Ana I. Pereira · Armando Mendes ·
Florbela P. Fernandes · Maria F. Pacheco ·
João P. Coelho · José Lima
Editors

Optimization, Learning Algorithms and Applications

Third International Conference, OL2A 2023
Ponta Delgada, Portugal, September 27–29, 2023
Revised Selected Papers, Part II

Springer

Editors
Ana I. Pereira
Instituto Politécnico de Bragança
Bragança, Portugal

Armando Mendes
University of Azores
Ponta Delgada, Portugal

Florbela P. Fernandes
Instituto Politécnico de Bragança
Bragança, Portugal

Maria F. Pacheco
Instituto Politécnico de Bragança
Bragança, Portugal

João P. Coelho
Instituto Politécnico de Bragança
Bragança, Portugal

José Lima
Instituto Politécnico de Bragança
Bragança, Portugal

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-3-031-53035-7 ISBN 978-3-031-53036-4 (eBook)
https://doi.org/10.1007/978-3-031-53036-4

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2024
Chapters 13 and 23 are licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/). For further details see license information in the chapters.

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

Preface

The volumes CCIS 1981 and 1982 contains the refereed proceedings of the III International Conference on Optimization, Learning Algorithms and Applications (OL2A 2023), a hybrid event held on September 27–29.

OL2A provided a space for the research community in optimization and learning to get together and share the latest developments, trends and techniques as well as develop new paths and collaborations. OL2A had the participation of more than four hundred participants in an online and face-to-face environment throughout three days, discussing topics associated with areas such as optimization and learning and state-of-the-art applications related to multi-objective optimization, optimization for machine learning, robotics, health informatics, data analysis, optimization and learning under uncertainty and 4th industrial revolution.

Six special sessions were organized under the topics Learning Algorithms in Engineering Education, Optimization in the SDG context, Optimization in Control Systems Design, Computer Vision Based on Learning Algorithms, Machine Learning and AI in Robotics and Machine Learning and Data Analysis in Internet of Things. The event had 66 accepted papers. All papers were carefully reviewed and selected from 172 submissions. All the reviews were carefully carried out by a scientific committee of 115 PhD researchers from 23 countries.

The OL2A 2023 volume editors,

September 2023

Ana I. Pereira
Armando Mendes
Florbela P. Fernandes
Maria F. Pacheco
João P. Coelho
José Lima

Organization

General Chairs

Ana I. Pereira	Polytechnic Institute of Bragança, Portugal
Armando Mendes	University of the Azores, Portugal

Program Committee Chairs

Florbela P. Fernandes	Polytechnic Institute of Bragança, Portugal
M. Fátima Pacheco	Polytechnic Institute of Bragança, Portugal
João P. Coelho	Polytechnic Institute of Bragança, Portugal
José Lima	Polytechnic Institute of Bragança, Portugal

Special Session Chairs

João P. Teixeira	Polytechnic Institute of Bragança, Portugal
José Cascalho	University of the Azores, Portugal

Technology Chairs

Paulo Medeiros	University of the Azores, Portugal
Rui Pedro Lopes	Polytechnic Institute of Bragança, Portugal

Program Committee

Ana Isabel Pereira	Polytechnic Institute of Bragança, Portugal
Abeer Alsadoon	Charles Sturt University, Australia
Ala' Khalifeh	German Jordanian University, Jordan
Alberto Nakano	Federal University of Technology – Paraná, Brazil
Alexandre Douplik	Ryerson University, Canada
Ana Maria A. C. Rocha	University of Minho, Portugal
Ana Paula Teixeira	University of Trás-os-Montes and Alto Douro, Portugal
André Pinz Borges	Federal University of Technology – Paraná, Brazil

André Rodrigues da Cruz	Federal Center for Technological Education of Minas Gerais, Brazil
Andrej Košir	University of Ljubljana, Slovenia
António José Sánchez-Salmerón	Universitat Politècnica de València, Spain
António Valente	University of Trás-os-Montes and Alto Douro, Portugal
Armando Mendes	University of the Azores, Portugal
Arnaldo Cândido Júnior	Federal Technological University – Paraná, Brazil
B. Rajesh Kanna	Vellore Institute of Technology, India
Bilal Ahmad	University of Warwick, UK
Bruno Bispo	Federal University of Santa Catarina, Brazil
C. Sweetlin Hemalatha	Vellore Institute of Technology, India
Carlos Henrique Alves	CEFET - Rio de Janeiro, Brazil
Carmen Galé	University of Zaragoza, Spain
Carolina Gil Marcelino	Federal University of Rio de Janeiro, Brazil
Christopher Expósito Izquierdo	University of Laguna, Spain
Clara Vaz	Polytechnic Institute of Bragança, Portugal
Damir Vrančić	Jožef Stefan Institute, Slovenia
Dhiah Abou-Tair	German Jordanian University, Jordan
Diamantino Silva Freitas	University of Porto, Portugal
Diego Brandão	CEFET - Rio de Janeiro, Brazil
Dimitris Glotsos	University of West Attica, Greece
Eduardo Vinicius Kuhn	Federal Technological University – Paraná, Brazil
Elaine Mosconi	Université de Sherbrooke, Canada
Eligius M. T. Hendrix	Malaga University, Spain
Elizabeth Fialho Wanner	Federal Center for Technological Education of Minas Gerais, Brazil
Felipe Nascimento Martins	Hanze University of Applied Sciences, The Netherlands
Florbela P. Fernandes	Polytechnic Institute of Bragança, Portugal
Florentino Fernández Riverola	University of Vigo, Spain
Francisco Sedano	University of León, Spain
Fredrik Danielsson	University West, Sweden
Gaukhar Muratova	Dulaty University, Kazakhstan
Gediminas Daukšys	Kauno Technikos Kolegija, Lithuania
Gianluigi Ferrari	University of Parma, Italy
Glaucia Maria Bressan	Federal University of Technology – Paraná, Brazil
Glotsos Dimitris	University of West Attica, Greece
Humberto Rocha	University of Coimbra, Portugal
João Paulo Carmo	University of São Paulo, Brazil
João Paulo Coelho	Polytechnic Institute of Bragança, Portugal
João Paulo Teixeira	Polytechnic Institute of Bragança, Portugal

Jorge Igual	Universitat Politécnica de Valencia, Spain
Jorge Ribeiro	Polytechnic Institute of Viana do Castelo, Portugal
José Boaventura-Cunha	University of Trás-os-Montes and Alto Douro, Portugal
José Cascalho	University of the Azores, Portugal
José Lima	Polytechnic Institute of Bragança, Portugal
José Ramos	Nova University Lisbon, Portugal
Joseane Pontes	Federal University of Technology – Ponta Grossa, Brazil
Josip Musić	University of Split, Croatia
Juan A. Méndez Pérez	University of Laguna, Spain
Juan Alberto García Esteban	University de Salamanca, Spain
Júlio Cesar Nievola	Pontifícia Universidade Católica do Paraná, Brazil
Kristina Sutiene	Kaunas University of Technology, Lithuania
Laura Belli	University of Parma, Italy
Lidia Sánchez	University of León, Spain
Lino Costa	University of Minho, Portugal
Luca Davoli	University of Parma, Italy
Luca Oneto	University of Genoa, Italy
Luca Spalazzi	Marche Polytechnical University, Italy
Luis Antonio De Santa-Eulalia	Université de Sherbrooke, Canada
Luís Coelho	Polytechnic Institute of Porto, Portugal
M. Fátima Pacheco	Polytechnic Institute of Bragança, Portugal
Mahmood Reza Khabbazi	University West, Sweden
Manuel Castejón Limas	University of León, Spain
Marc Jungers	Université de Lorraine, France
Marco Aurélio Wehrmeister	Federal University of Technology – Paraná, Brazil
Marek Nowakowski	Military Institute of Armoured and Automotive Technology in Sulejowek, Poland
Maria do Rosário de Pinho	University of Porto, Portugal
Martin Hering-Bertram	Hochschule Bremen, Germany
Matthias Funk	University of the Azores, Portugal
Mattias Bennulf	University West, Sweden
Michał Podpora	Opole University of Technology, Poland
Miguel Ángel Prada	University of León, Spain
Mikulas Huba	Slovak University of Technology in Bratislava, Slovakia
Milena Pinto	Federal Center of Technological Education Celso Suckow da Fonseca, Brazil
Miroslav Kulich	Czech Technical University Prague, Czech Republic
Nicolae Cleju	Technical University of Iasi, Romania

Paulo Alves	Polytechnic Institute of Bragança, Portugal
Paulo Leitão	Polytechnic Institute of Bragança, Portugal
Paulo Lopes dos Santos	University of Porto, Portugal
Paulo Medeiros	University of the Azores, Portugal
Paulo Moura Oliveira	University of Trás-os-Montes and Alto Douro, Portugal
Pavel Pakshin	Nizhny Novgorod State Tech University, Russia
Pedro Luiz de Paula Filho	Federal Technological University – Paraná, Brazil
Pedro Miguel Rodrigues	Catholic University of Portugal, Portugal
Pedro Morais	Polytechnic Institute of Cávado e Ave, Portugal
Pedro Pinto	Polytechnic Institute of Viana do Castelo, Portugal
Roberto Molina de Souza	Federal University of Technology – Paraná, Brazil
Rui Pedro Lopes	Polytechnic Institute of Bragança, Portugal
Sabrina Šuman	Polytechnic of Rijeka, Croatia
Sancho Salcedo Sanz	Alcalá University, Spain
Sandro Dias	Federal Center for Technological Education of Minas Gerais, Brazil
Sani Rutz da Silva	Federal Technological University – Paraná, Brazil
Santiago Torres Álvarez	University of Laguna, Spain
Sara Paiva	Polytechnic Institute of Viana do Castelo, Portugal
Shridhar Devamane	Global Academy of Technology, India
Sławomir Stępień	Poznań University of Technology, Poland
Sofia Rodrigues	Polytechnic Institute of Viana do Castelo, Portugal
Sudha Ramasamy	University West, Sweden
Teresa Paula Perdicoulis	University of Trás-os-Montes and Alto Douro, Portugal
Toma Rancevic	University of Split, Croatia
Uta Bohnebeck	Hochschule Bremen, Germany
Virginia Castillo	University of León, Spain
Vítor Duarte dos Santos	Nova University Lisbon, Portugal
Vitor Pinto	University of Porto, Portugal
Vivian Cremer Kalempa	State University of Santa Catarina, Brazil
Wojciech Giernacki	Poznań University of Technology, Poland
Wojciech Paszke	University of Zielona Gora, Poland
Wynand Alkema	Hanze University of Applied Sciences, The Netherlands
Zahia Guessoum	University of Reims Champagne-Ardenne, France

Contents – Part II

Computer Vision Based on Learning Algorithms

Machine Learning and AI in Robotics

Deep Learning-Based Localization Approach for Autonomous Robots in the RobotAtFactory 4.0 Competition

Luan C. Klein[1,2](✉), João Mendes[2,3], João Braun[2,3,4,5], Felipe N. Martins[6], Andre Schneider de Oliveira[1], Paulo Costa[4,5], Heinrich Wörtche[6,7], and José Lima[2,3,5]

[1] Universidade Tecnológica Federal do Paraná, Campus Curitiba, UTFPR/PR, Curitiba, Brazil
`luanklein@alunos.utfpr.edu.br, andreoliveira@utfpr.edu.br`
[2] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Bragança, Portugal
`{joao.cmendes,jbneto,jllima}@ipb.pt`
[3] Laboratory for Sustainability and Technology in Mountain Regions (SusTEC), Instituto Politécnico de Bragança, Bragança, Portugal
[4] Faculty of Engineering of the University of Porto, Porto, Portugal
`paco@fe.up.pt`
[5] INESC Technology and Science, Porto, Portugal
[6] Sensors and Smart Systems Group, Institute of Engineering, Hanze University of Applied Sciences, Groningen, The Netherlands
`{fe.nascimento.martins,h.j.wortche}@pl.hanze.nl`
[7] Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract. Accurate localization in autonomous robots enables effective decision-making within their operating environment. Various methods have been developed to address this challenge, encompassing traditional techniques, fiducial marker utilization, and machine learning approaches. This work proposes a deep-learning solution employing Convolutional Neural Networks (CNN) to tackle the localization problem, specifically in the context of the RobotAtFactory 4.0 competition. The proposed approach leverages transfer learning from the pre-trained VGG16 model to capitalize on its existing knowledge. To validate the effectiveness of the approach, a simulated scenario was employed. The experimental results demonstrated an error within the millimeter scale and rapid response times in milliseconds. Notably, the presented approach offers several advantages, including a consistent model size regardless of the number of training images utilized and the elimination of the need to know the absolute positions of the fiducial markers.

Keywords: Indoor Localization · CNN · Robotic Competition

1 Introduction

A necessary skill for agents in numerous circumstances, notably in robotics with Autonomous Mobile Robots (AMR), is the ability to localize themselves in an

A. I. Pereira et al. (Eds.): OL2A 2023, CCIS 1982, pp. 181–194, 2024.
https://doi.org/10.1007/978-3-031-53036-4_13

environment. Solving the localization problem means determining the pose—a combination of position and orientation—concerning some reference frames, often the global reference frame. Several strategies have been created employing sensor data and different techniques. Artificial intelligence approaches are also increasingly being used to solve this issue.

Many countries have created robotic competitions to encourage research and inspire students. In this study, the Portuguese Competition RobotAtFactory 4.0 was chosen to carry out the methods and validate the concepts proposed here. In this competition, an AMR must move boxes through a model warehouse in the shortest possible time.

One of the current approaches used in this competition to solve the robot localization issue is presented in [1], in which previous knowledge of the location of fiducial markers (ArUcos markers) in the environment is used. The robot's pose is estimated using the relative pose of the robot's camera concerning the ArUcos, through geometry. A limitation of this approach is that the pose of all ArUcos in the field must be precisely known. This is acceptable in a known scenario (like the competition) but can be challenging in situations in which the position of the markers is not precisely known, such as in potentially hazardous circumstances. To overcome this issue, we investigated some Machine Learning (ML) approaches in previous works [2] and [3]. However, the dependency on the tag's identification remains since the approaches presented are based on the relative ArUco's pose.

Within this context, the present work aims to explore and validate the application of Convolutional Neural Networks (CNN) to the robot localization problem. In the proposed approach, there is no need to know the pose of the ArUco markers, neither in the global reference frame nor relative to the robot's camera. As in our previous works, we use the RobotAtFactory 4.0 competition as a scenario for the reported experiments and results.

The rest of the text is divided into four sections: Sect. 2 presents related works; Sect. 3 presents theoretical concepts and explains how the work was developed, namely the data collecting, pre-processing of data and the models used; Sect. 4 presents the results and the discussions; Finally, Sect. 5 presents the conclusion and future works.

2 Related Work

The study of localization is crucial in various contexts, especially in the AMR. Given its importance, research in this area has generated considerable interest. There are two categories of localization: Outdoor and Indoor. The widespread Global Positioning System (GPS) is a frequent method for outside settings, but due to physical restrictions, it is ineffective indoors [4]. So, multiple alternatives have been explored and developed. Base future decisions on wrong pose estimations might have terrible consequences [5]. Therefore recognizing the associated uncertainty in pose estimation is crucial.

Many strategies have been developed, and one of the most well-known is the Kalman Filter, first introduced in [6]. It is a mathematical method that uses (noisy) measurements over time to produce estimates that approximate the real values. The Kalman filter was designed for linear systems, but a similar approach can be used in nonlinear situations, such as the Extended Kalman Filter (EKF) [7]. Another interesting approach is Markov Localization, a probabilistic algorithm that maintains a probability distribution throughout all hypotheses instead of just one [8]. An alternative method is the Monte Carlo Localization, which uses numerous samples (particles) with different weights, which represent hypotheses of the interest variable [9] [10].

More techniques for localization include map-matching algorithms such as Perfect Match [11], Iterative Closest Point (ICP) [12], and Normal Distributions Transform (NDT) [13]. A comparative analysis of these approaches was performed in [14]. Further approaches were developed using landmarks in the environment, such as the use of fiducial markers [15]. Using them in a SLAM (Simultaneous Localization and Mapping) problem with other localization techniques was also explored [16].

Recent years have seen a substantial increase in the use of machine learning techniques due to improvements in computing capacity. Various methods for feature extraction, selection, and regression in the localization context were examined in a survey in 2020 [17]. This survey highlights that this field is still in its early stages, and several problems require further exploration.

Additionally, by employing pictures from the robot's camera and other sensors, Convolutional Neural Networks (CNN) have been used to support robot localization [18] [19]. An interesting study on this topic is in [20], where the authors employed images for localization in a 6-DoF (Degrees of Freedom) scenario. They applied transfer learning from GoogLeNet [21] and achieved promising outcomes in outdoor and indoor environments.

Focusing on the RobotAtFactory 4.0 competition, one of the current methods utilized to address the localization issue is considering the stored data of the ArUcos pose and its relative position to the camera reference frame [1]. Based on these two pieces of information, analytical geometry estimates the robot's pose concerning the global reference frame. Stochastic filters like the EKF and Mahalanobis filters are also used to aggregate the guesses and improve their accuracy.

The analytical approach's disadvantage is that it necessitates accurate information about each ArUco's pose, which might be difficult to get in hostile conditions. In [2], a machine learning (ML) solution is proposed to overcome this restriction, where prior knowledge about the ArUcos' pose is not required. However, the demand for the ArUcos to be recognized in the images remains.

3 Background and Methodology

This section is divided into three parts: Sect. 3.1 is focused on the theoretical explanation of the CNNs, exploring the transfer-learning concept and the VGG16 model. Section 3.2 aims to explain the competition scenario, explaining the robot, the scenario, and the simulator. Finally, Sect. 3.3 describes how the work was done, the data collection, the model training, and the metrics considered to evaluate the models.

3.1 Theoretical Background

Artificial Neural Networks (ANN) were developed to replicate how the human brain works. This method was founded on the premise that brain cells, or neurons, and the connections between them, or synapses, are the root of mental activity. The earliest neuron model was created in 1943 [22] and involves multiple inputs, each with an associated weight. These inputs are multiplied by the corresponding weights before being added along with a bias term. The neuron output is obtained by passing the resultant sum through an activation function. An ANN is, in essence, a collection of neurons arranged in layers. Feeding data into a neural network in which each neuron's output serves as an input for the layer above it is known as feedforward. Backpropagation is the process of determining optimal weights and biases. It uses the learning rate as a critical variable to control how rapidly the network may modify its weights and biases [23].

Convolutional Neural Networks (CNN) are a particular type of ANN frequently employed in deep learning for processing data with a known grid-like architecture, such as images [24]. The use of the convolution operation in at least one layer, as described in [25], is the distinguishing feature of CNN. In summary, the two basic components of CNN are feature extraction and classification/regression. The first one consists of filters, using convolutional layers and layer-pooling techniques that reduce the size of the representation, speed up computation, and improve feature resilience. Fully connected layers handle the classification or regression operation.

Most AI approaches have been trained to perform a specific task. Coming later to the conclusion that the learning acquired for a specific function can be reused, a new field called transfer learning emerged. Several pre-trained models are accessible, particularly in CNN[1]. To carry out this work, the pre-trained network VGG16[2] was chosen [26], due to its ease of interpretation and possible modification of hyperparameters. This, despite its age and not guaranteeing the best results today, continues to be a good choice considering several parameters, such as the simplicity of its structure, the reduced number of parameters if used as a transfer-learning model, and the low computational capacity required by it.

The VGG16 model's architecture was initially proposed with a fixed input ConvNet (224×224 RGB image), and only necessitating normalization of the image pixel values during pre-processing [26]. Compared to other presented models, VGG16 demonstrated a deeper structure with 16 weight layers (13 convolutional layers and three fully-connected layers) using 3×3 filtering to extract

---

[1] Some examples are available at https://keras.io/api/applications/.

[2] The ranking of the approaches considering the ImageNet dataset are available at https://paperswithcode.com/sota/image-classification-on-imagenet.

features from the input image effectively. To reduce information volume, a 2 × 2 max-pooling layer was applied at the end of the filter stack. In the ImageNet-trained[3] model version, the final max-pooling layer is connected to a fully connected layer containing 4096 neurons. This layer's output is then fed into a softmax layer for 1000 classifications. A graphical representation of the VGG16 architecture is presented in Fig. 1.

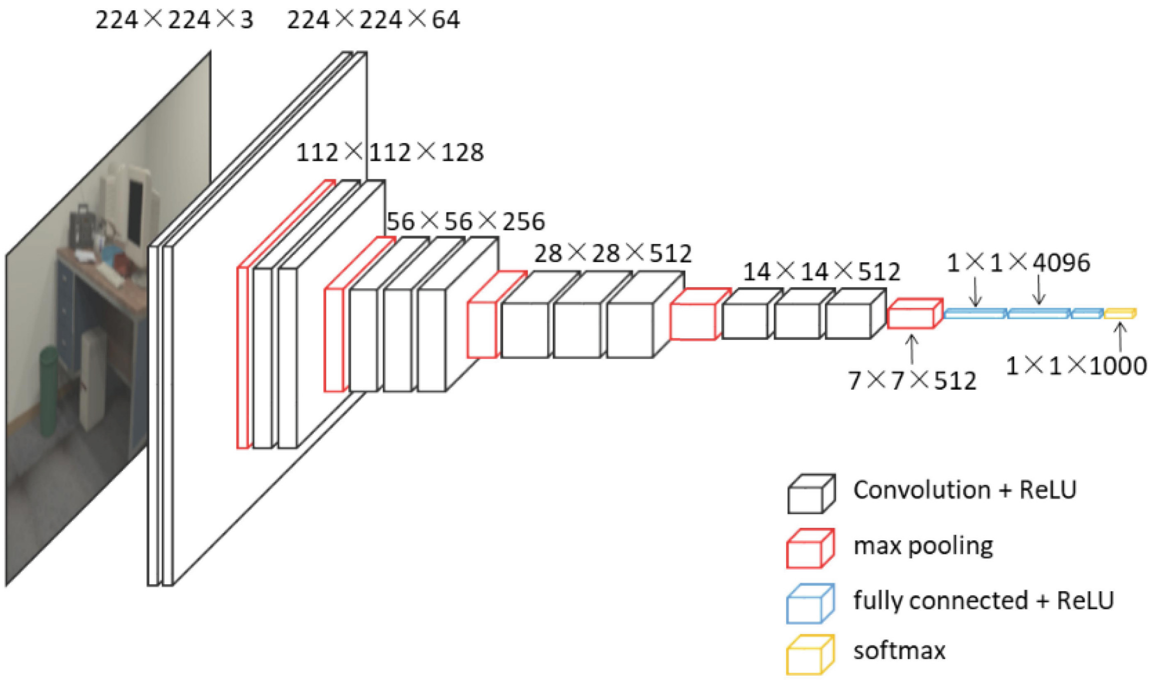

Fig. 1. VGG16 architecture. Adapted from [27].

3.2 Scenario Context

This study investigates if the localization problem in the RobotAtFactory 4.0 Competition[4] can be resolved using CNN. In this competition, an AMR must move as many boxes as possible in the shortest time around the warehouse. This challenge aims to replicate a condition in automated factories, such as warehouses, where several procedures to meet the requirements are necessary. Since the warehouse is part of an automated system, robots may replace human workers, increasing efficiency and safety.

The robot must comply with some rules to participate in this competition, including fitting in a cube of 30 × 30 × 30 cm and being entirely autonomous-any connection with an external system the organization does not provide is forbidden.

---

[3] Details in https://www.image-net.org/.

[4] More details at https://www.festivalnacionalrobotica.pt/2023/robotfactory-4-0/.

The principal parts of the robot are depicted in Fig. 2: While the Arduino Uno handles the low-level control of the robot, such as motors and encoders, the Raspberry Pi handles the high-level control of the robot, managing, for instance, the RGB camera, localization, navigation, and decision-making [28].

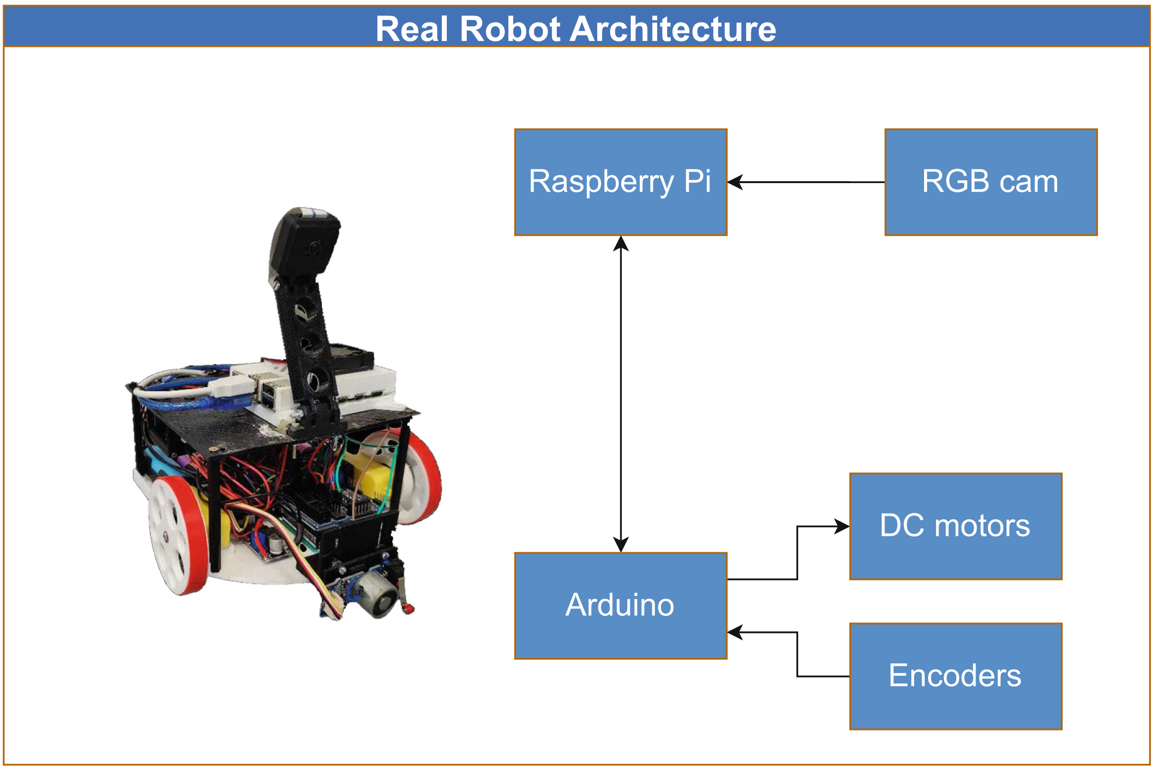

Fig. 2. A representation of robot architecture. Two boards compose the system: the Raspberry Pi, which manages decision-making and regulates the RGB camera, and the Arduino, which is in charge of interfacing with low-level hardware such as the motors and encoders. Source: [2].

Additionally, the competition organization created a simulation setting of the competition[5]. This simulator works with rigid-body dynamics interactions and constraints [29]. The user can interact with a variety of features on the simulator. For instance, an editor for XML files allows the user to modify definitions for the robot and the surroundings. A code editor in the Pascal language is another feature that makes it possible to develop the robot's programming, for example, by creating an algorithm to define the robot's route [28]. More details about the simulator can be found in [29]. The simulator depicted in Fig. 3 presents a virtual version of the RaF competition field. Additionally, Fig. 4 displays an image taken by the robot's camera in the simulator.

---

[5] Available at: https://github.com/P33a/SimTwo.

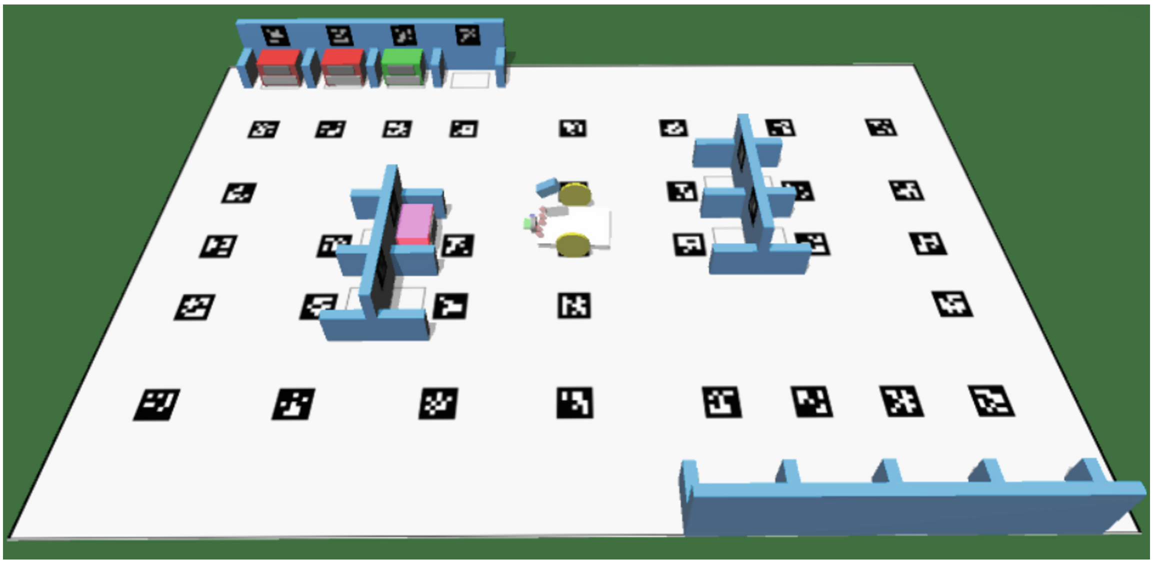

Fig. 3. Simulation scene that displays the robot and the competition's simulated environment.

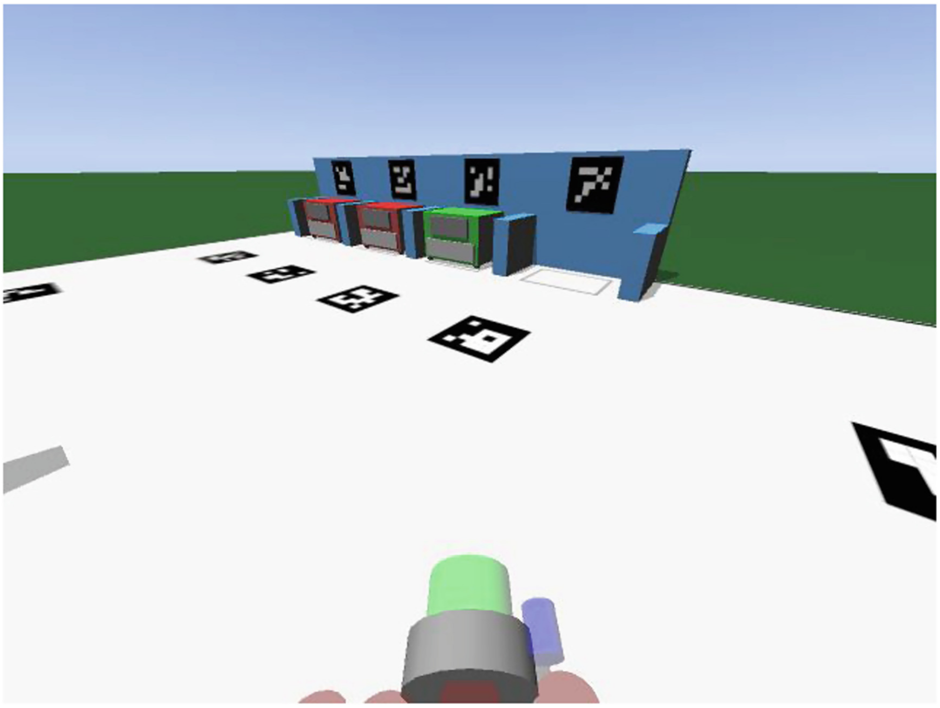

Fig. 4. Example of an image taken by the robot's camera in simulation.

3.3 Methodology

Data Collection. Only data from a simulated scenario was used in this study to facilitate implementing and evaluating the proposed solutions. The field was discretized into a grid to gather the data, and the robot was placed in each open spot - that is, a spot without obstacles - to collect data. The robot rotates

360° while taking about 60 photos in each location to build a database for ML training.

Five data collections were done, considering different aspects. The **Dataset A** have images from the whole environment and considering a grid resolution equals 1 cm, i.e., each square in the grid has a side with a size equal to 1 cm. Only a small portion of the field—a square measuring 10 × 10 cm in the field's center—was considered while creating the other datasets. The grid's resolutions considered were 10 mm, 5 mm, 2.5 mm, and 1 mm, creating, in this way, the datasets **B1**, **B2**, **B3** and **B4**, respectively. In addition, each image collected is associated with a robot pose, composed of three values: $\{x, y, \theta\}$.

Data Preprocessing. The initial data preprocessing necessary is the removal of "ambiguous images", i.e., images that can not be distinguished uniquely. An image is considered "ambiguous" when no ArUco[6] is visible in the frame. This restriction was done to avoid images that do not improve the training and the validation of the models. Furthermore, this limitation will be treated in future works, for example, by using filters and odometry. After this, the preprocessing in each image is done using the pre-defined function specific to the VGG16 model[7].

Aiming to improve the quality of the estimations, two different CNNs were built, with the same structure: one model to estimate the robot position (x and y); and the other to estimate its orientation (θ). Each model is based on transfer-learning from the VGG16 model, as explained in Sect. 3.1. The model's adaptation was based on removing the last layers (dense and softmax) and adding three dropouts (with factor 0.2), three dense layers, 2 with 4096 neurons, and 1 with 1072. Finally, the output layer has a linear activation function and 1 or 2 outputs (with one output for the model to estimate θ and with two outputs to estimate x and y). It is essential to highlight that only the new layers were trained, and all the original weights of VGG16 were maintained, i.e., no fine-tuning was done. Figure 5 presents the adapted model.

To evaluate the quality of the models, four metrics were used: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Normalized Root Mean Squared Error (NRMSE), and R^2 [30]. The following equations give these metrics:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{1}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{2}$$

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}}, \tag{3}$$

---

[6] To identify the ArUcos, the OpenCV library was used: https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html.

[7] More detail at https://www.tensorflow.org/api_docs/python/tf/keras/applications/vgg16/preprocess_input.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y}_i)^2}, \tag{4}$$

where y_i represents the true value and the $\hat{y}_i$, represents the predicted value, for instance, i, y_{max} and y_{min} represents the max and min value, respectively, of the observations, while $\overline{y}$ represents the mean. The best value possible for MAE and RMSE is 0, and the worst value is $+\infty$, while for R^2, the range is $(-\infty, 1]$, where $-\infty$ is the worst possible value, and one is the best.

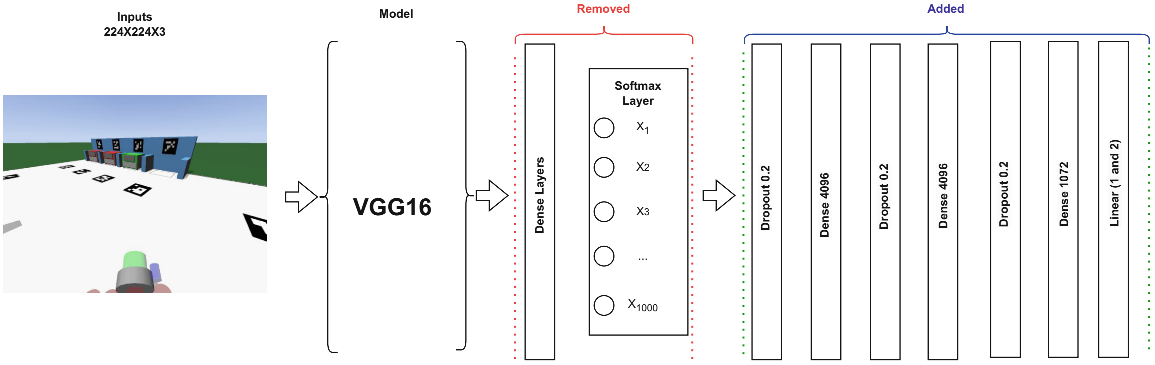

Fig. 5. Architecture of the proposed model.

The datasets were divided into three parts: 80% to training, 10% to validation, and 10% to testing. Since the goal of this study is to propose a possible solution and not a final localization system, the traditional split of the datasets was done aiming to keep it the most as simple as possible. In this way, these data were applied to the CNN models. The cross-validation technique was applied, aiming to avoid overfitting. This way, the data was divided again (independent of the previous division), and a new model was trained and tested with the latest data. This process was done three times, and the final results were based on the average of metrics calculated for the three executions.

Furthermore, as part of the training process, the x and y values were scaled by a factor of 1000 due to their small initial magnitude to facilitate significant improvements. Subsequently, after the estimation phase, the values were rescaled by dividing them by 1000 to restore their original scale.

In addition to the scaling operation, various settings were configured for the Convolutional Neural Network (CNN) model. Specifically, the training was executed over 200 epochs, utilizing an adaptive learning rate strategy[8] with an initial value of 0.0005. The MAE metric was employed for monitoring purposes, with a reduction factor of 0.8, the patience of three epochs, and a minimum

---

[8] https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html.

value constraint set at 0.00000001. Furthermore, an early stopping criterion was defined, terminating the training process after ten epochs.

All the tests were performed using a GPU NVIDIA A100, with 16GB. The operating system used was Ubuntu 22.04.2 LTS (Jammy Jellyfish), the Python version used was 3.10.6, and the libraries used were: Pandas 2.0.0 and Tensorflow 2.12.0. The CUDA version used was 11.8, with CUDNN 8.9.1.

4 Results and Discussions

The first result to be shown is considering the entire field, using the grid's resolution equal a 1 cm. These results are presented in Table 1, where each column shows a metric for each one of the three components of the pose. Each value is the average of the three executions, and the respective standard deviation is presented.

Table 1. Results obtained considering the whole field.

	MAE			RMSE			NRMSE			R2		
	x[m]	y[m]	θ[°]	x[m]	y[m]	θ[°]	x	y	θ	x	y	θ
Avg.	0.0100	0.0070	6.05	0.0211	0.0142	11.79	0.015	0.019	0.033	0.997	0.996	0.985
Std. Dev.	0.0029	0.0017	2.72	0.0091	0.0028	5.05	0.007	0.004	0.014	0.002	0.002	0.012

The average training time required to train the model for the position (x, y) was 10,047.15 s, with a standard deviation of 3,412.12 s. On the other hand, the model for the orientation (θ) required 5078.67 s with a standard deviation of 1334.86 s. Regarding the inference, the necessary time to estimate the pose of a preprocessed image was 1.9 ms to x and y, and 1.9 ms to θ, with standard deviations of 0.7 ms and 0.5 ms, respectively. In this way, the total time to estimate the complete pose was around 3.8 ms, not considering the necessary time to preprocess the image.

The second result is considering the limited part of the field but varying the grid's resolution. Table 2 presents the results, where the columns represent the resolutions and the lines the metrics for each pose component. All the values (except for the training time) are averages of the three executions.

Comparing the result in the whole field, presented in Table 1 with the results presented in [2], it is possible to see a similarity between the random forest (RF) and CNN. To emphasize, the RF was the approach that presented the best results in that previous study, with errors equal 7 mm in x, 6 mm in y, and 3.05° in θ. Comparing with the results in the current study, a millimeter difference was obtained in x and y (3 mm and 1 mm, respectively), while the difference in θ was 3.0°. In addition, the RMSE and NRMSE presented by the CNNs were similar to that presented by the RF. Furthermore, all the results presented a small standard deviation for the three executions (considering the cross-validation process), indicating that the results of the models were satisfactory and trusty.

Table 2. Results obtained considering the limited part of the field, using different grid's resolution, with *Avg.* columns indicating the average and *Std. Dev.* indicating the standard deviation.

Grid's Resolution		10 mm		5 mm		2.5 mm		1 mm	
Quantity of images		8306		33,291		113,596		655,130	
		Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.
MAE	x[m]	0.0026	0.0001	0.0026	0.0004	0.0023	0.0002	0.0023	0.0006
	y[m]	0.0026	0.0000	0.0027	0.0004	0.0023	0.0004	0.0023	0.0007
	θ[°]	2.97	0.29	2.76	0.14	1.58	0.10	4.51	3.82
RMSE	x[m]	0.0034	0.0002	0.0033	0.0005	0.0029	0.0002	0.0029	0.0006
	y[m]	0.0038	0.0006	0.0042	0.0002	0.0032	0.0003	0.0030	0.0007
	θ	6.03	1.50	7.09	0.76	4.06	0.71	6.99	3.80
NRMSE	x	0.0307	0.0015	0.0302	0.0043	0.0293	0.0026	0.0262	0.0059
	y	0.0345	0.0053	0.0368	0.0021	0.0292	0.0031	0.0279	0.0062
	θ[°]	0.02	0.00	0.02	0.00	0.01	0.00	0.02	0.01
R2	x	0.990	0.001	0.990	0.003	0.990	0.002	0.989	0.005
	y	0.985	0.005	0.984	0.002	0.990	0.002	0.990	0.004
	θ	0.996	0.002	0.995	0.001	0.998	0.001	0.994	0.006
Training time (Position) [s]		678.63	80.85	898.40	153.75	4505.48	647.51	8281.08	233.89
Training time (Orientation) [s]		592.89	80.17	1437.00	401.92	17369.99	3399.39	18509.41	8368.63
Inference time (Position) [ms]		2.95	0.74	2.02	0.07	2.44	0.35	2.06	0.15
Inference time (Orientation) [ms]		2.46	0.12	2.04	0.05	2.22	0.19	1.96	0.40

Another exciting result of this work is the constant size of the models regardless of the number of images used. Each model has a size equal to 512 MB, totaling, in this way, 1024 MB for the complete pose estimation. This size is fixed due to the structure of the CNN, where only the values of parameters are updating, and not the architecture of the CNN. This is interesting because some ML approaches, such as Random Forest, can increase the size considerably since the structure of the model can vary, i.e., the trees can be deeper, according to the problem [3].

Analyzing the results presented in Table 2, it is possible to notice the errors were practically constant for all resolutions. This behavior differs from that found for the same test done in [2], where the prediction quality improved with the grid's resolution. So, it indicates that for the proposed model, the trade-off found in [2] is not applied here. Nevertheless, for the resolutions 10 mm, 5 mm, and 2.5 mm, the results using CNN were better or equal to that presented by random forest, and only for 1 mm the CNN results were worst than RF. Again, all the presented results show a small standard deviation, indicating that the results for all three executions in the cross-validation were close to each other.

5 Conclusions

The results confirm the main hypothesis and show that using CNN to address the localization issue in RobotAtFactory 4.0 is an effective strategy. Therefore, the main goal of this study was achieved. Comparing the results presented with the ones in [2], a significant improvement is seen, showing better results when the whole field is considered. When different grid resolutions are considered (10 mm, 5 mm, and 2.5 mm), the performance of the proposed CNN is similar to the ML techniques shown in [2]. Furthermore, the uniformity of the predictions, i.e., through the RMSE, evidence that the CNN presents more stable results than the other ML approaches.

In addition, another advantage to the ML approaches, especially Random Forest, was the stable size of the models. While the Random Forest can drastically vary the size of the trained models, such as presented in [3], the CNN models were constantly the same size, due to a pre-defined structure, before the training. This is an exciting advantage because it is important to understand the necessary resources if these models are executed in an embedded system.

Another exciting advantage to the ML approaches presented in [2] is that CNN does not necessarily the identification of ArUco's markers. While the input of the ML models is the relative pose of each ArUco, in the CNN, this identification is irrelevant since the input is just the image. However, it is essential to emphasize that the presented approaches are not the final system to pose estimation but a part of that. A complete localization system, which can include filters such as the Extended Kalman Filter, can solve problems with ambiguous images and improve the quality of the estimations.

Finally, in future work, we aim to embed the complete localization system and implement it in a real environment, making necessary adjustments, such as retraining the models with images from the real scenario. In addition, it is planned to explore various conditions and scenarios and assess their respective impacts on localization accuracy. Another interesting work is exploring other transfer-learning models and performing fine-tuning in the models, aiming to improve their quality.

Acknowledgments. The project that gave rise to these results received the support of a fellowship from "la Caixa" Foundation (ID 100010434). The fellowship code is LCF/BQ/DI20/11780028. João Braun is a PhD Student at the Faculty of Engineering, University of Porto (FEUP).

References

1. Braun, J., et al.: A robot localization proposal for the RobotAtFactory 4.0: a novel robotics competition within the Industry 4.0 concept. Front. Rob. AI **9** (2022). https://doi.org/10.3389/frobt.2022.1023590
2. Klein, L. C., et al.: A machine learning approach to robot localization using fiducial markers in RobotAtFactory 4.0 competition. Sensors **23**(6), 3128 (2023). https://doi.org/10.3390/s23063128
3. Klein, L.C., et al.: Using machine learning approaches to localization in an embedded system on RobotAtFactory 4.0 competition: a case study. In: 2023 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 69–74. IEEE (2023)
4. Grewal, M.S., Weill, L.R., Andrews, A.P.: Global Positioning Systems, Inertial Navigation, and Integration. John Wiley & Sons, Hoboken (2007)

5. Huang, S., Dissanayake, G.: Robot localization: an introduction. In: Wiley Encyclopedia of Electrical and Electronics Engineering, pp. 1–10 (1999)
6. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Basic Eng. **82**(1), 35–45 (1960). https://doi.org/10.1115/1.3662552
7. Welch, G., Bishop, G.A.: An introduction to the kalman filter. New York EBooks **1**(4), 1–16 (1995). https://academic.csuohio.edu/simond/reduce/ijar.pdf
8. Fox, D., Burgard, W., Thrun, S.: Markov localization for mobile robots in dynamic environments. J. Artif. Intell. Res. **11**, 391–427 (1999)
9. Gordon, N.J., Salmond, D.J., Smith, A.F.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In IEE Proceedings F (Radar and Signal Processing), vol. 140, no. 2, pp. 107–113. IET Digital Library (1993)
10. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Signal Process. **50**(2), 174–188 (2002)
11. Lauer, M., Lange, S., Riedmiller, M.: Calculating the perfect match: an efficient and accurate approach for robot self-localization. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 142–153. Springer, Heidelberg (2006). https://doi.org/10.1007/11780519_13
12. Besl, P.J., McKay, N.D.: Method for registration of 3-D shapes. In: Sensor Fusion IV: Control Paradigms and Data Structures, vol. 1611, pp. 586–606. Spie (1992)
13. Biber, P., Straßer, W.: The normal distributions transform: A new approach to laser scan matching. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453), vol. 3, pp. 2743–2748. IEEE (2003)
14. Sobreira, H., et al.: Map-matching algorithms for robot self-localization: a comparison between perfect match, iterative closest point and normal distributions transform. J. Intell. Rob. Syst. **93**, 533–546 (2019)
15. Michail, K., Cain, B., Carroll, S., Anand, A., Camden, W., Nikolaos, V.: Fiducial markers for pose estimation. J. Intell. Rob. Syst. **101**(4) (2021)
16. de Oliveira Júnior, A., Piardi, L., Bertogna, E. G., & Leitão, P.: Improving the mobile robots indoor localization system by combining slam with fiducial markers. In 2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE), pp. 234–239. IEEE (2021)
17. Nessa, A., Adhikari, B., Hussain, F., Fernando, X.N.: A survey of machine learning for indoor positioning. IEEE Access **8**, 214945–214965 (2020)
18. Sadeghi Esfahlani, S., Sanaei, A., Ghorabian, M., Shirvani, H.: The deep convolutional neural network role in the autonomous navigation of mobile robots (SROBO). Remote Sens. **14**(14), 3324 (2022)
19. Atanasyan, A., Roßmann, J.: Improving self-localization using CNN-based monocular landmark detection and distance estimation in virtual testbeds. In: Tagungsband des 4. Kongresses Montage Handhabung Industrieroboter, pp. 249–258. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-662-59317-2_25
20. Kendall, A., Grimes, M., Cipolla, R.: Posenet: a convolutional network for real-time 6-dof camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2938–2946 (2015)
21. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
22. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. **5**, 115–133 (1943)

23. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
24. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT press, Cambridge (2016)
25. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
27. Chen, Y., Chen, R., Liu, M., Xiao, A., Wu, D., Zhao, S.: Indoor visual positioning aided by CNN-based image retrieval: training-free, 3D modeling-free. Sensors **18**(8), 2692 (2018)
28. Braun, J., Júnior, A.O., Berger, G.S., Lima, J., Pereira, A.I., Costa, P.: RobotAtFactory 4.0: a ROS framework for the SimTwo simulator. In: 2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 205–210. IEEE (2022)
29. Costa, P., Gonçalves, J., Lima, J., Malheiros, P.: Simtwo realistic simulator: a tool for the development and validation of robot software. Theory Appl. Math. Comput. Sci. **1**(1), 17–33 (2011)
30. Sammut, C., Webb, G.I. (eds.): Encyclopedia of Machine Learning. Springer, Heidelberg (2011). https://doi.org/10.1007/978-0-387-30164-8