

# Development and Evaluation of a Computer Vision System for Robot Navigation and Object Recognition in Real-World Environments

Malene Helgo

Federal University of Maranhão, Vila Bacanga, Sao Luís - MA, 65080-805, Brazil.  
malenehelgo91@aol.com

Correspondence should be addressed Malene Helgo : malenehelgo91@aol.com

## Article Info

Journal of Computing and Natural Science (<http://anapub.co.ke/journals/jcns/jcns.html>)

Doi: <https://doi.org/10.53759/181X/JCNS202404007>

Received 16 May 2023; Revised from 30 July 2023; Accepted 06 October 2023.

Available online 05 April 2024.

©2024 The Authors. Published by AnaPub Publications.

This is an open access article under the CC BY-NC-ND license. (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

**Abstract** – The article discusses the vision framework for computing that includes image recognition, classification, prioritization, and navigation control modules. In this framework, a user model is used to feed the robotic controllers, whose performance improves in dynamic virtual contexts. In contrast, the vision module uses a multi-level perceptual neural network capable of efficient image segmentation, object recognition, and color segmentation, using the control module Position-Based Vision Servoing (PBVS) and actions such as Avoid Collision (),Go-Ahead (), and Follow(). It controls the motion of the robot, so the system successfully tested and met the requirements of the Antimedia Robotics Pioneer I robot. In addition, it was consistent with real life. The results show the effectiveness of the system in providing effective guidance and avoiding obstacles. Furthermore, the study investigates the use of artificial neural networks for image recognition and classification. In addition, it requires the use of SpCoMapping to add language maps to useful information. In summary, studies have emphasized the potential of computer vision and neural networks to improve robotic communication and language learning.

**Keywords** – Navigation Control Module, Computer Vision System, Position-Based Visual Servoing, MLP Neural Networks, Feature Extraction Module

## I. INTRODUCTION

Robotics is a multidisciplinary domain that encompasses the creation, assembly, functioning, and use of robots [1]. Robotics encompasses several disciplines that specialize in certain facets of the technology. Within the field of mechanical engineering, the word robotics pertains to the creation of the physical components of robots. In contrast, within computer science, robotics mostly involves the examination of robotic software. Additionally, there are several other facets of robotic development and related areas that intersect with it, such as electrical, control, software, information, electronic, telecommunications, computer, mechatronic, materials, and biomedical engineering. The objective of robotics is to develop devices capable of aiding and supporting humans.

The discipline of robotics focuses on the creation of robots capable of automating activities and performing a wide range of professions that may surpass human capabilities. Robots are employed in various scenarios and serve diverse functions. Presently, they are particularly utilized in hazardous settings such as inspecting radioactive substances, detecting, and neutralizing bombs, as well as in manufacturing processes [2]. Additionally, robots are deployed in environments where human presence is not feasible, such as outer space, underwater, high-temperature conditions, and the cleanup and containment of dangerous materials and radiation. Robots have the ability to adopt many physical forms, with some specifically designed to closely mimic human beings. This is said to facilitate the acceptability of robots in certain imitative actions typically carried out by humans. The robot's intent to mimic activities of human like thinking, lifting, speaking, walking, and other duties. A significant number of contemporary robots draw inspiration from natural phenomena, therefore making valuable contributions to the domain of bio-inspired robotics.

Some robots need operator input for operation, whilst others run independently. The idea of developing independent robots may be traced back to ancient times, but, important advancements in perception of their competence and prospective imposition did not occur until the 20<sup>th</sup> century [3]. Historically, many researchers, inventors, engineers, and technicians have often believed that robots would eventually possess the ability to imitate human behavior and perform activities in a manner similar to humans. The area of robotics is seeing fast growth nowadays due to ongoing technology advancements. Researching, creating, and constructing new robots serves a range of practical uses, including home, commercial, and military applications. Several robots are specifically designed to do tasks that pose significant risks to

human beings, such as disarming explosive devices, locating individuals trapped in unsafe structures, and investigating underground mines and sunken vessels. Robotics is used in STEM education as a pedagogical tool.

Computer vision is a multidisciplinary domain that focuses on enabling computers to acquire advanced comprehension from digital pictures or movies. From an engineering standpoint, its objective is to mechanize activities that can be performed by the human visual system. Computer vision focuses on the automated extraction, analysis, and comprehension of valuable data from either a single picture or a series of images [4]. We aim to develop a framework for devices to capture what they see using an automatic visual recognition system. Moving into computer vision, the field explores how artifacts interpret image-based information. Events vary; it can be video or images from multiple lenses and scanners. This task involves preprocessing, cutting images into blocks, and importing them into a module that handles and directs them. Studies show that when robots must work in environments full of change and motion, an efficient approach works best for their control systems.

The visual component is based on multi-layer perceptron (MLP) neural networks; these handle tasks such as sorting colors, sculpting blocks from images, and dotting objects. The course also delves into design fine-tuning, which is best suited to describing neural networks so they can learn effectively. The results highlight the precision and effectiveness of the visual system and its capacity for practical use in real-life scenarios. In addition, the research examines the use of artificial neural network models in picture processing and their potential to enhance language learning attributes. In summary, the research presents a justification for creating and using a computer vision system to enable robots to navigate and avoid obstacles.

The rest of the article has been organized as follows: Section II presents a discussion of the related works on computer vision system for robot navigation and object recognition in real-world environments. Section III focusses on computer vision system, discussing elements such as the robotic hardware, Saphira environment, interface, vision module, and the navigation control module. Section IV presents a critical analysis of the results, which focus on the system tuning, and related experimentations. Lastly, Section V presents a conclusion of the system used for robotic navigation and object recognition.

## II. RELATED WORKS

Human-robot interaction is a contemporary field of study that explores the use of robots with visual systems in production, opening up new possibilities [5]. By prioritizing the gesture modality and image processing capacity, an industrial robot may immediately interpret visual input and initiate an action. The majority of industrial robotic systems are focused on the duties and objectives particular to a certain context. The primary objective of the robotic program is to investigate the interface to a tele laboratory, which is specifically designed and developed to provide the user with several capabilities for the robot. These capabilities include cameras, control framework, and different representation methods such as augmented and virtual reality. When the application is introduced into an industrial setting, the interaction between humans and robots may become a difficulty. However, this issue has been more easily managed in recent times due to the advancements in current vision systems.

In the manufacturing environment, robots with vision systems can easily avoid problems. New innovations have resulted in bots with vision that can independently find clear paths in industrial areas. Independence requires such machines to travel without human assistance. Depending on the program, vision technologies fall into two categories: event-based or object-specific. They each outfit the robots with cameras and sensors and capture footage critical to their projects. 3D imaging technology enhances the ability to identify controllable features while identifying potential obstacles in humans. Object handling strategies—think object selection functions—depend exactly on the locations and visibility of objects in this environment. The additive 3D vision for a robot's energy is high in offices. This improvement comes from advanced hardware combined with sophisticated (voice-activated) software.

Using 3D thinking, robots develop honed skills that allow them to perform precise automated tasks (passive voice). Thus, accuracy in these areas is greatly improved by the continued improvement of such systems. The OPC architecture is a safe and reliable communication protocol designed for manufacturers, capable of connecting the smallest sensor all the way up to the corporate IT level and cloud. OPC Vision is a specialized image processing system built specifically for use on factory floors and general industrial platforms. The proposed aim is to include any element of image processing components into industrial automation applications in order to develop machine vision technologies capable of interacting with the whole plant. The user-level system of processing image offers a linguistic representation of visual data.

As presented by Quigley [6], manufacturing investigation duty systems of image processing interact with programmable logic controllers (PLCs). This approach transmits a pass or fail outcome to the Programmable Logic Controller (PLC) after picture analysis. The OPC Vision implements standardized communication methods. An ERP system has the ability to accurately identify the attributes of a frame grabber or retrieve image processing system streams via events for clients. An industrial automation application requires data and code to allocate tasks in the cloud. The cloud serves as a comprehensive middleware solution utilized in the Robotic Operating System (ROS) to facilitate communication and allow robots. The Cloud-based Robotic framework is specifically designed for high-bandwidth robotic applications in the industry, enabling the outsourcing of vision-enabled tasks.

This study presents a computer vision system consisting of primary parts. This article explores computer vision as a branch of artificial intelligence (AI) that empowers systems and computers to extract significant insights from videos,

modern photos, and other visual inputs. These insights may then be used to initiate activities or provide suggestions. AI empowers computers with the ability to reason, whereas computer vision equips them with the capacity to see, scrutinize, and comprehend. According to Cyert and DeGroot [7], computer vision operates similarly to human vision, but with humans having a prior advantage. The human visual system benefits from extensive exposure to contextual information over the course of a lifetime, enabling it to effectively discern objects, determine their distance, detect motion, and identify abnormalities within a picture. Computer vision enables machines to complete these jobs using algorithms, data, and cameras instead of relying on visual brain, optic nerves, and retinas. Moreover, it accomplishes these tasks in a much shorter amount of time. Due to its ability to assess a large number of processes or items per minute, a system trained to monitor asset manufacturing may detect invisible faults or problems that transcend human capabilities.

### III. COMPUTER VISION SYSTEM

The system of computer vision described in this study consists of two primary elements. The first component is accountable for the recognition, segmentation, and preprocessing of the picture. The second module implements the control of navigation, which is responsible for directing the robot throughout the surroundings. An RS232 connection facilitates the transmission and reception of data between the laptop and the robot [8]. The interface of USB facilitated the establishment of a link between the camera and the laptop. The suggested system design may be categorized as adhering to the automatic model, indicating that the robot lacks previous knowledge of the environment and does not retain the obtained information.

The reactive paradigm is very appealing for implementing robots' controllers operating in dynamic and real environments. This may be deduced from the reality that the automatic acts, under this model, are clearly defined relationships that dictate how the robot should respond based on the data gathered from its sensors. Furthermore, the behavior established in this paradigm may serve as a foundation for developing controllers that adhere to the hybrid paradigm. In this scenario, the robot would be capable of doing more intricate tasks, such as exploring the surroundings by generating a map and navigating around barriers.

#### *The Robotic Hardware*

The trials for the proposed vision system were conducted and evaluated only on an ActivMedia Robotics Pioneer I Robot. The robot's initial perception is formed using a set of seven Polaroid sonars 6500. External communication may be conducted via radio or the RS232 interface. In order to achieve visual awareness of the surroundings, the suggested graphic system used a Creative WebCam Go Plus camera. The camera has been positioned atop the robot and oriented downwards towards the floor. This design allows the robot to detect graphic data across a range of up to 4.8 meters in front of its base.

#### *Saphira Environment*

The interaction between the Pioneer I robot and advanced system is facilitated by the Saphira [9]. The Saphira is an ecosystem comprised of many collections used for the development of robotics applications. The Artificial Intelligence Center at Stanford Research Institute is responsible for its maintenance. The library contains functions that enable users to develop applications in the C/C++ language for controlling moving robots such as the Pioneer I. It provides an abstract interface to the hardware of the robot.

#### *The Interface*

The interface is a streamlined component that facilitates the communication between the system and the user. The parameters required for experiment control include color selection, maximum velocity, behavior weights, translation and rotation angles for picture segmentation (the specified color to be tracked). The interface also includes connection routines that are responsible for establishing and terminating the association with the simulator or the Pioneer I robot. The interface allows the user to configure the test's settings and see all the processing data via images and data created throughout the experiment.

#### *Vision Module*

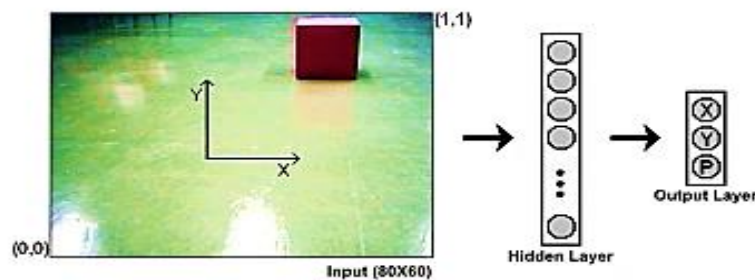
A module may have several meanings, including: In the context of computer software, a module refers to a distinct and self-contained unit of code that may be developed and managed separately for use in other systems. For instance, a developer may design a module that encompasses the necessary code for using a sound card or conducting input/output operations on a certain kind of filesystem. The module may thereafter be disseminated and used by any system requiring that specific feature, and the development of the module can progress autonomously. This methodology is often referred to as a modular design. In the context of computer hardware, a module refers to a self-contained constituent of a larger and more intricate system. For example, the memory module may interface with a computer motherboard to function as an integral part of the system. The module of vision is responsible for data processing and acquisition, which is then transmitted to the control module. The vision module acts as an interpreter that takes an image from the environment and produces a high-quality rendering of the image as a result.

For instance, there is a crimson item located at location XY. The control module may create the required instructions for the robot's navigation established the output of the automatic module. The vision module is comprised of three distinct tasks: recognition, image segmentation, and preprocessing. The first task involves the capture of a picture from the camera, which has a tenacity of 320X240 pixels, and then reducing it to a resolution of 80X60 pixels. The picture decrease is achieved by the computation of the average of the data points. A 4X4 pixel window is applied to the acquired picture, resulting in a point that represents the mean value of the 16 pixels inside the window. The resolution of this level, set at 80X60 pixels, has been determined by empirical analysis using genuine environmental pictures. The system has implemented the minimal resolution required for object recognition. The picture segmentation procedure is executed using a color categorization system.

The system of classification consists of a collection of MLP neural networks see **Fig. 1**, where each network is responsible for categorizing a certain color, distinguishing it from other colors in the picture. Inquiring about a crimson color, we label every speck on an image—crimson or not. This task unfolds as we examine RGB figures for each dot [10]. Feed these RGB snippets to an MLP neural network; the outcome reveals whether crimson reigns or another tint prevails. Employing what's termed segmentation approach, this technique isolates one pigment alone. Thus, in our divided display, only two colors emerge: white signifies our color of choice; black encompasses all that remains. Effectively identifying split segments within the visual part stands as our primary objective. Aiming for precision here, we consult a specialized MLP neural network uniquely suited for the task at hand. Within the segmented snapshot (measuring 80X60 dots), these serve as input signals [11]. Subsequently, our network renders a verdict: presence—or absence—of an object? Affirmation from this mechanism not only confirms its existence but also determines its precise location onscreen. Here dwells our distinguished MLP neural network—with trio neurons crowning its output tier.

The 1<sup>st</sup> and 2<sup>nd</sup> outputs reflect the X and Y positions, of the item in the picture. The third output (P) indicates whether or not the equipment is current in the **Fig 1**. The X and Y outputs were distinct based on empirical observations from trials conducted in a real setting, where the pictures were labeled according to the following values. X varies from 0.0 to 1.0 with an increment of 0.1, whereas Y ranges from 0.0 to 1.0 with the same increment. These values indicate the measurements corresponding to the object's location in the visual sector, not depending on the actual surroundings.

Nevertheless, during the conducted testing, it was seen that the Y output values of 0.0, 0.4, 0.7, 0.8, 0.85, 0.88, 0.9, and 0.91 could be associated with intervals of 0.5m in the actual habitat. When Y produces a value of 0.0, it indicates that the item is in close proximity to the robot. A Y output of 0.4 signifies that the image is about 0.5 meters away from the robot. This pattern continues with further Y outputs, each representing an additional distance of 0.5 meters. This estimate method for the robot's location in the actual environment has been designed based on the assumption that obtaining the actual 3D position of an item is not feasible without using stereo vision.



**Fig 1.** The Recognition MLP Neural Network.

*Navigation Control Module*

A control module was created to enable the robot to travel around the habitat by keeping off barriers and use Position-based visual servoing (PBVS) for guidance. The PBVS system consists of three fundamental modules: the module of Feature Extraction, the module of Pose Estimation, which utilizes methods of processing image, and the Control module where conventional or intelligent controllers are implemented, as seen in **Fig 2**. The posture of the end-factor robot is determined by using the graded vision sensorrepresentation of the object. This estimation is achieved by the matrix of transformation  $T_O^E$ . The retrieved characters  $f$ , derived from the reconstruction, are used to evaluate the pose  $P_a$ . This pose is a role of both the location of the orientation  $\theta$  and the end effector  $x(t)$ . The alignment is determined by the use of inverse kinematics that compute the location of the target item  $x(t)$  relative to the frame of a camera. The joint restrainer is specifically intended to ensure that the error  $e$ , which represents the difference between the actual pose  $P_a$  and the reference posture  $P_d$ , is reduced to zero.

Ultimately, PBVS control computes the angular velocity ( $\omega c$ ) and linear velocity ( $vc$ ). The inverse Jacobian matrix is computed using these numbers to determine the velocities that are joint required for the robot's movement. A benefit of PBVS is that it allows for the formulation of desired tasks in cartesian space, making it easy to calculate the error. In their study, Vijayan and Ashok [12] conducted a comparative investigation of PBVS and IBVS for industrial assembly,

focusing on the aspects of accuracy and speed. The PBVS method was applied to the ABB robot and it was discovered that it had a quicker execution time compared to the IBVS method. Additionally, the PBVS method was evaluated using numerous cameras. Sharma et al. used PBVS (Pose-Based Visual Servoing) to determine the transformation between the camera and robot base of a mobile robot. They employed the gradient descent approach for this estimation. Cheng, Li, Jiao, and An [13] used PBVS (posture-Based Visual Servoing) to estimate the posture of a robot and navigate around barriers. The authors used the aversion torque approach to determine the manipulators that are repulsive inverse kinematics, yielding results that were characterized by speed, flexibility, and accuracy.

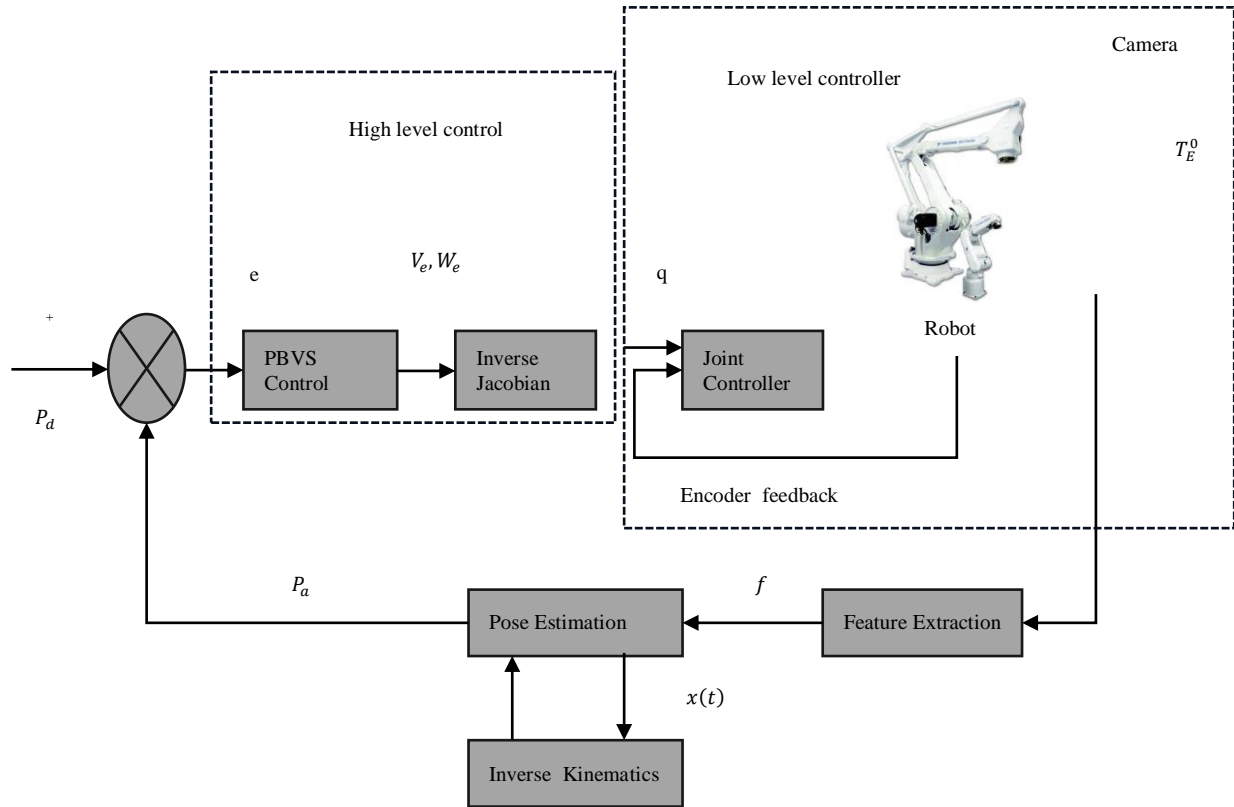


Fig 2. Position-Based Visual Servoing is a Closed-Loop Feedback Approach.

Using this controller, the robot may autonomously navigate around the surroundings in a random manner while looking for the designated color item. Once the robot detects the item, the regulator guides the robot towards it. The controller was designed using the field potential approach, with the objective of producing the actual trajectory that the robot should traverse, considering the data produced by three actions enforced: Avoid Collision(), Follow(), and Go-Ahead(). The primary function, Go-Ahead(), aims to maintain continuous movement of the robot. The Follow() behavior is responsible for propelling the robot towards the destination by using information obtained from the vision module. The last activity, Avoid Collision(), ensures that the robot moves in the contradictory direction of any impediments encountered. The action is shown by a vector of 2D that includes the direction ( $\theta_F$ ) and magnitude  $\theta_{\vec{F}_g}$ ). The action Go-Ahead() does not include the perspective of the environment. The objective of this action is to maintain continuous motion of the robot. A vector, denoted as ( $C_g$ ), is used to depict the uniform field behavior. This vector has a constant direction and magnitude.

$$|\vec{F}_g| = C_g \tag{1}$$

$$\theta_{\vec{F}_g} = 0 \tag{2}$$

In contrast, the Follow() behavior utilizes the visual perception module to guide the robot towards the destination. As previously stated, the output of neural network yields the X and Y coordinates that correlate to the object's location. Subsequently, the output Y has been discretized in the 3D actual world with values that increase by 0.5m. By using this method, it becomes feasible to compute an estimated numerical representation for the 3-dimensional coordinates, specifically referred to as X3D and Y3D. This function takes the input signals X and Y from the MLP network recognition and outputs the duo ([X3D], [Y3D]) in millimeters, which represent estimations of the image's location in the actual habitat. Subsequently, the behavior of Follow() may be executed. The Follow() behavior operates using an attractive force,

denoted by a transmitter with a constant magnitude ( $C_a$ ), and an angle that is rotation that shows the robot's orientation towards the destination. The direction is derived from the information acquired from the module of vision

$$|\vec{F}_a| = C_a \quad (3)$$

$$\theta_{\vec{F}_a} = \arctan(X_{3D}, Y_{3D}) \quad (4)$$

The Avoid Collision behavior is responsible for maintaining a safe distance between the robot and any impediments. In this scenario, the sonar signals are taken into consideration. The Avoid Collision() function of the robot calculates the resulting force by summing together the seven unique vectors generated by its seven sonars, each representing a repulsion force. The Avoid Collision() activity generates a repulsion force via a radial field that follows a decay pattern described as follows:

$$|\vec{F}_{ri}| = \exp((-D + L)|T) \quad (5)$$

The variables in the equation are explicitly specified as follows: The variable  $D$  denotes the length between the obstacle and the robot, while  $L$  indicates the limit of closeness. Additionally,  $T$  is a constant that governs the pace at which the function of exponential declines. The indication of the aversion force, represented by the angle  $\theta$ , is determined by:

$$\theta_{\vec{F}_{ri}} = \arctan(-XS_i, YS_i) \quad (6)$$

The variables  $XS_i$  and  $YS_i$  represent the  $X$  and  $Y$  coordinates, measured in millimeters, collected from each of the seven sonar devices, denoted by  $i$ . The negative indicator is used to reverse the direction, since this action is necessary to maintain a safe distance between the robot and the barriers. The resulting repelling weight is determined by the vector total of the vectors that are individual produced by each sonar, as seen in Equation 7.

$$\vec{F}_R = \sum_I \vec{F}_{ri} \quad (7)$$

The resulting force (robot trajectory) is determined by calculating the forces for each prescribed behavior.

$$\vec{F} = P_1 \vec{F}_G + P_2 \vec{F}_A + P_3 \vec{F}_R \quad (8)$$

The weights of each force in the system are denoted by  $P_i$ , where  $i = 1, 2, 3$ . The values of  $P_i$  in the studies were stated as follows: •  $P1 = 0.5$  •  $P2 = 1.0$  •  $P3 = 0.7$ . Once the resultant weight has been computed by integrating the behaviors, the robot must penetrate using the values:

$$Vel_T = |\vec{F}| Vel_{Max} \quad (9)$$

$$\theta_{\vec{F}} = \arctan(F_X, F_Y) \quad (10)$$

The translation velocity of the robot is denoted as  $Vel_T$ , whereas  $Vel_{Max}$  represents the high speed of the robot. The robot's direction is determined by the  $\arctan F_X, F_Y$  function.

#### IV. RESULTS AND DISCUSSION

The findings are divided into two parts: the first part discusses the system tuning procedure, including parameter design and neural network learning, while the second section details the tests conducted with the Innovator I Robot in an actual-world setting.

##### System Tuning

The first section of this document outlines the conducted tests and the resulting outcomes of the system tuning process. To be more precise, we are presenting the outcomes of the process of learning of the NNs that are accountable for the recognition task and the picture segmentation job. Two algorithms, namely backpropagation [14] and Rprop[15], were used for the learning process. Two different implementations of the backpropagation method were used: batch and on-line mode.

Image Segmentation Neural Network

A database containing 100 patterns was built for the purpose of training the image segmentation neural network. To train the neural network to identify the color red, a dataset consisting of 50 pixels that are red patterns and 50 other colors was generated. This piece included three colors: red, blue, and yellow. By splitting the data bank into two sections—80% for the technique of learning and 20% for testing the set geology—a few experiments were conducted to determine an appropriate neural network topology. These studies revealed that the earth science 3X3X2, which denotes a NN with three RGB neurons of input, three hidden layer neurons, and two output layer neurons, was more suited to the red and blue colors than the yellow color, where the topology 3X5X2 produced superior results. Tests using the 10-Fold Cross-Validation approach were conducted after the identification of an appropriate topology. The outcomes of the learning process employing the following algorithms are shown in Fig 4: On-Line Backpropagation, Rprop, and Backpropagation for the red color (the results were almost same for the blue and yellow colors).

Fig 4 represents the number of iterations required for the neural network to converge, as well as the standard deviation (SD) and the mean square error (MSE) averages during the learning process's ten-fold execution. The findings show that the algorithms' accuracy was almost identical, with the only difference being the number of iterations each learning method utilized. For every approach, the amount of processing time required for learning was about equal. To assess the picture segmentation process's real-time performance, certain Pentium IV 2.26 GHz tests were run.

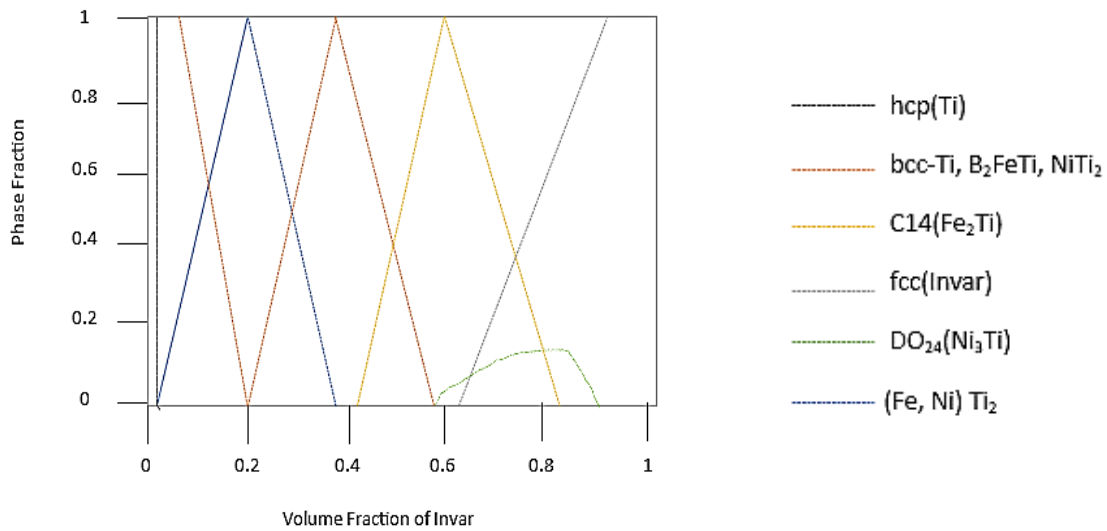


Fig 3. Discretized Sigmoid Function.

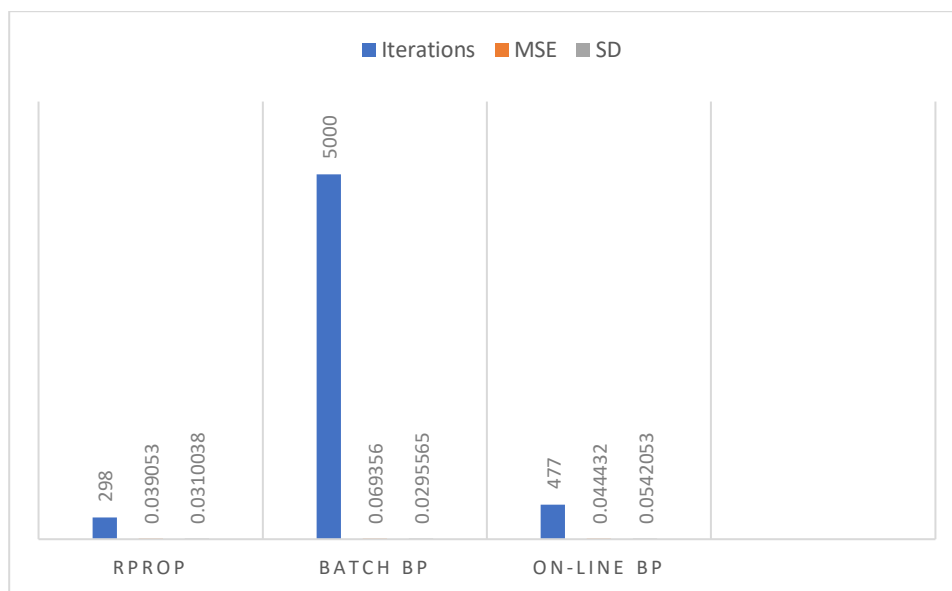


Fig 4. The Image Segmentation Neural Network's Learning Procedure Yields.

The neural network took 0.0125 seconds for the red and blue colors to fragment a picture with 80X60 pixels, and 0.0187 seconds for each frame including yellow.

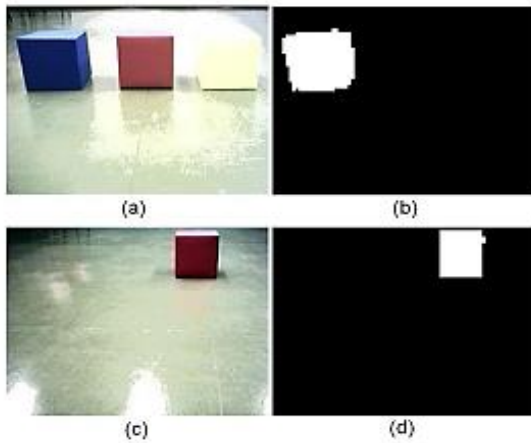


Fig 5. Final Product of Segmenting the Image Using the RGB Color Space.

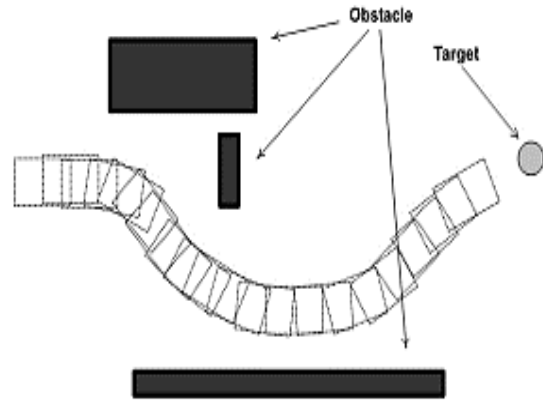


Fig 6. The Robot Operating in A Real-World Setting.

In an effort to enhance the segmentation task's real-time responsiveness, we also tested a disconnected logistic function ( $S_l(x)$ ) (see Fig 3) that approaches the logistic function that serves as the neurons' call up function. The segmentation of the frame for the hues blue, red, and yellow took 0.0062 seconds and 0.0125 seconds, respectively, to complete using this new sigmoid function. The effectiveness of the suggested scheme to work in actual-time setting is further shown by these findings, which highlight the benefit of using this discretized function over the original sigmoid function. Using this function did not result in any quality loss throughout the picture segmentation process. The result of the sectionalization for the blue and red colors is shown in Fig 5. The segmentation outcomes for the  $S_l(x)$  and the original sigmoid function were indistinguishable. In Fig 5, (a) is the original picture 01; (b) is image 01 after being segmented and preprocessed for the color blue. (c) is the original image, and (d) is the segmented and preprocessed image 02 for the color red.

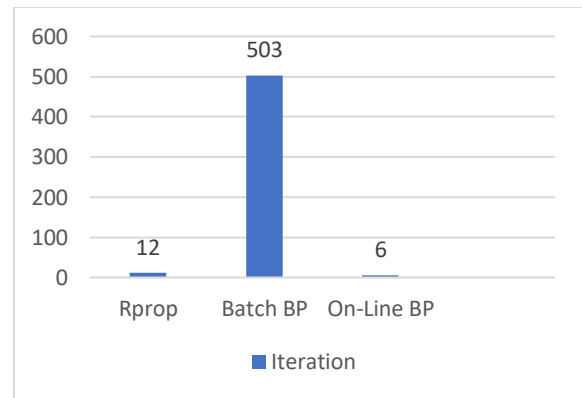
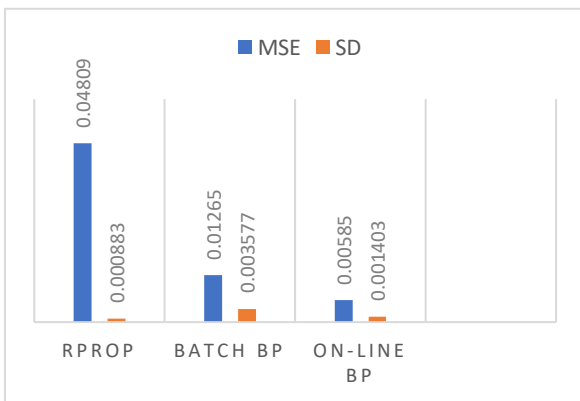


Fig 7. The Outcomes of The Recognition Neural Network's Learning Process.

Recognition Neural Network

The use of artificial neural network models in image processing, where parallel architectures and high computing speed are necessary, is become more appealing. Neural network applications to challenges requiring some level of intelligence or human-like performance have been the subject of several articles published recently. An innovative neural network design for image identification and classification is described in [16]. According to Cios and Shin [17], an object's attribute may be estimated or recognized using a suggested neural network known as an image recognition neural network (IRNN). An analog gray level picture is fed into an IRNN, which outputs an appropriate recognition code.

A database of seven hundred patterns was developed in order to aid the MLP in charge of the recognition job during its learning phase. Every pattern is made up of a preprocessed, segmented frame (a binary picture of 80 by 60 pixels) and the label for that frame, which indicates where the item is in the visual field. We also partition the database into two portions, 20% for testing the configured neural network and 80% for the process of learning, in order to determine the topology of this neural network. The three algorithms—On-Line Backpropagation, Batch Backpropagation, and Rprop—were used to determine that the topology 4800X10X3 is suitable for accomplishing this recognition task. The design of this neural



network comprises of 10 cells in the layer that is hidden, 3 cells in the outer layer (representing  $X, Y,$  and  $P$ ), and 4800 cells in the inner layer (corresponding to  $80 \times 60$  pixels of binary). The cross-validation technique, using a 10-fold approach, was conducted to the neural network set with dimensions of  $4800 \times 10 \times 3$ . The results of this procedure, employing the three designed algorithms, are presented in **Fig 7**. **Fig 7** displays the almost identical results of the two methods, with the benefit that the Rprop algorithm ran quicker than the others.

*Experimentations*

Following system calibration, a number of tests using the Pioneer I robot were conducted to evaluate the CVS's performance in an actual setting. A Creative WebCam Go Plus and a Pentium III 500MHz laptop were used to run the CVS. As stated in [18], the purpose of the first studies was to determine appropriate parameters for the behavior force weights. These tests were conducted in a basic setting in order to fine-tune the system's ability to track a target and avoid obstacles without colliding. Following the first trials, which were carried out to determine certain system parameters, we constructed many scenes in a real setting with arbitrary barriers and a target that was also positioned arbitrary inside these scenes (as shown in **Fig 6**). In every trial conducted with the determined settings, the robot demonstrated navigational skills by dodging barriers and tracking a target—characterized by a distinct color—when it entered its range of vision.

Wolf and Sukhatme [19] conducted an experiment in which they generated a semantic map inside a real-world setting. **Fig 8** displays the robot and the environment we used, respectively. The laboratory room doubles as the researcher's study area and living quarters for experiments. In this experiment, we show that SpCoMapping can learn terminologies as location names without establishing them by extracting word information from provided phrases. To demonstrate how many words may be associated to a location without the place names pre-setting, we utilized lines as word characteristics. We gave each of these 20 lines—which include 50 vocabulary words—five times. Approximately 407 times, we gave the RGB information. We used SpCoMapping in this experiment since we did not know how many spatial notions there were. We have 120 as the maximum number of spatial ideas. The hyperparameters that we have specified are as shown:  $\gamma = 1.0 * 10^6, \beta = 0.6, \chi = 100.0,$  and  $\gamma = 4.0$ . Using the *tf-idf* technique as collective data between sentences and words, we establish the weight for vocabulary features. The following equation was used to determine each word's weight in the sentence.

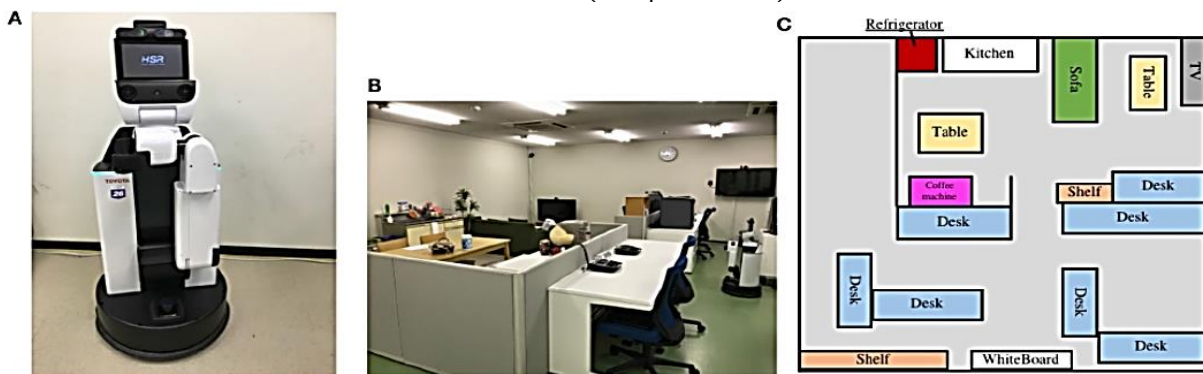
$$Weight_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \log \frac{D}{D_i} \tag{11}$$

where  $n_{i,j}$  is the total word count in the phrase  $j, D$  is the quantity of phrases, and  $D_i$  indicates how many sentences there are, including word  $i$ . The weights of words that are used in numerous phrases, such as “is,” “here,” and “you,” are decreased by using the *tf-idf* algorithm. This procedure aids in the learning of vocabulary associated to locations. The suggested approach will be stable if we sample  $w_l$  100 moments and use the mean as  $w_l$ . Furthermore, we used word information to facilitate pre-learning via spatial concept formation. Klein et al. [20] computed 1000 iterations while keeping the hyperparameters set to the same values. The scholars initialized  $\pi, \mu$  in Algorithm 1 line 1 using the pre-learning outcome. In Analytic 1, line 2, we set the value of  $C_{i,j}$  as follows: If  $m_{x,t}$  is not an unoccupied area, then  $C_{i,j}$  lack any dimensional concept.

$$C_{i,j} = 0 \tag{12}$$

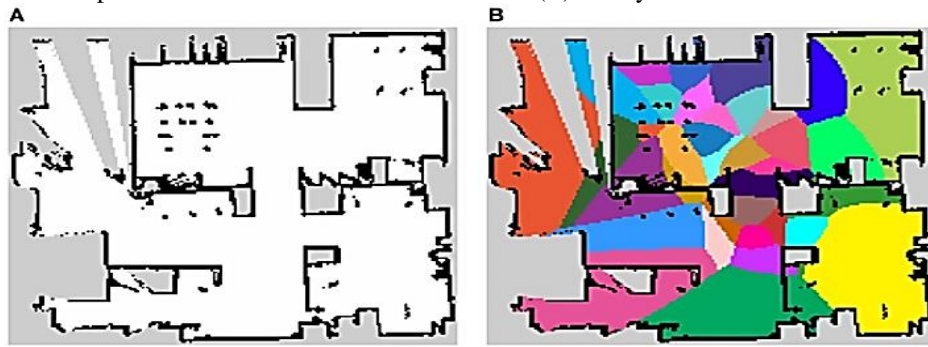
The sampling equation may be expressed as:

$$C_{ij} = argmax_N \left( (i,j) \mid \mu_i^{pre}, \Sigma_i^{pre} \right) \tag{13}$$



**Fig 8.** The robot and an example of trials conducted in an actual setting. A robot used for conducting research on an individual's daily habitat Human Support Robot. We used the HSR developed by TOYOTA, which is equipped with the Xtion PRO LIVE sensor manufactured by ASUS for capturing RGB information, and the UST-20LX sensor manufactured

by Hokuyo for obtaining laser-range information. (A) The PC was equipped with ROS Kinetic and Ubuntu 16.04. (B) The depiction of the habitat used in this test and (C) the layout of the habitat.



**Fig 9.** The cartographic representations used in the trials. (A) The map representing the distribution of occupied spaces in an individual living habitat and (B) the outcome of prior studying.

The distribution of multivariate Gaussian is represented as  $N(\mu_{pre}, \Sigma_{pre})$ , where  $\mu_{pre}$  is the median beeline of the position dispersion for the  $l^{th}$  pre-studying group, and  $\Sigma_{pre}$  denotes the matrix of covariance of the transformation of position for the  $l^{th}$  category of pre-studying. Utilizing pre-studying enhances the stability of language acquisition characteristics and reduces the number of iterations needed for learning. **Fig 9** displays the ownership map of grid of the user's staying habitat and the pre-learning result. The ownership map of grid has 19,255 pixels that are categorized as unoccupied space.

## V. CONCLUSION

This research introduced a computer vision system comprising of preprocessing, segmentation, and image identification, with a module for navigation control. It was decided that the responsive model is well suited for the design of robots operating in a realistic, dynamic environment. The vision module used a multilayer perceptron (MLP) neural network for color segmentation, image segmentation, and object recognition. The navigation control module used position-based visual servoing (PBVS) to guide the robot and incorporated Go-Ahead(), Follow(), and Avoid Collision() functions to facilitate obstacle avoidance and navigation. Study Active Media Robotics with the Go Plus Camera of Creative Webcam was developed on the Pioneer I robot. The tuning process includes parameterization using the backpropagation RProp algorithm and learning a neural network. The findings indicated that the algorithms exhibited comparable levels of accuracy, differing mainly in the number of iterations used by each learning approach. The segmentation task's real-time performance was evaluated using Pentium IV 2.26 GHz testing. The research also examined the use of models of artificial neural network in processing of image and their capacity to estimate or identify object properties. A further experiment was carried out using SpCoMapping to produce a semantic map in an actual environment. The researchers used sentences as lexical features to illustrate the extent to which words might be linked to a specific area without prior specification of place names.

### Data Availability

No data was used to support this study.

### Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

### Funding

No funding agency is associated with this research.

### Competing Interests

There are no competing interests.

### References

- [1]. A. L. De Assis Simões, A. Pinto, J. S. Baptista, S. D. Pinheiro, and D. Romero, "Designing human-robot collaboration (HRC) workspaces in industrial settings: A systematic literature review," *Journal of Manufacturing Systems*, vol. 62, pp. 28–43, Jan. 2022, doi: 10.1016/j.jmsy.2021.11.007.
- [2]. J. Trevelyan, W. R. Hamel, and S. Kang, "Robotics in hazardous applications," in *Springer handbooks*, 2016, pp. 1521–1548. doi: 10.1007/978-3-319-32552-1\_58.
- [3]. M. Hägele, W. Schaaf, and E. Helms, "Robot Assistants at Manual Workplaces: Effective Co-operation and Safety Aspects," *Proceedings of the 33rd ISR (International Symposium on Robotics)*, Jan. 2002.
- [4]. H. Tian, T. Wang, Y. Liu, X. Qiao, and Y. Li, "Computer vision technology in agricultural automation —A review," *Information Processing in Agriculture*, vol. 7, no. 1, pp. 1–19, Mar. 2020, doi: 10.1016/j.inpa.2019.09.006.

- [5]. A. Moniz and B.-J. Krings, “Robots Working with Humans or Humans Working with Robots? Searching for Social Dimensions in New Human-Robot Interaction in Industry,” *Societies*, vol. 6, no. 3, p. 23, Aug. 2016, doi: 10.3390/soc6030023.
- [6]. M. Quigley, “ROS: an open-source Robot Operating System,” *International Conference on Robotics and Automation*, Jan. 2009, [Online]. Available: <http://ci.nii.ac.jp/naid/10030771702>
- [7]. R. M. Cyert and M. H. DeGroot, “Rational expectations and Bayesian analysis,” *Journal of Political Economy*, vol. 82, no. 3, pp. 521–536, May 1974, doi: 10.1086/260210.
- [8]. V. Vijaya, R. Valupadasu, B. R. Chunduri, C. Rekha, and B. Sreedevi, “FPGA implementation of RS232 to Universal serial bus converter,” *2011 IEEE Symposium on Computers & Informatics*, Mar. 2011, doi: 10.1109/isici.2011.5958920.
- [9]. H. Utz, S. Sablatnög, S. Enderle, and G. K. Kraetzschmar, “Miro - middleware for mobile robot applications,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 493–497, Aug. 2002, doi: 10.1109/tra.2002.802930.
- [10]. R. Linker, O. Cohen, and A. Naor, “Determination of the number of green apples in RGB images recorded in orchards,” *Computers and Electronics in Agriculture*, vol. 81, pp. 45–57, Feb. 2012, doi: 10.1016/j.compag.2011.11.007.
- [11]. C. Fahn, C. Lee, and Y.-S. Yeh, “A real-time pedestrian legs detection and tracking system used for autonomous mobile robots,” *2017 International Conference on Applied System Innovation (ICASI)*, May 2017, doi: 10.1109/icasi.2017.7988208.
- [12]. A. Vijayan and S. Ashok, “Comparative study on the performance of neural networks for prediction in assisting visual servoing,” *Journal of Intelligent and Fuzzy Systems*, vol. 36, no. 1, pp. 675–688, Feb. 2019, doi: 10.3233/jifs-171475.
- [13]. B. Cheng, Z. Li, J. Jiao, and G. An, “MLP Neural Network-Based Precise Localization of robot assembly parts,” in *Lecture Notes in Computer Science*, 2023, pp. 608–618. doi: 10.1007/978-981-99-6480-2\_50.
- [14]. Y. LeCun et al., “Backpropagation applied to handwritten Zip code recognition,” *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989, doi: 10.1162/neco.1989.1.4.541.
- [15]. M. Riedmiller and H. Braun, “A direct adaptive method for faster backpropagation learning: the RPROP algorithm,” *IEEE International Conference on Neural Networks*, Dec. 2002, doi: 10.1109/icnn.1993.298623.
- [16]. R. Vankdothu, M. A. Hameed, A. Ameen, and R. Unnisa, “Brain image identification and classification on Internet of Medical Things in healthcare system using support value based deep neural network,” *Computers & Electrical Engineering*, vol. 102, p. 108196, Sep. 2022, doi: 10.1016/j.compeleceng.2022.108196.
- [17]. K. J. Cios and I. Shin, “Image recognition neural network: IRNN,” *Neurocomputing*, vol. 7, no. 2, pp. 159–185, Mar. 1995, doi: 10.1016/0925-2312(93)e0062-i.
- [18]. M. G. Quiles and R. A. F. Romero, “A Computer Vision System based on Multi-Layer Perceptrons for Controlling Mobile Robots,” *18th International Congress of Mechanical Engineering*, Jan. 2005, [Online]. Available: [https://www.abcm.org.br/symposium-series/SSM\\_Vol2/Section\\_X\\_Computer\\_Vision/SSM2\\_X\\_05.pdf](https://www.abcm.org.br/symposium-series/SSM_Vol2/Section_X_Computer_Vision/SSM2_X_05.pdf)
- [19]. D. F. Wolf and G. S. Sukhatme, “Semantic mapping using mobile robots,” *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 245–258, Apr. 2008, doi: 10.1109/tro.2008.917001.
- [20]. A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, “Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets,” *Proceedings of Machine Learning Research*, pp. 528–536, Apr. 2017.