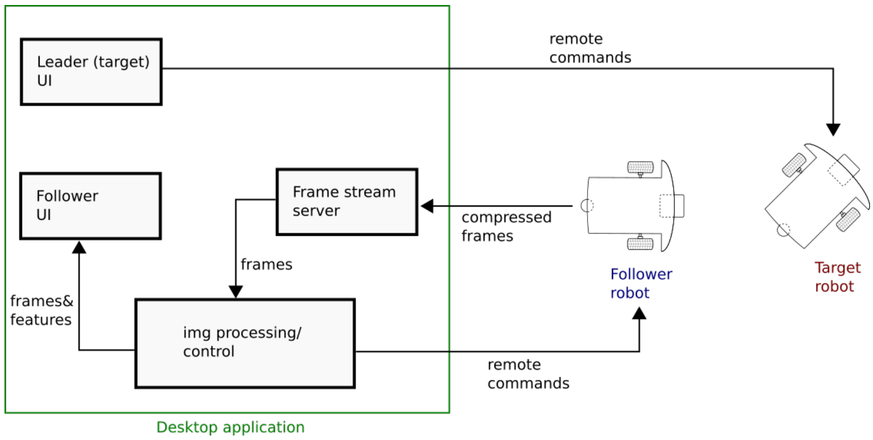


Figure 1: Block diagram of software control platform[21]

2 System architecture and control principle

2.1 Robot system and software control platform

Part of the solution developed to accomplish the objectives of this research comes in the form of a desktop application referred to as a *software control platform* or a *prototyping platform*, as it allows testing many different vision-based control schemes with any type of mobile robot prototypes. It is agnostic of specific robot details, as it receives input data from the robot sensors over the network communications channel and responds with remote commands which are serialized and sent to the robot, effectively closing the vision-based control loop specific to any type of Visual Servoing method. While not being the focus of this paper, its components and principle of operation is illustrated in Figure 1 for a better understanding of the method used. Together with a leader and follower group it constitutes the entire vision control system enabling the necessary information flow. Implementation details for the prototyping platform and mobile robots have been presented in [21].

The **problem statement** for the target tracking task can be defined in this manner: the follower robot has to maintain the target robot in the centre of its field of view, closing in on the distance as the target is accelerating away or decelerating (approaching). In order to propose a control strategy which achieves the task and use a simplified mathematical model, several assumptions have to be made regarding the mobile robot group.

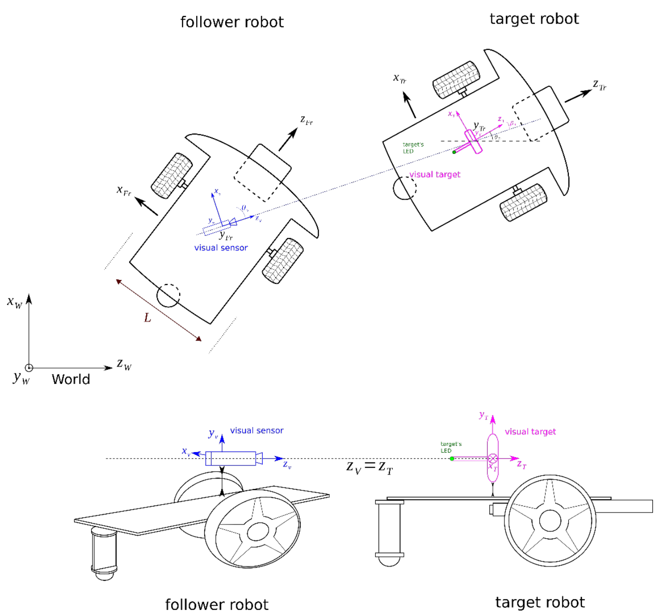


Figure 2: Multi-robot system configuration

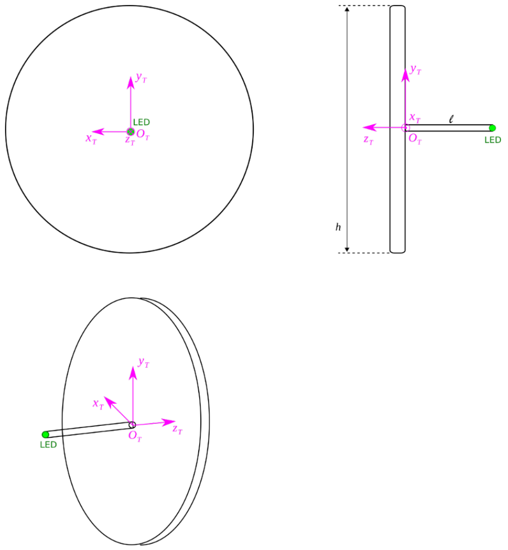


Figure 3: Visual disk tracked by the follower robot

As can be seen in Figure 2, both robots perform motion on the same surface plane.The camera sensor as well as the visual target object of interest are mounted on their respective mobile robot. The optical axis of the camera sensor is situated at the same height on the vertical z-axis as the middle of the visual target disk. Formally, if we attach the $O_V x_V y_V z_V$ coordinate system to the camera and the $O_T x_T y_T z_T$ coordinate system to the visual target disk, then at all times:

$$z_V = z_T \tag{1}$$

The special object mounted on the leader robot, known as the *visual target disk*, is a circle-shaped custom mechanical part (Figure 3) mounted in a central position on the chassis, approximately in the middle of the gauge between the wheels ($\frac{L}{2}$). The disk features a perpendicular cylinder ending with a green LED, in the opposite direction of the robot's front part. This shape serves to differentiate visual target orientation within the image when rotated to the left or right, in visual control schemes involving target pose estimation. Image-based methods such as the one presented in this paper do not make use of this feature. This visual disk is a 3D printed custom part in the physical mobile robot prototype.

2.2 Vision-based target tracking

Given the previous assumption regarding system geometry and visual target shape, a vision-based control scheme can further be defined. Image-based visual servoing principle mandates that features are to be extracted from each captured frame and their values compared with a predefined reference known as *goal* or *desired* feature value. These goal values can be empirically obtained or through some elaborate means, it makes little difference at this point. The error between desired feature values and actual feature values is computed and it must result in a control signal which generates the appropriate motion to reduce that error.

The proposed method is named ***M-H** feature strategy*, as it involves two image features to be extracted: the horizontal middle position (M) of the visual disk and its height (H). These two features are relevant to the target tracking task in the following way: when the target gains distance from the follower, its height diminishes. Similarly, when the target performs a turning motion followed by a lateral displacement with respect to the follower robot, the middle of the visual disk shifts towards the edge of the captured image frame. This can be easily proved by drawing projection lines and intersecting them with the virtual image plane. The camera sensor used in practice performs a perspective projection.

A controller which makes use of the error between a set-point and a measured value is the PID family of controllers, with all their variations. These amplify the error by a certain constant, while

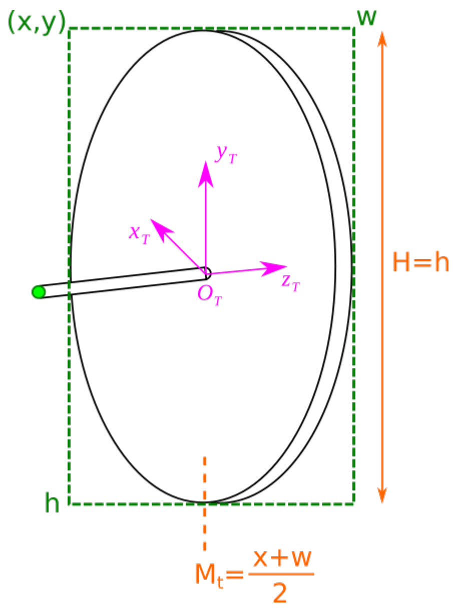


Figure 4: Visual features extracted from bounding rectangle

also computing the time derivative and integral of the errors (the D and I components). To realize the required motion, a variation of PD controller was implemented using the Python language. The output of this controller represents a value for the wheel speed control and is used directly by the motor bridge software component to accelerate or turn the follower.

The M and H features can be extracted through simple computations from the bounding rectangle of the visual target disk. The relation between these can be deduced from Figure 4. The middle of the target can be computed as in eq. 2 (where w is the bounding rectangle width), while the height of the bounding rectangle is used directly as the H feature ($H = h$).

$$M_t = \frac{x+w}{2} \tag{2}$$

Obtaining the bounding rectangle represents a typical target detection problem in Computer Vision. Within the research project that this work is part of, detection has been initially achieved using a colour-based image segmentation technique performed within the HSV colour space. Given that the follower camera is able to capture frames only in BGR, the colour space is converted by the software control platform into HSV for each frame received from the robot. The current paper presents an alternative approach to target detection, based on Deep Learning (which does not require additional computations for such a conversion).

3 Deep-Learning visual target detection method

3.1 Model

Object detection can be achieved in different ways using Deep Learning. Traditional object detectors use multiple passes to propose several candidate regions for the object to be detected, and then select the best candidate location. State-of-the-art detectors, such as YOLO or SSD, use a single pass to perform the detection task [22, 23]. Variations also exist on architectures different than SSD or YOLO, and the one proposed by Komorowski et al. [24] achieves results for a task similar to our circle-shaped object detection by relying on a pyramidal type architecture. Their *FootAndBall* model propagate the output of several base convolutional layers to the dense fully-connected layers which have the role of a regressor, determining the object location from the image features extracted by the convolutional layers.

This idea has been used in the model presented within this paper, as well. However, given that the detection task is single-class and more trivial than in other scenarios, instead of defining and training a complex model from the ground-up, our detector benefits from the Transfer Learning property that artificial neural networks possess. Its architecture is presented in Figure 5. The overall model proposed for detecting the visual target object uses a pre-trained model as a feature extractor base - in this case a ResNet. The output of the convolutional layers is forwarded to another sub-model which acts as the regressor for the bounding rectangle location. This regressor sub-model is composed of fully-connected layers.

The weights of the base extractor are frozen, as it has been pre-trained on a large dataset. Only the regressor is trained on the specific dataset necessary for the detection task attempted with the follower robot. This architecture and training method for a detector-type model have been described in several web resources on the topic[25, 26].

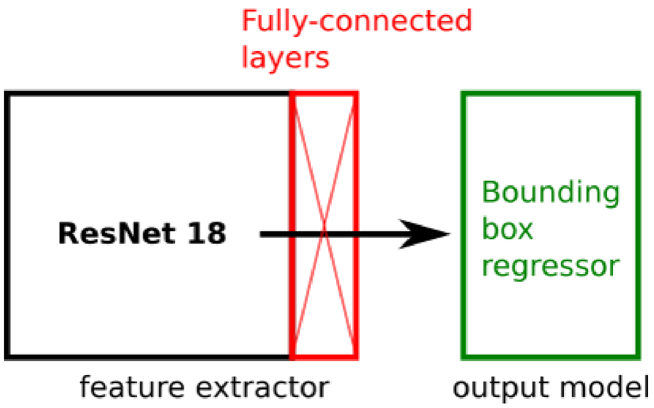


Figure 5: Deep Learning detector model architecture

As Chilamkurthy presented in [25], this model has been implemented using PyTorch framework. The source code definition is provided in Listing 1.

```
import torch.nn as nn
from torchvision.models import resnet18, ResNet18_Weights

class ObjectDetector(nn.Module):
    def __init__(self):
        super().__init__()

        conv = resnet18(weights=ResNet18_Weights.IMAGENET1K_V1)
        num_features = conv.fc.in_features
        conv.fc = nn.Identity()

        for param in conv.parameters():
            param.requires_grad = False

        self.extractor = conv
        self.regressor = nn.Linear(num_features, 4)

    def forward(self, x):
        x = self.extractor(x)
        return self.regressor(x)
```

Listing 1: PyTorch detector model definition

3.2 Dataset

Training the model for the particular detection task in the research project required a custom image dataset having the visual target object in image frames, correctly labelled with the coordinates

Hyper-parameter	Value
Learn rate	.0001
No. of epochs	20
Batch size	4

Table 1: Initial detector training hyper-parameters

of its bounding rectangle. In order to obtain this dataset, an UI component was added to the software control platform which captures the next frame streamed from the follower robot (named *Dataset Gatherer*). For automatically labelling the captured frames, the previously mentioned colour-based image segmentation procedure has been applied, saving the label in a separate CSV file containing the image filename and the coordinate values.

Given the weaknesses of the colour segmentation procedure, some frames are mis-labelled. In order to address this issue and create a suitable training or validation dataset, these errors required manual intervention. To enable a visual approach to editing, a specialized graphical tool was developed, named *Dataset Editor*. This iterated over the entire dataset, drawing the bounding box according to the coordinates saved for each frame. SpinBox inputs allowed for editing the coordinates and the whole label CSV file could be saved with the new values, therefore correcting any errors that were present.

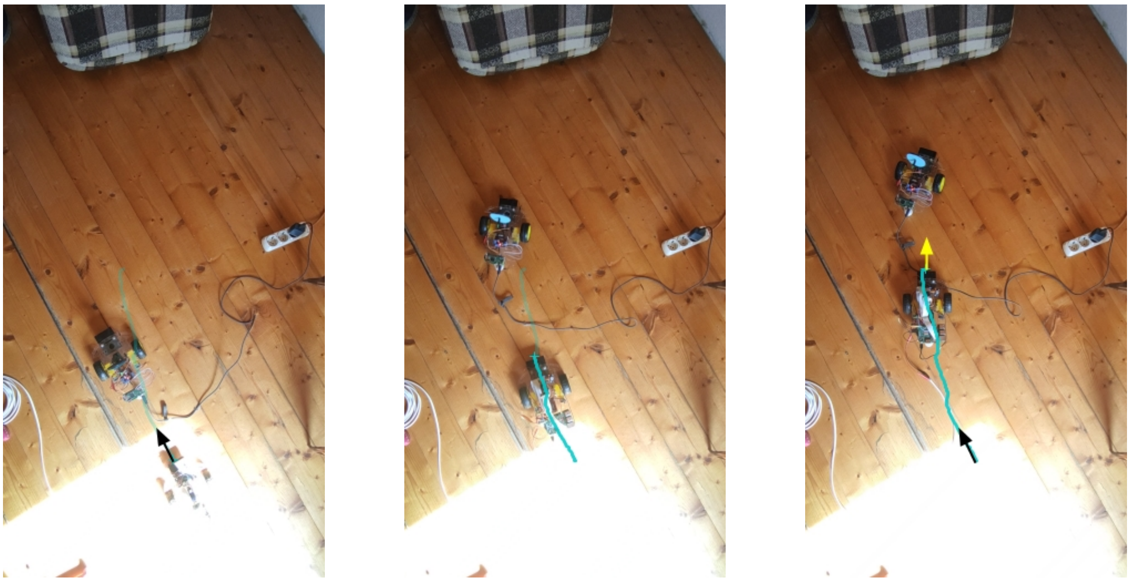
4 Experimental results

The previously discussed target tracking strategy has been experimentally tested using the software control platform and a pair of physical prototypes for the leader-follower mobile robot group. The two robots involved in the experiment have a differential-drive configuration, using a similar chassis and mechanical structure yet having specific particularities. Each of them feature DC geared motors attached to the side wheels and mounted in a symmetrical fashion, together with a ball pillar mounted in the back of the robot for static balance. Motors are driven using a L298N module controlled by PWM signals for speed and 3V3 digital logic for direction, in a *sign-magnitude* mode as presented by Doug Paradis[27]. Given that the main controller unit on the follower robot is a Raspberry Pi 3 single-board computer (SBC), which features a very limited number of hardware PWM outputs, another electronic component was necessary for generating sufficient control signals. This came in the form of a PCA9685 module, having an IC with the same name capable of generating up to 16 different PWM signals with the same frequency.

For the experimental setup the camera sensor mounted on the follower streamed images with a rate of 24 FPS, limited from the software running on the Raspberry Pi. Remote control commands were only issued when the value was different than the previous one, as a bandwidth-saving measure and for increased responsiveness in the control loop. The resolution used for each frame transmitted over the network was 320x240 pixels, values chosen for efficient streaming at the cost of a reduced field of view. Each frame was compressed in JPEG format, thereby creating a MJPEG-like video stream. The motor driver board was controlled with PWM signals operating at 100 Hz, which improved motor control accuracy at lower speeds as opposed to higher frequency values.

The results of a curved right-turning trajectory experiment can be seen in the sequence of images from Figure 6. Several annotations were added onto the captured video using Kinovea software, to facilitate visual comparison of leader and follower poses. Follower trajectory has also been highlighted. A powerful sunlight can be noticed in the bottom part of each image capture. The experiment was conducted during the afternoon to benefit from natural light. All the mobile robots are powered via cables and the camera sensor information gathered or the remote commands are sent over the wireless network.

The initial training parameters for the model are given in Table 1.



(a) (b) (c)

Figure 6: Follower trajectory frame sequence

5 Conclusions and future directions

The proposed control strategy has been shown to be a viable approach to simple generic tasks such as target tracking, which scale to more complex systems or can be part of a larger control scheme. Simple and practical implementations make it suitable for a wide variety of applications.

The software control platform presented in this paper allows a great flexibility regarding the types of mobile robots used or the vision-based control methods employed. There are numerous future directions of development for the system. Improvements to network communications, such as a single-channel compact protocol, may allow implementations based on wireless technologies such as Bluetooth which can be operated outdoor and are not as dependant on a fixed laboratory network. Mobile robot prototypes can be improved by using components with increased performance and a different control unit with a real-time nature. Last but not least, the desktop application consisting the remote control platform can be developed into a plugin-based architecture to allow an increased flexibility in the study of various control strategies, while also being rewritten in a more efficient, lower level programming language such as Rust or C++.

In what regards the control technique presented, improvements can be made to the Deep Learning model itself, by choosing a dedicated architecture which achieves the detection task with minimal computational power required. Robot controllers can be layered so that additional PID components adjust DC motor PWM signals to bring the wheel to the desired velocity setpoint.

Acknowledgment . *The research has been funded by the University of Oradea, within the Grants Competition "Scientific Research of Excellence Related to Priority Areas with Capitalization through Technology Transfer: INO - TRANSFER - UO", Project No. 326/21.12.2021.*

References

[1] Vesselenyi, Tiberiu and Dzitac, Ioan and Dzitac, Simona and Vaida, Victor (2008) Surface Roughness Image Analysis using Quasi-Fractal Characteristics and Fuzzy Clustering Methods, In *International Journal of Computers Communications & Control*, DOI: https://doi.org/10.15837/ijccc.2008.3.2398, 3(3), 304–316, 2008

[2] Noje, Dan and Dzitac, Ioan and Pop, Nicolae and Ţarcă, Radu Cătălin (2020) IoT Devices Signals Processing Based on Shepard Local Approximation Operators Defined in Riesz MV-Algebras, In *Informatica*, DOI: 10.15388/20-INFOR395, 31(1), 131–142, 2020

[3] Noje, Dan and Țarcă, Radu Cătălin and Dzitac, Ioan and Pop, Nicolae (2019), IoT Devices Signals Processing based on Multi-dimensional Shepard Local Approximation Operators in Riesz MV-algebras, In *International Journal of Computers Communications & Control*, 14(1), 56–62, 2019

[4] Noje, Dan and Țarcă, Radu Cătălin and Pop, Nicolae and Moldovan, Alin Octavian and Moldovan, Ovidiu Gheorghe (2022), Automatic System Based on Riesz MV-algebras, for Predictive Maintenance of Bearings of Industrial Equipment Using Temperature Sensors, In *ICCCC 2022: Intelligent Methods Systems and Applications in Computing, Communications and Control*, Springer, Cham, DOI: 10.1007/978-3-031-16684-6_1, 3–19, 2022

[5] Araar, Oualid and Aouf, Nabil (2014), Visual servoing of a Quadrotor UAV for autonomous power lines inspection, In *2014 22nd Mediterranean Conference on Control and Automation, MED 2014*, DOI: 10.1109/MED.2014.6961575, 1418–1424, 2014

[6] Khattar, Fawzi and Luthon, Franck and Larroque, Benoit and Dornaika, Fadi (2021), Visual localization and servoing for drone use in indoor remote laboratory environment, In *Machine Vision and Applications*, Springer Berlin Heidelberg, DOI: 10.1007/s00138-020-01161-7, 32(1), 1–13, 2021

[7] Kilyen, Nandor Alpar and Lemnariu, Rareș Florin and Muntean, Ionuț and Moiș, George Dan (2024) An Indoor Localization System for Automotive Driving Competitions, In *International Journal of Computers Communications & Control*, DOI: https://doi.org/10.15837/ijccc.2024.1.6030, 19(1), 2024

[8] Yıldırım, Şahin and Sertaç, Savaş (2022), Design of a Mobile Robot to Work in Hospitals and Trajectory Planning Using Proposed Neural Networks Predictors, In *International Conference on Reliable Systems Engineering (ICoRSE) - 2021*, Springer International Publishing, DOI: 10.1007/978-3-030-83368-8_4, 32–45, 2022

[9] Feng, Xin and Velinsky, Steven A. (1997), Distributed control of a multiple tethered mobile robot system for highway maintenance and construction, In *Computer-Aided Civil and Infrastructure Engineering*, DOI: 10.1111/0885-9507.00071, 12(6), 383–392, 1997

[10] Prassler, Erwin and Ritter, Arno and Schaeffer, Christoph and Fiorini, Paolo (2000), A short history of cleaning robots, In *Autonomous Robots*, DOI: 10.1023/A:1008974515925, 9(3), 211–226, 2000

[11] Anton, Daniel Melentie and Milaș, Raul Cristian and Țicărat, Teofil and Mihalca, Vlad Ovidiu and Țarcă, Radu Cătălin and Birouaș, Flaviu Ionuț (2023), Experimental Testing of UV-C Light Sources Used Disinfection Mobile Robots, In *2023 17th International Conference on Engineering of Modern Electric Systems (EMES)*, IEEE, DOI: 10.1109/EMES58375.2023.10171692, 1–4, 2023

[12] Yıldırım, Şahin and Ulu, Burak (2023), Deep Learning Based Apples Counting for Yield Forecast Using Proposed Flying Robotic System, In *Sensors*, DOI: 10.3390/s23136171, 23(13), 2023

[13] CORNEA, Mihai and MIHALCA, Vlad Ovidiu (2016), A REVIEW OF SWARMING UNMANNED AERIAL VEHICLES, In *ANNALS OF THE ORADEA UNIVERSITY. Fascicle of Management and Technological Engineering*, DOI: 10.15660/AUOFMTE.2016-3.3270, 25(3), 2016

[14] Mihalca, Vlad Ovidiu and Anton, Daniel Melentie and Țarcă, Radu Cătălin and Yıldırım, Şahin (2023), Simulating Simple Tasks for a Kinematic Model of Differential-drive Mobile Robots, In *2023 17th International Conference on Engineering of Modern Electric Systems (EMES)*, IEEE, DOI: 10.1109/EMES58375.2023.10171646, 2023

[15] Vancea, Codruța and Mihalca, Vlad Ovidiu and Toia-Creț, Titus Ruben (2011), Mathematical and Programmatic Design in Matlab: A Rewarding Experience, In *Journal of Computer Science and Control Systems*, 4(1), 203–207, 2011

[16] Khuwaja, Komal S.A. and Chowdhry, B. S. and Khuwaja, K. F. and Mihalca, V. O. and Ţarcă, R. C. (2018) Virtual Reality Based Visualization and Training of a Quadcopter by using RC Remote Control Transmitter, In *IOP Conference Series: Materials Science and Engineering*, DOI: 10.1088/1757-899X/444/5/052008, 444(5), 2018

[17] Corke, Peter (2017) *Robotics, Vision and Control - Fundamental algorithms in MATLAB (2nd ed.)*, Springer International Publishing AG, DOI: 10.1007/978-3-319-54413-7, 2017

[18] Ţarcă, Radu Cătălin (2001), *Conducerea roboţilor utilizând sisteme servovizuale*, Editura Universităţii din Oradea, ISBN: 973-613-038-X, 2001

[19] Ţarcă, Radu Cătălin (2010), Virtual and Remote Control Lab Experiment Using Matlab, In *ANNALS OF THE ORADEA UNIVERSITY. Fascicle of Management and Technological Engineering*, DOI: 10.15660/auofmte.2010-3.2017, XIX(3), 78–81, 2010

[20] Pasc, Ildiko and Csokmai, Lehel and Popenţiu-Vlădicescu, Florin and Ţarcă, Radu Cătălin (2014), Augmented reality used for robot remote control in educational laboratories, In *Applied Mechanics and Materials*, DOI: 10.4028/www.scientific.net/AMM.658.672, 658, 672–677, 2014

[21] Mihalca, Vlad Ovidiu; Anton, Daniel Melentie and Ţarcă, Radu Cătălin (2023). Prototyping platform design for visual robot control experiments, In *Annals of the University of Oradea: Fascicle of Management & Technological Engineering*, Ed. Universităţii din Oradea, DOI: 10.15660/AUOFMTE.2023-1.36821, 1, 1, 2023.

[22] Redmon, J. and Divvala, S. and Girshick, R. and Farhadi, A (2016) You Only Look Once: Unified, Real-Time Object Detection In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 779–788, 2016

[23] Liu, Wei and Anguelov, Dragomir and Erhan, Dumitru and Szegedy, Christian and Reed, Scott and Fu, Cheng-Yang and Berg, Alexander C (2016) SSD: Single Shot MultiBox Detector In *Computer Vision – ECCV 2016*, Springer International Publishing 21–37, 2016

[24] Komorowski, Jacek and Kurzejamski, Grzegorz and Sarwas, Grzegorz (2020) Footandball: Integrated player and ball detector In *VISIGRAPP 2020 - Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* DOI: 10.5220/0008916000470056, 5, 47–56, 2020.

[25] Chilamkurthy, Sasank, *Transfer Learning for Computer Vision Tutorial*, [Online]. Available: `https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html`, Accesed on 24 June 2024.

[26] Chakraborty, Devjyoti, *Training an object detector from scratch in PyTorch*, [Online]. Available: `https://pyimagesearch.com/2021/11/01/training-an-object-detector-from-scratch-in-pytorch/`, Accesed on 24 June 2024.

[27] Paradis, Doug, *L298N Motor Driver Board - Drive Modes*, [Online]. Available: `https://www.dprg.org/l298n-motor-driver-board-drive-modes/`, Accesed on 25 June 2024.

Cite this paper as:

V.O. Mihalca, O. Moldovan, I. Țarcă, D. Anton and D. Noje(2024). Integrating deep learning in target tracking applications, as enabler of control systems *International Journal of Computers Communications* & *Control*, 19(6), 6854, 2024.

https://doi.org/10.15837/ijccc.2024.6.6854