automation

MDPI

Communication

Design and Implementation of a Robotic Arm Assistant with Voice Interaction Using Machine Vision

George Nantzios *, Nikolaos Baras * and Minas Dasygenis

Department of Electrical and Computer Engineering, University of Western Macedonia, 501 00 Kozani, Greece; mdasyg@ieee.org

* Correspondence: gnantzios@gmail.com (G.N.); nbaras@uowm.gr (N.B.)

Abstract: It is evident that the technological growth of the last few decades has signaled the development of several application domains. One application domain that has expanded massively in recent years is robotics. The usage and spread of robotic systems in commercial and non-commercial environments resulted in increased productivity, efficiency, and higher quality of life. Many researchers have developed systems that improve many aspects of people's lives, based on robotics. Most of the engineers use high-cost robotic arms, which are usually out of the reach of typical consumers. We fill this gap by presenting a low-cost and high-accuracy project to be used as a robotic assistant for every consumer. Our project aims to further improve people's quality of life, and more specifically people with physical and mobility impairments. The robotic system is based on the Niryo-One robotic arm, equipped with a USB (Universal Serial Bus) HD (High Definition) camera on the end-effector. To achieve high accuracy, we modified the YOLO algorithm by adding novel features and additional computations to be used in the kinematic model. We evaluated the proposed system by conducting experiments using PhD students of our laboratory and demonstrated its effectiveness. The experimental results indicate that the robotic arm can detect and deliver the requested object in a timely manner with a 96.66% accuracy.

Keywords: robotics; robotic arm; niryo-one; machine vision; YOLO; opencv; assistance robot

check for updates

Citation: Nantzios, G.; Baras, N.; Dasygenis, M. Design and Implementation of a Robotic Arm Assistant with Voice Interaction Using Machine Vision. *Automation* **2021**, *2*, 238–251. https://doi.org/10.3390/automation2040015

Academic Editor: Felipe N. Martins

Received: 31 August 2021
Accepted: 26 October 2021
Published: 31 October 2021

1. Introduction

The rapid development of the Information Technologies in the fields of hardware, along with the continuous reduction of the embedded systems manufacturing cost, has stimulated engineers, scientists, and hobbyists to develop advanced robotic systems at low-cost [1–3]. Technologies, such as Machine Vision and Artificial Intelligence have made feasible the usage of robotics in various areas of people's daily life, such as social and health care of the elderly [4,5], and this has dramatically increased their sales (Figure 1), according to the International Federation of Robotics [6]. Technological progress has been conducted to a continuous and increasing communication and social interaction between humans and robots. Taking a closer look into social and interactive robots that are in constant communication with humans, such as robots in roles of assistants, we observe that several attempts have been made to enhance the experience further [7,8]. However, it is obvious that the majority of these are still challenging to obtain due to their unaffordable cost [9] and not always easy to adapt to the needs of each user.

Except the cost, we must note the society attempts to adapt to the automation trends that are taking place. The gradual increase in the popularity of robots in our everyday life, has created new foundations in the relationship between humans and robots. People treat them as tools to help and even relieve them of repetitive actions when needed. Now we can interact through natural language with robots. As a result, this more direct contact replaces the cold and unnatural interaction, typically performed using buttons and levers.

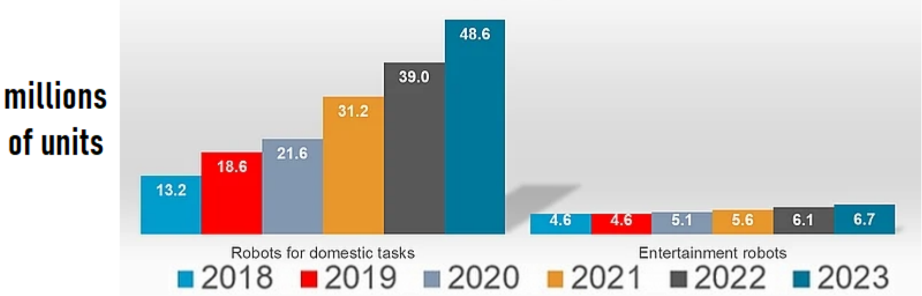


Figure 1. Unit sales and potential development for 2021–2023 for personal and domestic service robots, according to World Robotics, showcasing the increasing trend in using robots for everyday tasks.

Furthermore, the spread of robotics and its innovations could not entail negative implications regarding the future of jobs. According to information from Oxford Economics, jobs that will be most affected are those that require the fewest skills. More than a few analysts argue that if automation has contributed to more jobs than the ones being eliminated, it has also created a rift between higher and lower-skilled jobs, leaving many workers on the shelf [10].

Soon, every consumer will own a robot which will be imperative to provide assistance in everyday tasks. Certain groups of people will benefit greatly from these robot assistants, for example the aging population and people with special needs. The technology however in this application domain is still developing. Developing effective, efficient and cost-effective robot assistants requires improving existing algorithms or designing new ones. Here, we present our contribution to this domain. This research focuses on developing a robotic arm for voice interaction to detect and recognize objects, picking them up, and delivering them to the user. This project aims to combine machine vision and voice interaction with developing a robotic system that will act according to the user commands. We have modified the well-known YOLO object detection algorithm to enhance its functionality to be used for robotic applications. At the same time, an intuitive HRI (Human Robot Interaction) voice-controlled software has been developed to provide ease of use to everyone, even to technology illiterate people.

The main novelty and scientific contributions of our research are:

- we present an extension on the YOLO algorithm to detect objects at an angle
- we seamlessly orchestrated several technologies and developed a modular system, meaning that it can be easily customized further, unlike other implementations found in the literature
- we provide a low implementation cost solution to the robotic arm assistant problem

The remainder of the manuscript is structured as follows: Section 2 presents related research works to our project, Section 3 provides necessary information regarding the tools and techniques that we used, Section 4 highlights the implementation of the proposed system, Section 5 discusses the safety and security of the robotic system, Section 6 presents information regarding the application domain of cloud computing and robotics and Field Programmable Gate Arrays. Finally, Section 7 presents the experimental results and Section 8 gives the conclusions.

2. Related Work

Over the years, there has been a progressive increase in the motivation for young scientists and researchers to engage in the fields of machine vision and robotics [11]. This has resulted in the creation of new applications and techniques that provide solid solutions to various issues of concern to the scientific community. Over the years, robots have been employed in various domains, such as manufacturing [12,13] and healthcare [14]. To date, there are several examples of applications that have been developed to create a robotic arm to recognize and deliver objects with voice interaction but most of them do not provide all the functionality of our implementation.

The robotic application presented in [15] is based on a 4-degree-of-freedom robotic arm and the recognition is achieved only for pretrained objects in concert with our research which can detect and deliver unknown objects. In order to accomplish this, the Region Proposal Network RPN architecture using R-CNN (Convolutional Neural Network) with inception v2 has been used through the TensorFlow framework. At the same time, communication with the user is achieved through voice commands and a graphical interface. This approach allows the user to identify the objects he or she uses most often. However, the training process, the successful recognition rate, the limited workspace, and the camera placement in the surrounding area of the robotic arm make the system difficult to use on a daily basis due to the inability to quickly adapt to changes in the user's needs and different usage environments.

Pulokottil et al. [16] propose a robotic system that is mainly targeted at patients with mobility and vision problems. Specifically, they developed a robotic application in the ROS (Robotic Operating System) framework based on the JACO2 arm that does not support machine vision. It allows being operated either by voice commands or by a joystick. Communication with the user is accomplished using the CMU Pocketsphinix API (Application Programming Interface). This project was tested by two patient volunteers, and it was rated in different tests that were performed. Based on the results, the users would use the robotic system most for lifting and moving objects and preferred the robotic system to be placed in both a fixed position or attached to a wheelchair. However, practice is required to perform an action. It is necessary to guide the robotic arm step by step, while estimating the distance between the end-effector and the object in order to give the appropriate voice command "long" or "small". Thus, practical issues of application usability arise in both manipulation and guidance, which compel the user to have their attention constantly focused on the arm movement environment.

Bandara et al. [17] present a robotic assistant system featuring gesture recognition and voice operation. The aforesaid system consists of a 6-degree-of-freedom arm that has capabilities of lifting and positioning objects in the workspace. The recognition of shapes and colors is implemented by Kinect V2 sensor in combination with the application developed in C#.Net with algorithms from Aforge.Net. Microsoft Speech platform was used for voice operation recognizing the English language. The voice commands follow a specific dictionary, and the calculation of the joint angles is obtained using inverse kinematic model functions implemented in MATLAB. A significant drawback based on the results is considered to be the failure rate of actions taken, which is caused by coordinate errors in each axis as there is a lack of feedback from the joint motors. According to the results, a significant disadvantage is the failure rate of actions taken, which is caused by coordinate errors in each axis due to a lack of feedback from the joint motors.

The presented robotic system differs from the aforementioned ones as it combines three different types of object recognition: (a) color and shape, (b) classes of objects, (c) QR code tags and gives great variety in terms of object recognition. At the same time, communication is achieved bidirectionally with both English and Greek language support, and the installation and operation process does not require any configuration or adjustments by the user. Furthermore, the basic object recognition program can also be executed in real time on the Raspberry Pi 4, making our robotic system in that way portable.

3. Materials and Methods

This section presents the tools and techniques that were used in this research. These tools are the fundamental elements in which this research is based on. Section 3.1 presents the shape and color detection techniques that were used in order to properly identify objects. The original YOLO v3 algorithm is presented in Section 3.2 and the modifications we have performed are presented in Section 3.3. Finally, the QR code expandability technique is introduced in Section 3.4.

3.1. Shape and Color Detection

In the present implementation the shape and color object detection function was chosen to detect the three basic colors red, green and blue and as far as shape is concerned the square, rectangle, triangle, pentagon and circle were detected. However, it should be mentioned that the algorithm can be modified to recognize any color. The first step for processing each frame is to apply a Gaussian blur filter kernel. The selection of Gaussian blur was based on the result it delivers in combination with the morphological manipulations that follow. The frame is then converted from the BGR (Blue Green Red) to the HSV (Hue, Saturation, Value) color space so that the desired color is then isolated. Following this, morphological manipulations are applied starting with the erosion and dilation filter. After all the morphological manipulations are completed, the visual result obtained is the isolation of objects from the background. Thereafter, the next step is to find the contours. Essentially this is the curve that joins all the continuous points along the object boundaries. For these bounds, a box is drawn to enclose them, and a bounding rectangle is drawn on their minimum surface, resulting in a calculated angle of rotation of the object. In the frames formed, the center of mass of the object is calculated which will be the final position coordinates of the object in the image.

To determine the shape, the list of two-dimensional space points enclosing the objects is used. The perimeter of the contour is first calculated and then a function is used to approximate the number of line segments formed by the points in the contour list. This approximation is based on the assumption that a curve can be approximated by smaller line segments and that the contour list consists of a list of vertices. The number of entries in this list is then checked to determine the shape of an object. For example, if the approximate contour has three vertices, then it must be a triangle. If an outline has four vertices, then it must be either a square or a rectangle. To determine which one it is, the aspect ratio of the shape is calculated, which is the width of the contour bounding box divided by the height. If the aspect ratio is approximately 1.0 then it is square (since all sides are approximately equal in length) if not then the shape is rectangular. Otherwise, by the process of elimination the shape must be a circle.

3.2. YOLO Algorithm

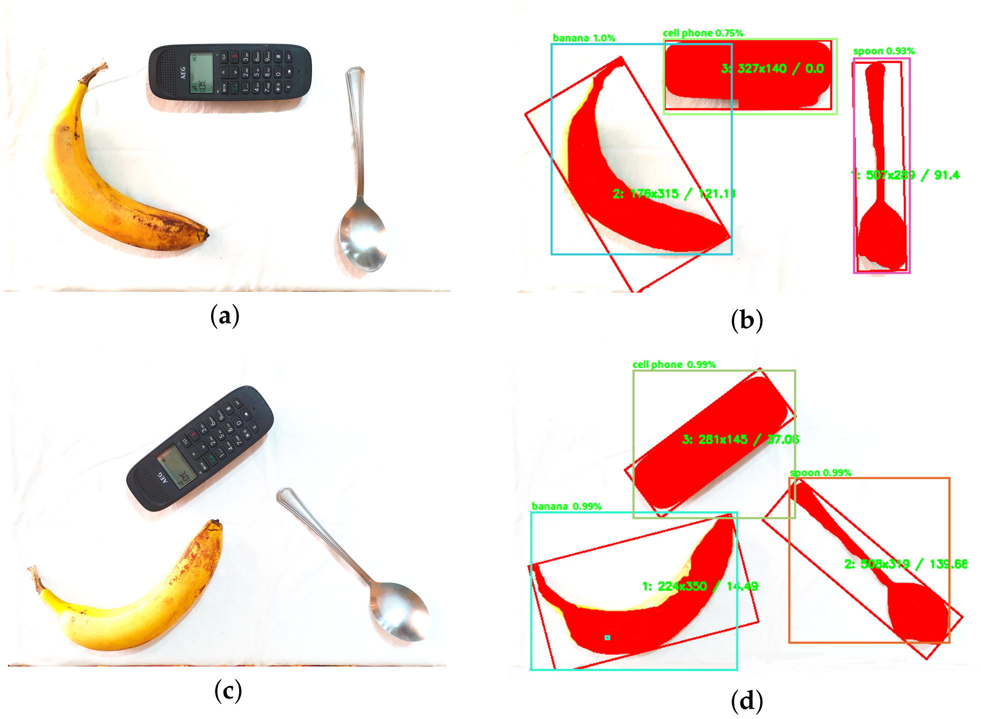
The object recognition as mentioned above is achieved using the YOLO v3 [18]. YOLO is an abbreviation for the term 'You Only Look Once'. YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect instances of various classes of objects (like a spoon, book, cell phone, remote, banana, etc.) The algorithm has gained popularity because of its superior performance over the aforementioned object detection techniques [19]. Its capabilities, in addition to speed and accuracy, are the simultaneous detection of many different objects in an image and their localization in space. YOLO achieves state-of-the-art results beating other real-time object detection algorithms by a large margin. Important features of the algorithm include:

- Speed: This algorithm improves the speed of detection because it can predict objects in real time.
- High accuracy: YOLO is a predictive technique that provides accurate results with minimal background errors.
- Learning capabilities: The algorithm has excellent learning capabilities that enable it to learn the representations of objects and apply them in object detection.

In the present implementation, the Darknet Framework was chosen to be used. This choice was made to achieve the highest possible FPS (Frames Per Second) since the execution of the algorithm would have to be executed by the GPU (Graphics Processing Unit). The algorithm was run on a system with AMD Ryzen™ Threadripper™ 2950X 64 GB RAM (Random Access Memory) and Nvidia GTX 2080 GPU with 6 GB VRAM (Virtual Random Access Memory) and the average number of FPS generated by the algorithm is 25 FPS in its full version. The COCO dataset was used for object recognition since it provides a collection of 80 classes of everyday objects. The selection criterion to consider an object recognized has been set by a confidence threshold of more than 45%. If this percentage increases, it means that detection is more accurate, but there may be fewer objects detected and vice versa. The algorithm generates the x and y coordinates of the centers of the detected objects, and the confidence percentage for the class label it recognized.

3.3. Our Proposed YOLO Algorithm Extension

Each object which is successfully detected by the available detecting methods should be followed by three basic attributes, the name tag, the coordinates of the center and the rotation angle it forms both relative to the camera position. The label and coordinate attributes are computed from all detecting methods. The angle attribute, however, is not computed by the YOLO algorithm. It would not be possible to skip these data, as it is necessary for the correct positioning of the final arm action tool to grasp the object. For this reason, an extension to the YOLO algorithm was developed which essentially gives the object rotated angle. To implement this type of detection, some functions were developed which takes as input the current frame from the camera and applies appropriate filters, and morphological manipulations. By applying these filters, the visual isolation of objects from the background is achieved. The next step is to find the contours which are closed curves that are obtained from edges and depicting a boundary of objects. Then we calculate the image moments which actually are how pixel intensities are distributed according to their location. It is a weighted average of the object pixel intensities, and we can obtain the centroid from the object. The final step consists of the drawing of the minimum area rotated rectangle which we can eventually calculate the rotation. Figure 2 visualizes the extension on the YOLO algorithm using 3 objects as an example.



(**a**) (**b**)

(**c**) (**d**)

Figure 2. Images (**a**,**c**) are the original frames taken from the camera. Images (**b**,**d**) show the extension of the YOLO algorithm and they include the name tag, center coordinates and the rotation angle of the detected items.

3.4. Enhancing the Detection Functionality Using the QR Code Technique

To increase the expandability of the system, we designed and implemented a system based on QR codes. The core idea is that any item can be detected by the system, given that it has a unique QR code on the visible area. It is clear that the dimensions of the QR code tag should be sufficient for the camera to read it. In our case, the minimum detected QR code by our camera Logitech C922 720P was computed to be 2.5 cm × 2.5 cm. By using this system, we can avoid the training of the object detection algorithm, a time-consuming process, each time we want to add a new object. The decision to use QR codes (2D barcodes) was made because of their advantages over barcodes. Specifically, they cover a smaller space on the surface of the object and they have a low error rate and can be read from any camera angle and position. The combination of all these advantages manages to fully serve the use cases and purposes of the present implementation. The detection of QR codes is based on the Zbar library making use of the Python module. To start the operation, the video stream from the camera is introduced as input to the decoding function. As output, the function returns a table of detected strings from the QR codes. The strings are actually the name preferred by the user to call the objects and they can be in English or Greek language as well. This allows multiple QR codes to be detected for each frame entered. In the next step, this table is traversed so that for all QR codes the name field and the 4 points in the space surrounding it are extracted. At the end of the process in the object tracking window, the bounding box is drawn with the name tag and the center coordinates of the QR code.

4. Implementation

The software that accompanies robotic arm prototype is complex, as it combines several technologies, such as voice recognition, voice synthesis, machine vision and kinematics. The execution flow of the software is necessary to be clear and precise in order to avoid malfunctions that could potentially damage the robotic arm itself or the user. Figure 3 presents the flow chart of the proposed system.

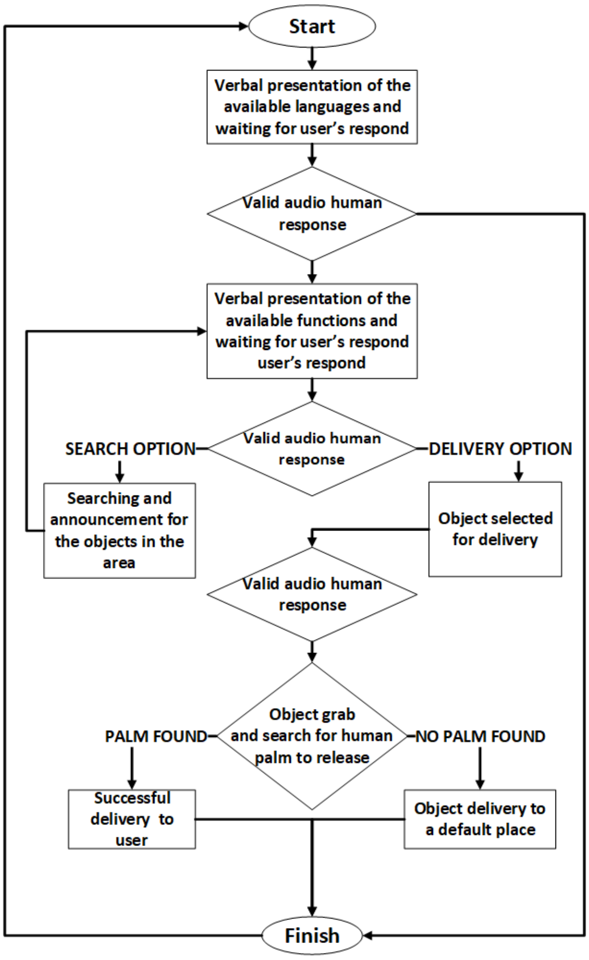

Start
Verbal presentation of the available languages and waiting for user's respond
Valid audio human response
Verbal presentation of the available functions and waiting for user's respond user's respond
SEARCH OPTION
Valid audio human response
DELIVERY OPTION
Searching and announcement for the objects in the area
Object selected for delivery
Valid audio human response
PALM FOUND
Object grab and search for human palm to release
NO PALM FOUND
Successful delivery to user
Object delivery to a default place
Finish

Figure 3. Flowchart of the proposed robotic arm system.

4.1. The Robotic Arm

The robotic arm used for the needs of this study is the Niryo-One (Figure 4) by the French company Niryo (Niryo-One—An accessible educational 6 axis robotic arm—https://niryo.com/niryo-one/, accessed on: 30 October 2021). The Niryo-One is the first and only 3D printed 6 Degrees Of Freedom (DOF) robotic arm offered by the company and thus our study can be easily in hardware replicated in every consumer home that has access to a 3D Printer. The STL files for printing the robot parts as well as the source code are available for free. The community aspect of this robot is an essential part of the company, whose main goal is to make robotics more accessible by providing low-cost, easy-to-use robots. Furthermore, it strives to build a complete set of services around Niryo-One as well as a community centered on open-source projects. This model is designed for educational purposes, introduction to industry 4.0, professional training as well as research labs. There are many ways of programming Niryo-One

- Programming the robot with the learning mode in which the user can move the robot manually to any position he wishes so that can be stored by the robot and be executed afterwards.
- Programming the robot in Python language using the provided Python API
- Connecting the robot to other devices such as Arduino and Raspberry Pi via digital interfaces
- Finally, the ROS source code is available for modification and customization to meet the needs of the user using Python or C++.

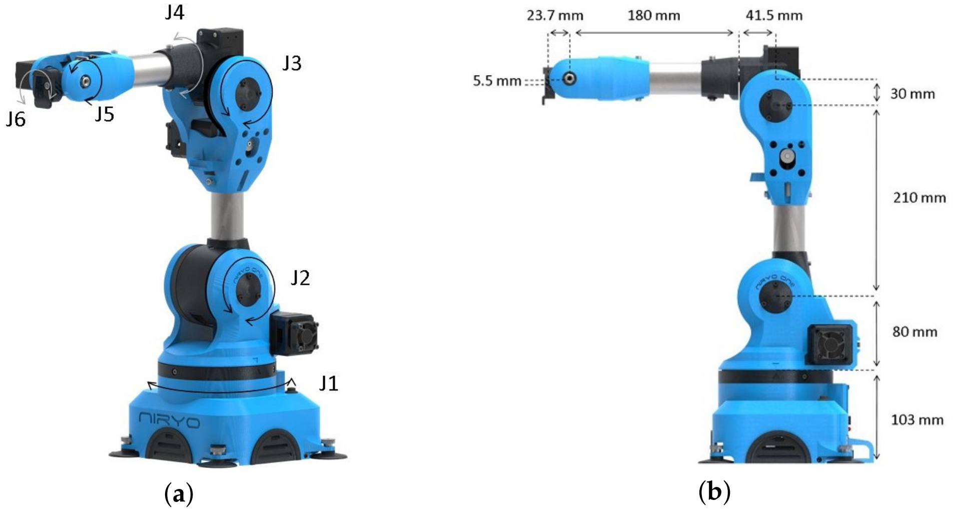


Figure 4. The Niryo-One robot. Image (**a**) shows the 6 joints of the robotic arm; image (**b**) presents its dimensions. (Image courtesy of https://niryo.com/, accessed on: 30 August 2021).

Niryo-One can be operated by adjusting the X, Y, and Z planes, as well as the roll, pitch, and yaw for wrist orientation, or by adjusting the degrees of torsion for each joint. The base area houses the arm control and processing unit which consists of the Raspberry Pi 3 and an expandable shield responsible for driving the actuators. For the purposes of this study, the GRIPPER 2 was chosen to be attached as the jaw opening and closing mechanism allow an optimal fit for everyday objects.

4.2. Voice Interaction

Voice interaction is crucial because it consists of a user-friendly communication to the robotic system. The purpose of this functionality is initially to successfully recognize voice commands and then deliver messages in an understandable way to the user. Of course there is always the advanced interaction using SSH (Secure Shell) and Python, but they do not belong to the scope of this paper. The correct and uninterrupted operation of the voice recognition system plays a vital role. In case of any failure, the robotic system and its functions are rendered useless. This function is incorporated in the core Speech code script developed to perform the basic execution routine of the robotic system. The implementation of the voice recognition system was based on the SpeechRecognition library, which supports various recognition engines and APIs. For the needs of the implementation, Google Voice Recognition API was chosen [20] which required online connection. This choice was made after testing and evaluating other speech recognition engines such as CMU Sphinx in both English and Greek. The Google API achieved the highest number of successful recognitions in the least amount of time when tested for the Greek language. The voice response function is used whenever the system wants to communicate messages to the user. The user is continuously informed with voice messages when the system is ready to execute commands and at the start and finish of each action performed by the system. The system also informs the user in the case of failure to find an object or to recognize a command. The function is implemented using the pyttsx3 text-to-speech library (Text-to-Speech (TTS) library for Python 2 and 3—https://pypi.org/project/pyttsx3/, accessed on: 30 October 2021).

4.3. Robotic Arm Movement

An undoubted factor of the effectiveness of the present research is the accurate execution of the commands by the robotic arm and, finally, the object delivery to the user's hand. The precondition for the correct placement of the arm and the orientation of the end-effector are coordinates from object's center of mass and its rotation angle. The main functions that are implemented are the search function, conversion of the object's coordinates to the position of the arm in space, approach and grasping the object, and finally, its release into the user's hand. The communication between the script and the Niryo-One robotic arm is achieved through Python API. The primary function of the script converts the center coordinates of the object to the appropriate pose that the arm should take. The center coordinates consist of values ranging from 0 to 640 for the horizontal axis and 0 to 480 for the vertical axis. These values refer to the number of pixels present in the window "seen" by the camera since the input resolution is downsampled to 640 × 480 to be easily computed in the embedded system platform. As we have discovered, this does not incur a loss in the detection accuracy. The position of the arm can have values for the x, y, z planes that define the position of the arm in space and roll, pitch, yaw for the orientation of the end-effector. The end-effector is pointed downwards so that the camera captures objects in the surrounding area. This position has been chosen as the camera is in full verticality and in a good position to identify and include enough objects in its capture frame. The search function is configured to cover the area around the periphery of the arm to a total range of 220 degrees. To cover this space, it has been divided into five sections, with the corresponding arm positions. In order to relate the coordinates from the camera's frame to the locations corresponding to the space. Initially, it is necessary to define the movement of the end-effector within each of these frames. To achieve this, the ability to "live" update the pose values for each arm position via the API was used. By noting points in the space that coincide with the corners of the five camera frames and placing the end-effector on them, it was possible to capture all the endpoints of space where the arm can move. The value range of the physical boundaries is mapped to the value range of the shooting frame. By entering the coordinates, the corresponding values in physical space are obtained. As for the orientation of the end-effector in approaching and grasping the object, it is perpendicular to the upper side of the object. The roll value of the end-effector depends on the rotation angle of the object.

5. Safety and Security of the System

When developing a system targeted to people with movement impairments (and potentially children), it is necessary to take safety into consideration. The safety of the system is defined as the condition of being protected from or unlikely to cause danger, risk, or injury. In our case, the safety of the robotic arm includes the immediate stop in emergency situations, the minimization of harm caused by materials, fire and electrical shock prevention and software security.

In extremely rare cases that the robotic arm malfunctions, the user may want to immediately terminate the execution and stop the movement of the robot. For this reason, we have programmed a hardware button located on top of the Niryo robot to immediately stop all actions and halt. To achieve this, the button is programmed to kill all currently running threads, including object detection, voice control and movement threads. As a result, when the emergency button is pressed, the robot stops.

To improve the physical safety of the robotic arm, we have taken into consideration the shape and materials. First of all, the interactive parts of the robot are mostly made out of PLA (polylactic acid), a material widely used in robotics and 3D printed models. This material is biodegradable, and it is often used in food handling and medical implants that biodegrade within the body over time. This means that even if parts are accidentally ingested, they will not cause serious damage. Another great benefit of PLA material is that it is an insulator and minimizes the risk of electrical hazards that could potentially cause fire. For this reason, the end-effector of the robot is made mostly out of PLA. Additionally, it is worth mentioning that the arm is powered by an external Power Supply Unit (PSU) that delivers 11.1 Volts, reducing the possibility of electric hazard. Finally, in order to eliminate the risk of causing harm to the user from the movement of the robot, we have reduced the movement speed of the motors to 30% of the attainable speed, resulting in gentle item delivery to the user.

The final measure that we have to take into account in order to prevent an undesirable situation is to harden the system and reduce the surface of vulnerability. To achieve this, we have used secure protocols, such as Hypertext Transfer Protocol Secure (HTTPS) and Domain Name System Security Extensions (DNSSEC). The HTTPS protocol ensures that all communication traffic between the controller and Google APIs is encrypted. Combined with DNSSEC that addresses known flaws of the DNS (Domain Name System) protocol, we have eliminated the possibility of Man In The Middle attack (MITM). This minimizes the risk of a malicious attacker remotely taking control of the robot and causing harm to the user.

6. Discussion

It is evident that in complex systems, such as the one proposed in this paper, large amounts of computational power are needed. Traditionally, in order to meet these computational needs, engineers were required to equip their systems with high-performance CPUs and GPUs [21]. This undoubtedly increases the implementation and maintenance cost of the system. In recent years, as cloud technologies are emerging, more and more researchers are moving to IoT solutions that use the power of the cloud. The core idea of a cloud enabled system is that all processing occurs in the cloud, as opposed to the traditional on-site approach. The usage of cloud for processing does have several drawbacks, including greater architectural complexity and the cloud server expenses. However, the advantages that it offers outweigh its disadvantages [22]. First, it offers decreased on-site implementation cost, since the end devices are not required to house expensive high-performance hardware. This advantage becomes more evident at higher scales (for example, one cloud server could serve tens or hundreds of robots for a fraction of the traditional cost). For our system, we calculated that a single server equipped with an AMD Ryzen™ Threadripper™ 2950× processor could serve more than 300 robotic arms. The server, however, has to be located close to the region that the majority of its users are located, in order to minimize the communication latency and offer a great overall experience to the user. Given that the frame resolution in which the YOLO is operated is well defined, we also calculated that a 1 Gigabit connection would be more than enough for serving that number of robotic arms. Second, the maintenance cost is significantly decreased, because of the single cloud processing unit; a change on the software code needs to be made once. Finally, in cases where remote monitoring is deemed necessary, cloud offers the ability to remotely collect data and detect emergency situations that would otherwise be catastrophic.

One other technique that can be used alongside cloud computing is the usage of Field Programmable Gate Array (FPGA) boards [23], such as PYNQ-Z2. In recent years, the demand and usage of FPGAs is increasing, because of the distinct advantages they offer. Their main advantages are the increased computation speed and low power consumption, compared to general purpose CPUs and GPUs. These advantages, of course, are a byproduct of the custom circuit that performs the computation task. The design of this circuit, however, is a very time-consuming process, and it is not always possible. The system designers are required to evaluate the situation and determine whether or not the implementation of FPGAs would benefit the system as a whole. For the proposed system of this paper, the implementation of FPGA on the cloud server could enhance the efficiency of the heavy computation tasks of the robots (for example machine vision) and reduce the power consumption.

7. Experimental Results

To evaluate the robotic arm prototype, we performed several experiments using different objects. Unfortunately, due to the spread of the SARS-CoV-2 disease, conducting experiments on actual patients suffering from movement impairments proved to be a difficult task. Therefore, we limited our evaluation to PhD candidates and staff from our University. The experiments were conducted at the Laboratory of Robotics Embedded and Integrated Systems at the University of Western Macedonia. A special area was set up to meet the requirements of the workspace, lighting, and background color of the objects. Objects capable of being detected by the available recognition functions were placed in the surrounding area of the arm. The set of objects placed were 11 of which 5 are recognized by the algorithm YOLO algorithm, 4 by the shape and color recognition function and 2 by the QR code recognition. Figure 5 shows the area that the arm and the objects in the area has been placed. Some objects are identified by their shape, some are identified by their color and some are identified by the QR code that is located on their upper side.

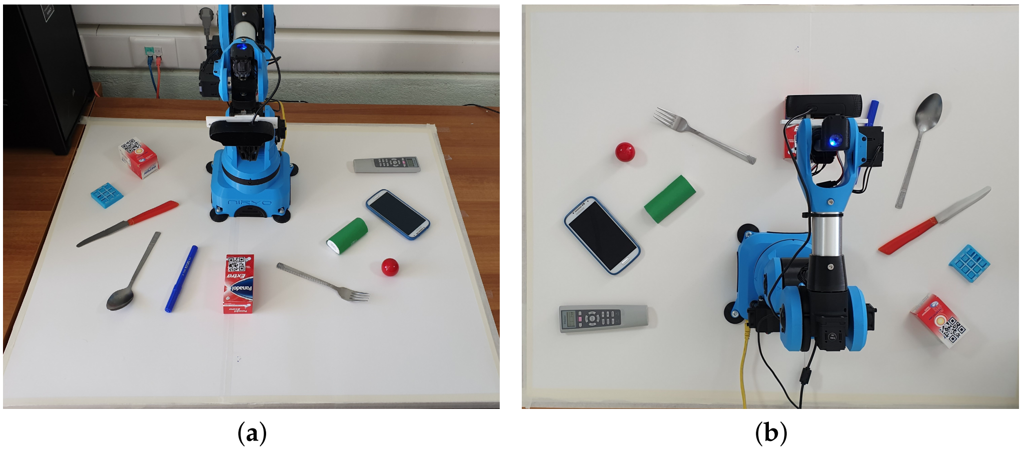


Figure 5. Images (**a**,**b**) present the initial state of the robot and the objects used for the experiments, from different angles.

In this configuration, for demonstrating purposes, we asked the robot to give us the item with label "Depon", identified by the QR code. Figure 6 presents the detection and identification of nearby objects, as the robot was trying to find the requested item.

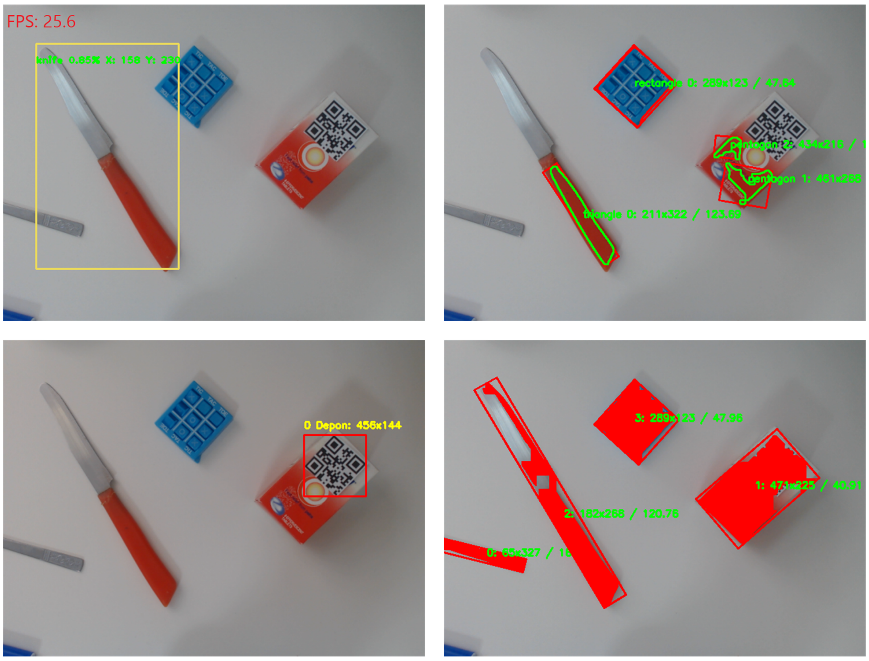


Figure 6. The robotic arm is identifying nearby objects including a knife, a blue rectangle and a triangle. The algorithm successfully identified the requested object, called "Depon".

After the robotic arm has identified the location of the correct object, it proceeds to pick it up (Figure 7) and deliver (Figure 8) it to the user.

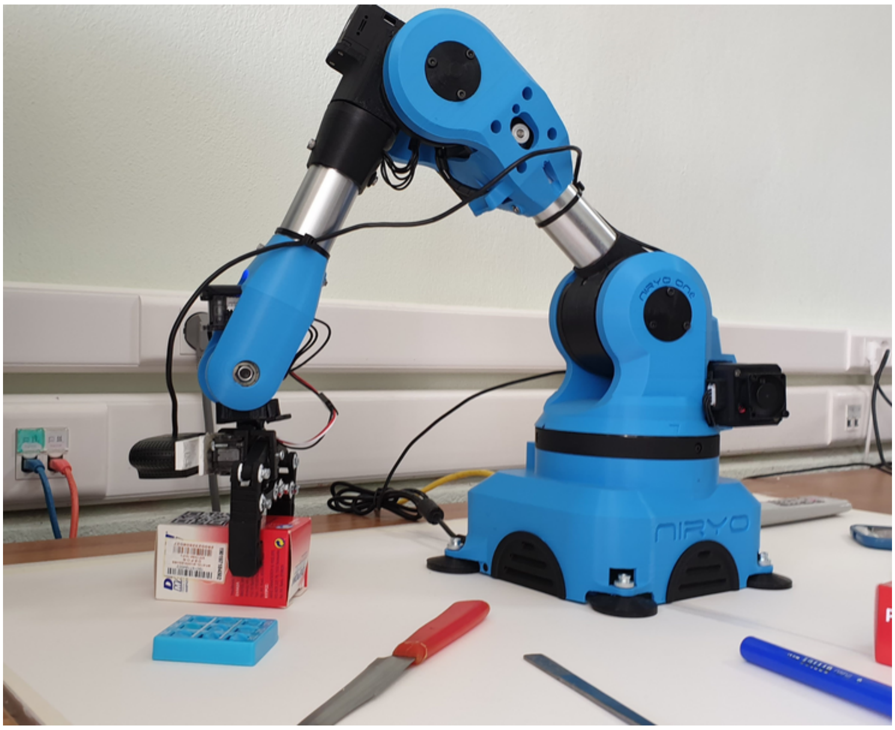


Figure 7. Niryo-One robot catching the correct object.

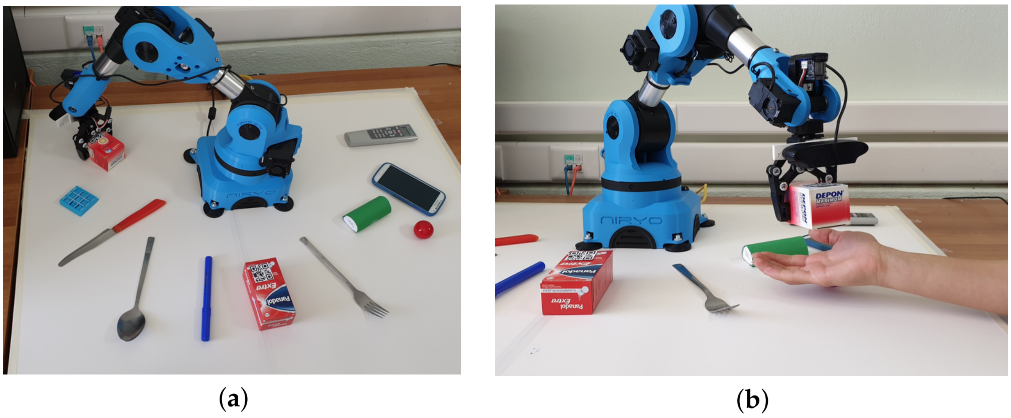


Figure 8. The robotic arm can accurately locate and pick up the requested object firmly (**a**), in order to deliver it to the user's palm (**b**).

To determine the accuracy, object drop rate and execution time of the system, we conducted 3 different experiments with 30 repetitions for each experiment. In the first experiment, we asked the robot to deliver an object using the YOLO algorithm, in the second experiment we asked the robot to deliver an object based on its shape and color, and on the third experiment we asked the robot to deliver a custom object using the QR code technique. There were 11 different objects that were used; the robot had to find the correct object and deliver it to the user. For each repetition, all the items were shuffled and had different positions. Table 1 presents the results of the experiments. It is worth mentioning that the voice recognition accuracy rate was directly affected by the pronunciation and the accent of the user. Easier to pronounce and common words are easier for the system to recognize successfully. Additionally, regarding the dropped object failure rate, we noticed that the texture and the shape of the object affected the chance of the robot to drop it. Glossy plastic items were had the highest drop rate during the experiments.

Table 1. Experimental results of the system for 3 experiments, using 30 repetitions for each experiment. The system managed to achieve an average accuracy rate of 92.33%, and a best accuracy of 96.66% when requesting objects using the QR code technique.

	YOLO	**Shape & Color**	**QR Code Extension**
Accuracy rate	90%	90%	96.66%
Voice recognition rate	90%	93.33%	96.66%
Drop item failure rate	3.33%	6.66%	3.33%
Average total execution time	52 s	49 s	55 s

8. Conclusions

This research presents a low-cost voice-controlled robotic arm, capable of detecting and delivering objects to the user. Our project aims to assist people with physical movement impairments, improving their quality of life. The robotic arm assistant is powered by machine vision and voice recognition software. In order to evaluate the efficiency and usefulness of the proposed system, we have conducted several experiments, both in our laboratory and on actual patients with physical disabilities. Experimental results indicate that the robotic arm is capable of performing the required task in a timely manner.

In the future, we plan to further enhance the system, improve its efficiency and add more features. One possible extension is to address the problem of insufficient illumination of the environment, which inevitably leads to an inability to identify objects. The addition of an LED (Light Emitted Diode) light source to the arm could solve this problem. A further extension could involve the incorporation of a magnetic switch or an IR (Infrared) sensor in the arm's final end-effector to inform the system if the claw failed to grasp the object or malfunctioned. With this feature, the system would have the ability to repeat the gripping process. Another interesting implementation would be the replacement of the USB camera with a stereoscopic camera or the addition of an ultrasonic sensor in order to detect the height of the objects and, by extension, to calculate the appropriate distance of positioning of the claw from the plane during the gripping process. Moreover, we aim to develop an intuitive graphical interface to visualize the various options and results. Finally, in the future we plan to further analyze the failure mode and effects. This will help to reduce the errors during operation and therefore improve the quality of the robotic system.

Author Contributions: Conceptualization, G.N. and M.D.; Supervision, M.D.; Software, G.N.; Writing—Review and Editing, G.N., N.B. and M.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. United Nations Conference on Trade and Development. *Impact of Rapid Technological Change on Sustainable Development*; OCLC: 1145601349; United Nations: San Francisco, CA, USA, 2020.
2. Garcia, E.; Jimenez, M.; De Santos, P.; Armada, M. The evolution of robotics research. *IEEE Robot. Autom. Mag.* **2007**, *14*, 90–103. [CrossRef]
3. Coito, T.; Firme, B.; Martins, M.S.E.; Vieira, S.M.; Figueiredo, J.; Sousa, J.M.C. Intelligent Sensors for Real-Time Decision-Making. *Automation* **2021**, *2*, 62–82. [CrossRef]
4. Andreu-Perez, J.; Deligianni, F.; Ravi, D.; Yang, G.Z. Artificial Intelligence and Robotics. *arXiv* **2018**, arXiv:1803.10813.
5. Zhao, Q.; Tu, D.; Xu, S.; Shao, H.; Meng, Q. Natural human-robot interaction for elderly and disabled healthcare application. In Proceedings of the 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Belfast, UK, 2–5 November 2014; pp. 39–44. [CrossRef]
6. International Federation of Robotics. 2020. Available online: https://ifr.org/news/service-robots-record-sales-worldwide-up-32 (accessed on 30 October 2021).
7. Kobayashi, Y.; Gyoda, M.; Tabata, T.; Kuno, Y.; Yamazaki, K.; Shibuya, M.; Seki, Y. Assisted-care robot dealing with multiple requests in multi-party settings. In *Proceedings of the 6th International Conference on Human-Robot Interaction—HRI'11*; ACM Press: Lausanne, Switzerland, 2011; p. 167. [CrossRef]
8. Goodrich, M.A.; Schultz, A.C. Human-Robot Interaction: A Survey. *Found. Trends® Hum.-Comput. Interact.* **2007**, *1*, 203–275. [CrossRef]
9. Zhao, X.; Wu, C.; Liu, D. Comparative Analysis of the Life-Cycle Cost of Robot Substitution: A Case of Automobile Welding Production in China. *Symmetry* **2021**, *13*, 226. [CrossRef]
10. Dahlin, E. Are Robots Stealing Our Jobs? *Socius* **2019**, *5*, 2378023119846249. [CrossRef]
11. Ojstersek, R.; Veber, M. Intrinsic Motivation of Students Learning Robotics in Mechatronics Education. Available online: https://www.researchgate.net/publication/320015965_Intrinsic_Motivation_of_Students_Learning_Robotics_in_Mechatronics_Education (accessed on 30 October 2021).
12. Papanastasiou, S.; Kousi, N.; Karagiannis, P.; Gkournelos, C.; Papavasileiou, A.; Dimoulas, K.; Baris, K.; Koukas, S.; Michalos, G.; Makris, S. Towards seamless human robot collaboration: Integrating multimodal interaction. *Int. J. Adv. Manuf. Technol.* **2019**, *105*, 3881–3897. [CrossRef]
13. Tsarouchi, P.; Matthaiakis, S.A.; Michalos, G.; Makris, S.; Chryssolouris, G. A method for detection of randomly placed objects for robotic handling. *CIRP J. Manuf. Sci. Technol.* **2016**, *14*, 20–27. [CrossRef]
14. Pineau, J.; Montemerlo, M.; Pollack, M.; Roy, N.; Thrun, S. Towards robotic assistants in nursing homes: Challenges and results. *Robot. Auton. Syst.* **2003**, *42*, 271–281. [CrossRef]
15. Nandan, N.; Thippeswamy, K. A Tensorflow Based Robotic Arm. In Proceedings of the 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Msyuru, India, 14–15 December 2018; pp. 326–330. [CrossRef]
16. Pulikottil, T.B.; Caimmi, M.; Dangelo, M.G.; Biffi, E.; Pellegrinelli, S.; Tosatti, L.M. A Voice Control System for Assistive Robotic Arms: Preliminary Usability Tests on Patients. In Proceedings of the 2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob), Enschede, The Netherlands, 26–29 August 2018; pp. 167–172. [CrossRef]

17. Bandara, H.; Edirisighe, M.; Balasooriya, B.; Jayasekara, A. Development of an interactive service robot arm for object manipulation. In Proceedings of the 2017 IEEE International Conference on Industrial and Information Systems (ICIIS), Peradeniya, Sri Lanka, 15–16 December 2017; pp. 1–6. [CrossRef]
18. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
19. Zhao, Z.Q.; Zheng, P.; Xu, S.T.; Wu, X. Object Detection with Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
20. Speech-to-Text: Automatic Speech Recognition. December 2020. Available online: https://cloud.google.com/speech-to-text (accessed on 30 October 2021).
21. Brůha, P.; Mouček, R.; Šnejdar, P.; Bohmann, D.; Kraft, V.; Rehor, P. Exercise and Wellness Health Strategy Framework - Software Prototype for Rapid Collection and Storage of Heterogeneous Health Related Data. In *Proceedings of the 10th International Joint Conference on Biomedical Engineering Systems and Technologies—HEALTHINF, (BIOSTEC 2017), INSTICC*; SciTePress: Setúbal, Portugal, 2017; pp. 477–483. [CrossRef]
22. Avram, M. Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective. *Procedia Technol.* **2014**, *12*, 529–534. [CrossRef]
23. Chen, F.; Shan, Y.; Zhang, Y.; Wang, Y.; Franke, H.; Chang, X.; Wang, K. Enabling FPGAs in the Cloud. In *Proceedings of the 11th ACM Conference on Computing Frontiers*; CF’14; Association for Computing Machinery: New York, NY, USA, 2014. [CrossRef]