International Journal of Applied Mathematics, Electronics and Computers 11(2): 062-071, 2023

IJAMEC

e-ISSN: 2147-8228

INTERNATIONAL JOURNAL OF APPLIED MATHEMATICS ELECTRONICS AND COMPUTERS

www.dergipark.org.tr/ijamec

International Open Access

Volume 11
Issue 02

June, 2023

Research Article

The Natural and Physical Effects on the Mobile Robot Designed to Recognize and Collect Objects

Erdem ABAT [a], ***Metin TURAN*** [a]

[a] *Department of Computer Engineering, Istanbul Ticaret University, Istanbul, Turkey*

ARTICLE INFO

Article history:
Received 22 May 2023
Accepted 1 June 2023

Keywords:
Arduino
Artificial intelligence
Image processing
Mobile robot
Raspberry pi

ABSTRACT

Today, studies are carried out in the field of robotics, such as the discovery of old or unknown places, their targeting, and object recognition. The designed robots can be dressed as cars, drones, people or arms for use. This common goal is to reduce humans work and to provide robots do everything what humans can do. These processes include hardware and software tasks. Artificial intelligence offers us the ability to teach and apply intelligence to the machines with the help of software. The aim of this study is to design a prototype mobile robot that can detect and recognize fruits that are far from the current location, but reachable by the robot, and can collect the fruits and transport them from one place to another. A prototype mobile robot was developed to obtain which conditions effects the robot purpose, when the environment was not planned for specific conditions. Experiments on the natural environment showed us that if sufficient light and angle were available then an average of 95% success rate in the correct recognition of objects and an average of 90% in detection achieved by the prototype robot. Moreover, the motor power and the arm handle area have very important role in the processes of the robot's hardware grasping and carrying the fruits. In addition, based on the observations obtained result of the experiments, ambient light is quite effective in capturing and detecting the object. The condition emerged result of an obstacle on the path was not considered in these experiments.

This is an open access article under the CC BY-SA 4.0 license.
(https://creativecommons.org/licenses/by-sa/4.0/)

1. Introduction

People have always improved their living standards by using the power of discovery and learning, as well as many features given by their existence. For example, even moving an object, which we do naturally, is a very complex and difficult process in its essence. If we simulate this process not as a reflex action by ourselves, then we can realize how long and difficult this job, which is actually seen as simple. Today, this process has been automated using different tools and methods for situations where it exceeds manpower.

By the simple example given above, we can see that humans being replaced by robots in most areas. In addition to this robotic process, studies in the field of artificial intelligence have accelerated in order to give to the robot the ability to think [2].

The contribution of this research is to coordination of visuality and mobility and find the main natural and physical effects to achievement of the goal. In this research, it is aimed to automate the action of recognizing an object that we intend and transport it from the current place to a target location by means of a robot in general. The natural effects on the camera view and the effects on physical movements were also evaluated and noted. Today, object recognition studies (computer vision) is a common field of this purpose. Physical separation of objects (classification) is also a method used in different application areas that follow this process. With this limitations, the study aims to be a model that exemplifies the integration of processes, both software and hardware, as a small prototype. It includes artificial intelligence technology besides robotic coding. Small fruits (light fruits) were chosen as target objects to transport within the navigation area of the robot in the home environment. Since a prototype mobile robot is designed, its physical capacity is limited by electricity and motor powers.

* Corresponding author. E-mail address: erdem.abat@istanbulticaret.edu.tr
DOI: 10.18100/ijamec.1300406

2. Literature Review

There are many important applications in the field of robotic classification. In recent years, it has been observed that studies have been carried out for different problems, especially in line with the easy availability of robotic parts and the reduction of costs. The researchers stated the hardware design difficulties less and the camera and other hardware usage disabilities more. Although there were some similar findings also seen on this research, more details about the hardware design were presented to the user at the end of findings section additionally. One of the important studies was carried out by Wührl et al. in 2021 [1]. They have developed a sorting robot that prepares mostly insect objects for barcoding individually from bulk samples. The robot detects, displays, measures individual samples, and then moves them into the wells of a 96-well microplate. The robot is designed for using parts that can be printed on a commercial (Fused Deposition Modeling—FDM) 3D printer. The robot is equipped with customized lenses, led lighting and image recognition software, and two high resolution cameras (overview and sample camera). In addition, a sample handling system is integrated that uses a suction pump to transfer insects into the wells of a standard 96-well microplate. The robot divided the detected insects into 14 different classes and classified all other insects with the “other” label. The best classification results were obtained for “Hymenoptera Diapriidae” and “Hemiptera Cicadellidae”, where all insects were correctly classified, while insects in the “Hymenoptera Ichneumonidae” class appeared to have the lowest correct classification rate (75%) [1].

In the study by Lary et al. [2], an autonomous robot was designed that can quickly learn the properties of environments it has never seen before. All data were collected automatically in a coordinated manner using an autonomous robotic team. In the study, Maritime Robotics Otter autonomous boat and Freefly Alta-X autonomous professional quad helicopter were used. Open source QGroundControl software was used to control autonomous operations. All robotics crew members carried a highly accurate GPS and INS so that every data point could be geolocated and timestamped. For each location sampled by the robotic boat, a VNIR remotely sensed spectrum provided by the hyper-spectral data cubes collected by the aircraft was correlated. The Machine Learning (ML) methods used include hyperparameter optimized shallow neural networks, hyperparameter optimized decision trees ensembles, hyperparameter optimized gaussian process regression, and a superlearner that includes all the aforementioned approaches. In Mode 1, the robotic boat autonomously measures the ground accuracy with multiple parameters while using sensors, while the robotic aircraft collects remotely sensed observations of exactly the same locations using hyper-spectral and thermal imaging. In Mode 2, it is aimed to perform a fine-grained multi-class surface classification of the entire domain. This is done by providing remotely sensed data to an unsupervised classification. It is seen that a consistent result is obtained between ML prediction and real on-site boat observations. They obtained thousands of training data points in just a few minutes. Thus, this training data enables ML methods to learn quickly with examples and to obtain large-area maps of the composition of the environment [2].

Rahman et al. [3] attempted to identify important questions regarding the safety and reliability of learning-based sensing systems and to summarize various approaches. The first trend includes techniques that use examples of past failures or estimate the quality of the output based on the similarity of the context or site of operation with previous experience. The second trend involves methods that detect inconsistencies in the perception output, either through a stream of input data or inputs from different sensors or outputs from different models. The third trend is based on the confidence learning and the uncertainty estimation, where perception modules express their own confidence in their output. In general, performance monitoring using failure samples depends on an auxiliary network to predict the failure of the underlying network. The base network can be responsible for any specific task; image classification, segmentation or object detection. The auxiliary network is trained using both positive and negative samples where the base network performs its specific task with the expected accuracy. During the deployment phase, the auxiliary network works with the base network and predicts the success or failure of the base network to perform a particular task. They can be categorized by whether they perform monitoring through input validation or output evaluation, internal activation auditing, or a combination of these [3].

Katija and his colleagues [4] presented a new ML Integrated Tracking algorithm based on a class of algorithms known as detection and tracking for underwater vehicle control to explore deep regions in the ocean. ROV MiniROV is a flight vehicle that equipped with a main camera. A Tensorbook (Lambda Labs) laptop was used on the board to receive stereo video data, run models and 3D monitoring software, and issue control commands to the vehicle. Detection of potential target classes was performed simultaneously on images taken from both left and right stereo cameras. As software, Lightweight Communication and Sequencing (LCM) was used to communicate between the various modules. The training data for the multiclass detector came from two separate sources. Color images used for training corresponded to representative mid-water animals commonly observed in the upper water column of Monterey Bay, which includes a subset from FathomNet that an underwater image training set. The measurement vector consists of 8 elements and the state consists of 5 elements. A separate UKF, used to minimize measuring and modeling errors for geometric and thermal errors, was located on each runway, as multi-object monitoring was required for the mid-water monitoring mission to be successful. To associate the bounding box pairs with the tracks, they converted each measurement into a state space by fitting a 3D box location to four 3D points corresponding to a stereo box pair. They then assumed multivariate Gaussian distributed noise in the state space and calculated the square of the distance of Mahalanobis measurement using the current state mean and covariance of the path. After about 50 hours of recorded footage, they determined that the viewing scenarios could be generalized into five different functional categories. From the data obtained, it was seen that the detections made under water could be successful and if the quality of the equipment increased, then the performance and results would also improve [4].

In the study by Hiçdurmaz et al. [5], they aimed to adapt moving without hitting a place which is succeeded by the people in daily life simply eye tracking, to robots. In this direction, with a camera that detects and processes real-time images placed on the robot, it is possible to see and control its surroundings with the help of sensors, and by providing distance control with sensors, the robot can move forward without crashing. While performing this process, the controls of the robot in the desired direction were provided by taking many information of the environment such as light, color and texture in the images taken as a basis. It were observed that problems called noise occur in the images due to many reasons such as the camera and the light in the environment while obtaining these images. These problems were solved by the segmentation and the threshold techniques applied during image processing. In this study, a statistical method was applied to the images to determine the floor areas. Thanks to a large number of filtering methods, it was possible to detect the corners in the images. In the proposed approach, quicker and easier ground control is achieved without using filters. As a result of the observations obtained in the study, although the ground and objects could be distinguished, it was a little more difficult to distinguish objects similar to the ground. Experimental results showed that the proposed algorithm achieves 92% accuracy, which was successful in separating the ground from other regions, making it a highly flexible method for obstacle avoidance. Therefore, it can be concluded that the proposed method provides successful results at low cost without the need for additional filters used in image processing [5].

Addressing developing and current speech recognition approaches in surgical robots, Ruby et al. [6] discussed existing ML methods that were noteworthy in the studies. Automatic Speech Recognition (ASR) used as a method is a strategy of transitioning from the acoustic data stream of speech to a word collection. Robotic medical surgical procedure is a training to accept medical procedure with methods for small instruments attached to a mechanical arm. The specialist controls the mechanical arm with a computer. The new research expressed in this study was to control the robotic arm through speech processing, which was unlike computer programming at all. The results clearly demonstrated the incredibleness of ML adapting ASR in medical surgical procedures. According to the statement, ML and traditional ASR techniques yielded significantly better accuracy gradually. ML methods such as Hidden Markov Models yielded higher word rate error compared to Dynamic Time Warp and Conditional Random Fields. However, ML-based speech recognition provided impressive word precision [6].

In the study aimed at cutting fruits with the help of a garden shear designed by Kahya et al. [7], first of all, the space coordinate axes of the fruits were determined. Image processing technique was applied to find the coordinate axes. A 2D camera model was used for image processing. Ultrasonic sensor was used for the distance (z) which was the third coordinate axis. C# programming language was used for the use of this sensor and the robot was prevented from reaching a certain distance. As a result of the experiments, the wrong value of the robot arm was determined as 15% for apples. As a result, it was understood that the structure used in the system could be developed a little more. With this research, it was observed that the most important factor in robot design was the gripper that provides the last movement. As a result of the study, it was expressed that the pneumatic cutting was the factor that affected the success of the system, beside all the parts and software of the robotic system [7].

In the Electroencephalogram (EEG) study by Karakoç et al. [8], which helps to investigate human cognitive activities, the method of measuring the electrical impulses of neurons was used. In the study, a single-channel Neurosky Company's EEG biosensor, which could detect the user's mental fatigue, brain waves and blinks, was used. The sensor, which touched the contact and reference points on the forehead and ear, transmitted all measured data to digital form software and applications. In this practical work for prototyping, the robot arm was controlled by using brain signals. The study first was created by a model close to the human hand and designed with a 3D printer, then the motors were installed to control this model, the motion information of the motors was transmitted to the Arduino board via Bluetooth and the control of the motors was made with the EEG sensor. As a result of this application, EEG sensor could open and close the model's fingers through the brain wave sensor [8].

Aivaliotis et al. [9] described the robotic manipulation of complex parts using a visual recognition service as an external tool to predict the position and/or orientation of a part after it had been grasped. The main ML service used for the implementation of visual recognition were Convolutional Neural Networks (CNN). First, in the training phase, a set of images was used to create a class with the same properties, labeled by the user according to their common properties. Later, an unlabeled image could be analyzed using the classifier. Based on ConvNets' confidence, a score was calculated showing the rate at which the unlabeled image matched each class. Finally, assuming the service was correct, the class with the highest score is the class that represents the correct label of the input image. This particular application of ML and visual recognition leveraged IBM's Watson Visual Recognition service. The class label with the highest score was the final result of the classification. An industrial pilot robotic manipulation case was set up to demonstrate the specific approach. A robotic arm that equipped with a parallel gripper was used to grip the curved handle of the shaver. An led light source was attached to the side of the camera in an angled position to provide stronger, uniform illumination of the complex part. The entire procedure of manipulation of the complex part was controlled by a host with the role of a communication node between devices/services. TCP/IP is the communication protocol used to implement this method. In the first set of experiments, the number of classes and the number of images of each class were fixed while the resolution of the visual system changed. After the handle was grasped, a photograph was taken and uploaded for classification. Experiments were 60 classes and 30 pictures per class. Assuming that only this construct was considered, the class 4 gave the highest response score in all cases. In the second set of experiments, the procedure was repeated for each experiment. Experiments were carried out with 60 classes and 90% image resolution. Likewise, assuming that only these classes were considered, it was seen that the number of images in each class mainly affects the Service Response Time, while the Response Scores were very close to all experiments [9].

Gopalapillai et al. [10] presented the analysis of robotic data using ML techniques when the data had multiple views of the environment. The study used dynamic time warp distance, which gave a measure of similarity of time series data. They reported clustering accuracies in the range of 82% to 100%. The study used data from Irvine's ML repository. In this study, full link clustering algorithm was applied and the results obtained with this method vary between 88% and 97%. Study used an experimental setup with a robot equipped with 4 sensors that follows a straight line path to gather information about the environment. The obtained accuracy was in the range of 73% to 98%. Artificial neural networks was widely used to solve data analysis tasks. The robot was equipped with infrared proximity sensors and thermal imaging sensors. The robot was placed in one corner and moved to opposite corners, while onboard sensors collected data about the environment. Seven different scenarios were created based on object placement to simulate the robotic environment for experimentation. Since this collected data were noisy, the collected data were clipped to eliminate unwanted readings. In this study, Information Gain (IG) based feature selection technique was applied to reduce the number of features and to remove features that were not important for classification. Classifiers chosen to classify data according to relevant scenarios, based on performance and popularity were Naive Bayes (NB), Logistic Regression (LR), Multilayer Perceptron (MLP), Adaboost and Support Vector Machine (SVM). Weka, an open source toolkit for solving real-world data mining problems, was used for classifying time series data. The ultimate goal was to divide the data into 7 classes, which were 7 scenarios. The Adaboost classifier showed the highest accuracy (99.03%) for the 4IR+2Thermal dataset. On the other hand, SVM, MLP, LR and NB resulted in 98.07%, 98.07%, 98.07% and 83.65% accuracies respectively on the same dataset [10].

Fard et al. [11] applied ML methods to evaluate automatically the surgeon's performance in Robotic-assisted Minimally Invasive Surgery (RMIS). In the study, they used the theory of kinematic analysis. The extracted features were used to measure the movement pattern of surgeons with different dexterity levels. Specifically, they compared two commonly used ML methods, LR and SVM. Eight right-handed surgeons of varying skill levels sutured approximately 5 times. They analyzed the kinematic data captured at 30 Hz using the da Vinci robot's API to extract the features. Surgeons were divided into two categories, specialists and novices, according to their scores. Before calculating the features, the raw data were filtered using a local regression-weighted linear least squares method that reduced noise in the signal data and retained the detail of the model. Finally, a total of 17 features were derived from each trajectory. They then used Principle Component Analysis (PCA), a dimensionality reduction technique based on an orthogonal transformation, to reduce the set of possible related features to a smaller set of unrelated features that were linear combinations of the original features. To calculate the performance of the system, classifier validation was performed using two model validation schemes. The first is a super exclusion (LOSO), in which one trial for each surgeon was excluded for testing. Second, it excluded all attempts of a surgeon (LOUO). The first validation method evaluated the robustness of a method for repeating a task by excluding one trial for all subjects, while the second setup evaluated the robustness of a method when a subject (i.e. surgeon) was not previously seen in the training data. It showed that the best overall accuracy achieved was 85.7% on LOSO and 71.9% on LOUO. The results also showed that logistic regression for LOSO and SVM for LOUO model validation scheme for LOSO provided the best classification performance. For almost all experiments, the best overall accuracy achieved was obtained from logistic regression for the LOSO scheme, while SVM gave the best results for LOUO [11].

Batur and his colleagues [12] discussed the flexible manufacturing cells consisting of CNC machines. Robotic cells can process different types of parts, which generally have different processing times. The problem addressed is to find the robot motion and part sequence that minimizes the completion time of a given production set. Simulated Annealing Algorithm was chosen in the study. The vector they determined was consisted of three parts, where the first part gave the order of the parts to be machined, the second part showed the relevant machines where each part would be machined, and the last part was about allocated part number and processing time. According to the order, assignment and placements defined by this notation; all parts were taken from the input buffer, transferred to the relevant machines and delivered to the output buffer after their processing is completed. The first solution for this problem was randomly generated and the process started. After the parts to be machined in the first part were sorted, the second part of the solution was created as either the first or the second machines, and finally one of the n parts was defined by the allocated time ratio. The four sub-strategies used in this study were part replacement, assignment change and change of allocated part or rate. Experimental results showed that it was possible to reach acceptable solutions very quickly and effectively [12].

In the study conducted by Kahya et al., researches were done on how active robots could be in the field of agriculture [13] and what would be their contributions to the human. In this direction, the kiwi tree was determined as the target. Experiments were carried out with branches that were plucked from the kiwi tree placed in front of the created robotic system and tree had fruits on it. These experiments were analyzed in two separate parts. First of all, the location of the fruits was determined, and then the movement of the robot arm and the cutting process were taken into account. In the process of determining the location of the fruit, the image processing technique was used to find coordinates, since the robot had to perform operations based on the coordinates of the fruits. 2D camera and ultrasonic sensor were used to reach this coordinate information. They used the C# language for programming. After the calculations made according to the coordinate coming from the camera to the processor on the robotic control card, the movement of the robot arm was provided. Inverse and straight kinematic calculations for the robotic arm were checked in the Matlab program, and the control of the motion shape was tested. In addition to the horizontal axis (x) and vertical axis (y) coordinate information, an ultrasonic sensor was used for the distance (z), which was the third coordinate axis. The success rate was 72.48% for 109 fruits. At the end of the trials, the robot arm error value for fruit declared as 27.52% [13].

Dr. Lukka et al. [14] discussed the ZenRobotics Recycler (ZRR), a system that sorts construction and demolition waste by collecting valuable objects from a conveyor belt using robotic hands. The ZRR is a system consisting of a set of sensors, a control system, and industrial robots. The sensors and control system control industrial robots to take selected materials from the waste stream on a conveyor into multiple chutes. ZRR uses ML for object and material recognition and object processing. In addition, various sensors sensitive from visual wavelengths to near infrared and a metal detector are used to determine the materials of detected objects. In addition to material recognition, high-resolution RGB cameras also offer a visual view of the waste stream for clarification. Given the identified objects and materials, the system optimizes a collection sequence that maximizes the monetary value of the recovered objects. A concept called handle was used to decide where to hold it. Adaptive algorithms were used to maximize the picking success for each object and optimize the handles. The software language used in this study is Clojure. Deployment of the first ZRR has proven that robotic classification of construction and demolition waste is possible [14].

Perkowski et al. [15] shared their research and experience in designing cheap natural-sized humanoid cartoons and realistic robot heads, agents for advertising, education and entertainment. Inexpensive servos, plastic, playwood and aluminum from Hitec and Futaba were used for the design. Image processing and pattern recognition from the Face Maria variants use software developed at PSU, CMU and Intel. They compared various commercial speech systems from Microsoft, Sensory, and Fonix. Andrea Electronics' microphone array was also included. The software is in Visual C++, Visual Basic, Lisp and Prolog. The goal is to integrate verbal and nonverbal robot behaviors in a uniform way. The system generates the robot's behaviors (C program codes) from the examples given by the user. A comprehensive ML/Data Mining methodology was used, based on constructive induction. Specifically, the hierarchical decomposition of decision tables of binary and multi-valued functions and relationships into simpler tables, until tables of trivial functions with direct counterparts in the behavior components were found. A unified internal language was used to describe behaviors in which text rendering and facial gestures were combined. Input sentences were coded in multi-valued logic and output sentences were produced as a result of logic-synthesis based generalization. This method automatically generalizes responses in case of insufficient information. The more degrees of freedom in the results obtained, the more animation realism, the synchronization of spoken text and head (especially jaw) movements were very important. Eyes, jaws, and other moving head components should be an exaggerated rather than natural size; similarly, head gestures and speech intonation should be slightly exaggerated to achieve better interaction. The sound should be praised, including sounds from engines and gears, and for better theatrical effect. The noise of the servos could also be reduced with proper animation and synchronization, the best available ASR and text-to-speech packages should be applied, especially those using word staining, the designer should look into new material and learn in many areas from puppet [15].

3. Materials and methods

In the study, the sizes of objects that the robot can recognize by image processing method are limited to small diameter fruits. The main reason for this is that the gripping arm motor is not powerful enough to grasp life-size objects. Therefore, a realistic simulation was tried to be performed by producing light samples of real fruits that are the same in shape. The feature extraction ML model will eventually be usable within the same software production model as it performs the object recognition action.

The hardware materials used in the design of the robot are roughly listed below;

2 pieces plexi plates
Robotic arm
4 pieces wheels
4 pieces engine
4 pieces servo engine
Arduino uno board
L298N dual motor driver board with voltage regulator
Servo driver shield
Jumper cables
Raspberry pi 3 model b+ combo kit
7.4 V 2S lipo battery 2800 mAh 35C
Ultrasonic sensor mounting apparatus type C
HC-SR04 ultrasonic distance sensor
HC06 bluetooth-serial modular board
A3 compact lipo (2-3S) charger - balancer

3.1. Body design of the robot

The plexi plates were numbered to be unified them properly and the bottom body design first was completed with servo motors which will control the wheels as seen in Figure 1. Since the main elements had to be positioned between the bodies, the placement of the engines on the first plate had to be completed first. After soldering the opposite polarity poles of the motors, they were fixed to the motor sockets and tested by mounting 4 wheels. However, it was tested considering that the wheels were not fully seated and due to the very hard plastic structure of the tires, even if the motors rotate, there may be vibration in the wheels. By connecting the + - poles of the lipo battery to the wires coming from the motors, the motors have been observed to spin the wheels properly. When the tests were completed, after disassembling the parts again, the rims of the wheels, engine stabilizers and body plexiglass plates were painted black and blue with the help of spray paints.

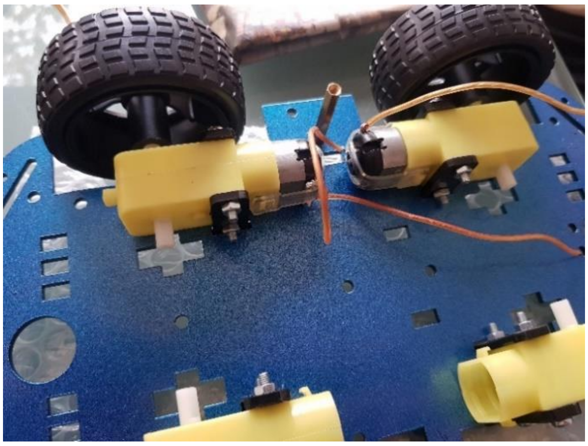

Figure 1. Bottom body design of the robot

3.2. Robotic arm design

After each of the plexiglass forming the arm were numbered, the assembly process was completed step by step, as can be seen in Figure 2, with servo motors. In this process, the stage of disassembling the gelatin of the parts and adjusting the angles of the servo motors during assembly was very challenging stage.

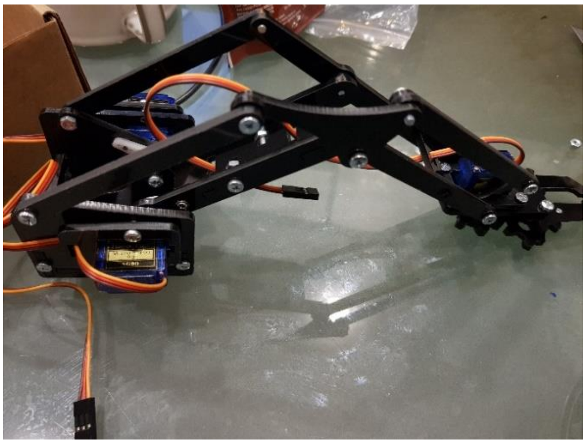

Figure 2. Robotic arm design

Then the cables from the DC motors were connected to the motor driver and the necessary power connections were made with the Arduino. By connecting the Arduino circuit to the computer via a USB cable, the necessary software was coded and uploaded. After the mobile application was installed, the controls were provided by providing the connection. In the first trials, forward and right movements were working, but left and back movements could not be made. It was seen that the problem was caused by the incorrect connection of the cables, so the connection was corrected and the problem was prevented.

3.3. Arduino uno board

Figure 3. Arduino and Raspberry Pi 3 circuit setup

As seen in Figure 3, 4 pins providing direction are connected to the 4, 7, 12 and 13 inputs in the Arduino circuit via 4 male-female cables. Then, 3 female end cables belonging to 4 servo motors are connected to the servo driver circuit board, which allows the use of multiple servo motors integrated into the Arduino circuit. Then connected to Raspberry Pi 3 with Arduino USB cable. So that, they could contact each other by using signals anymore.

3.4. Robotic programming

After the assembly phase was completed, when the power supply was connected to the circuit, the coding process was started in order to make the motors controllable with the help of Arduino.

```
void loop() {
    int olcum = mesafe(maximumRange, minimumRange);
    Serial.print("olcum: "); Serial.println(olcum);
    det = check();
    while (det == 'O')
    {
        det= check();
    }
    while (det == 'P')
    {
        det= check();
    }

    while (det == 'K')
    {
        det= check();
    }

    while (det == 'N')
    {
        det= check();
    }
    while (det == 'B')
    {
        digitalWrite(pingeri, HIGH);
        digitalWrite(pinsol, HIGH);
        det = check();
    }
    while (det == 'S')
    {
        digitalWrite(pinileri, LOW);
        digitalWrite(pingeri, LOW);
        digitalWrite(pinsol, LOW);
        digitalWrite(pinsag, LOW);
        det = check();
    }
    int olcum2;
    if(olcum>25 || olcum==0){
        det = check();
        while (det == 'F' && (olcum>10 || olcum==0))
        {
            olcum2 = mesafe(maximumRange, minimumRange);
            digitalWrite(pinileri, HIGH);
            digitalWrite(pinsag, HIGH);
            det = check();
            if(olcum2<25)
            {
                break;
            }
        }
```

Figure 4. Loops controlling mobility using distance

As can be seen in Figure 4, operations that would repeat more than once thanks to the Loop function were defined here. With the digitalWrite() function, the connection was provided on the Arduino, and provided the pins to take 1 (HIGH) or 0 (LOW) values. In this way, the directions of the wheels would be determined. Depending on the distance value, it was directed to the check() function according to the parameter value with the help of while loops and the rotation of the relevant motor was provided.

After the completion of the software, Arduino and Raspberry Pi 3 connection were made. The Raspberry Pi operating system was installed and the programming process with Python was completed after the connection of the camera.

od.py

```python
import cv2
thres = 0.45 # Threshold to detect object

cap = cv2.VideoCapture(0, cv2.CAP_V4L2)
cap.set(3,1280)
cap.set(4,720)

classNames= []
classFile = '/home/pi/Desktop/proje/coco.names'
with open(classFile,'rt') as f:
    classNames=[line.rstrip() for line in f]

configPath = '/home/pi/Desktop/proje/ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
weightsPath = '/home/pi/Desktop/proje/frozen_inference_graph.pb'

net = cv2.dnn_DetectionModel(weightsPath,configPath)
net.setInputSize(320,320)
net.setInputScale(1.0/ 127.5)
net.setInputMean((127.5, 127.5, 127.5))
net.setInputSwapRB(True)

while True:
    success,img = cap.read()
    classIds, confs, bbox = net.detect(img,confThreshold=thres)
    print(classIds,bbox)

    if len(classIds) != 0:
        for classId, confidence,box in zip(classIds.flatten(),confs.flatten(),bbox):
            cv2.rectangle(img,box,color=(0,255,0),thickness=2)
            cv2.putText(img,classNames[classId-1].upper(),(box[0]+100,box[1]+300),
            cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)

    cv2.imshow("Output",img)
    cv2.waitKey(1)
```

Figure 5. Image processing code

As can be seen in Figure 5, by means of this software, the camera was activated and the objects in front of the camera are recognized by using the MobileNet training set, which was included in the software and had previously trained data.

4. Findings and Comments

During the assembly phase of the wheels, they were tried to be placed in the gear system with a small amount of force due to the risk of breaking at the beginning, but they were not fully seated. However, as a result of the interview with a company operating in the field of robotics, it has been seen that the material is in a hard structure and sits in place with more force, through the examples of different projects they have shown. As a result, after the trials were made, it was seen that all 4 wheels were fully seated. It has been noticed that the screws sent to be used in the fixing phase of the 4 motors on the robot platform, due to their round cylinder heads, contact the wheels and cause them to rotate slowly. As a solution for this, 4 star countersunk headed screws are provided to ensure efficient rotation of the wheels.

The test processes of the L298N circuit, which is powered by 4 motors connected to the wheels, were started with the help of 2 7.4V lipo batteries. When you work with electronic boards making correct connections is very important, otherwise research cost would increase dramatically. It is seen in the Figure 6 that the picture of the burnt driver board accidentally at the first attempt, due to the lack of attention and connecting voltages reversely in this research.

Figure 6. Burnt out L298N voltage regulator

The driver motor circuit board, which works integrated with Arduino, was operated by feeding only the Arduino circuit in the first use. It was observed that the light that should be lit on the circuit did not turn on. Finally it is realized that this circuit needs to be fed with a power source other than Arduino. It was ensured that circuit also receives energy and ultimately drives the servo motors by providing an extra power source.

5. Results and Suggestions

According to the results obtained in the study, it was concluded that the software developed with the help of Arduino circuit, the designed robot can be controlled by any device by communicating with the motors with the help of motor driver circuits. It is possible to successfully hold objects suitable for the robot arm strength and also carry them from one place to another successfully.

One point to be taken into consideration is the voltage and amperage values of the power supplies feeding the circuits should be selected upon the power required. Moreover, the connections of these power supplies and circuits must be done very carefully, otherwise the circuits will burn.

One other important point is that the parts to be used in robot design should be selected in harmony and correctly. The choice of parts and their quality are very important in terms of both fixing the system with screws and stabilizing and work smoothly the system.

After the Raspberry Pi 3 installation, the previously trained MobileNet training data and the Opencv library were included in the project and the code was written with the help of Python to recognize the objects. Tests were carried out with 3 different fruits at different angles. As a result of these trials, it was observed that the objects can be recognized with high accuracy rates up to 95% when the necessary light and appropriate angles are provided.

As can be seen in Figures 7 and 8 the results obtained from the Python program with the help of 2D camera listed in Table 1.

It has been determined that the margins of error here are generally that the fruits are not completely in the camera view, difficulties in distinguishing the surfaces of the objects due to ambient lights, and problems in perceiving due to the angles of the objects and their proximity to each other (under-perceiving an object or assimilating a different object).

Table 1. Detection of real and foam fruits recognition rates

Fruits	Success Rates of Real Fruits	Success Rates of Foam Fruits
Apple	%98	%85
Banana	%93	%75
Orange	%97	%90
Apple and Banana	%96	%79
Apple and Orange	%95	%87
Banana and Orange	%86	%82
Apple, Banana and Orange	%90	%80

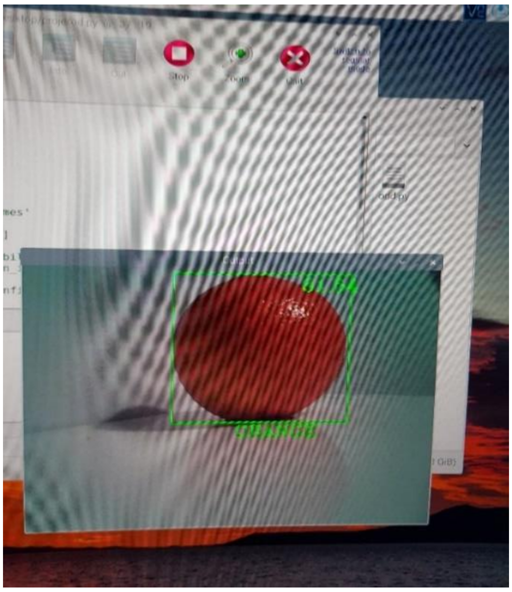

Figure 7. Orange image review

Prototype robot can hold and carry apple that made by foam currently. A hand made foam apple picture and capture it with robot arm are given in Figures 9 and Figure 10 respectively. Success rates of capturing some foam fruits are listed in Table 2.

Table 2. Success rates of capturing some foam fruits

Fruits	Success Rates
Apple	%70
Orange	%60
Banana	%35

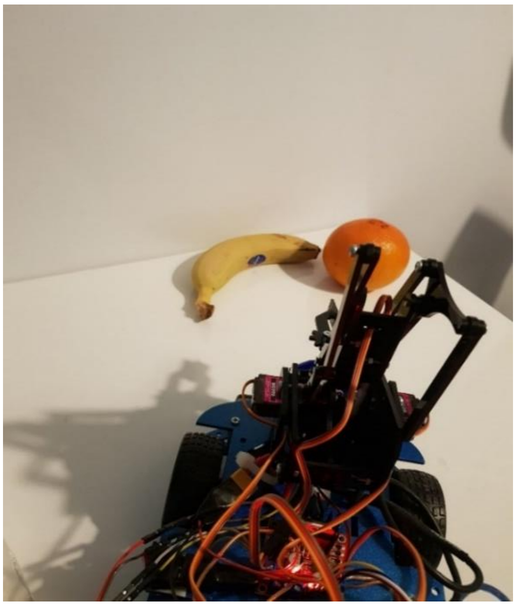

Figure 8. Orange and banana snapshot

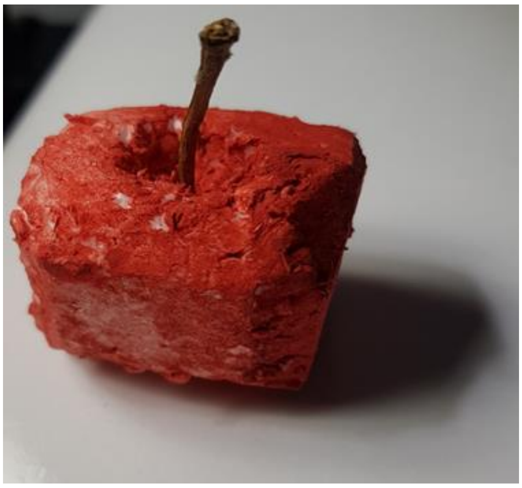

Figure 9. Image of foam apple

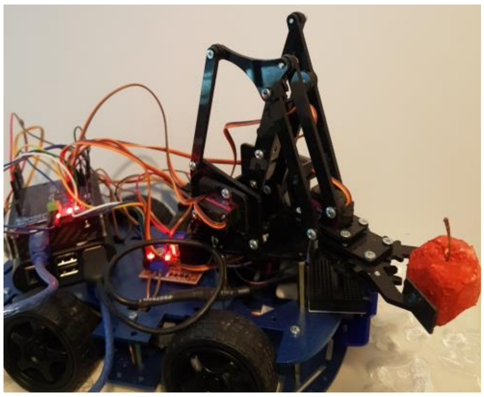

Figure 10. Image of capturing foam apple

6. Further works

The following issues should be considered in order to enhance the prototype model. First of all, more battery or powerful servo motors would provide better performance. Also, robot's arm could be modified with bigger and wider one. As a lack of movement stability, robot's wheels could be bigger for especially on pitted floors. Moreover, top of the robot can include 360° camera view for detect surrounding objects.

Author's Note

Previous version of this paper was presented at 10th International Conference on Advanced Technologies (ICAT'22), 25-27 November 2022, Van, Turkey.

References

[1] Wührl, L., Pylatiuk, C., Giersch, M., Lapp, F., Rintelen V, T., Balke, M., Schmidt, S., Cerretti, P. ve Meier, R., "Robotic handling of small invertebrates with machine learning methods", Molecular Ecology Resources, 0, 0, 2021, 3-8

[2] Lary J., D., Schaefer, D., Waczak, J., Aker, A., Barbosa, A., Wijeratne H. O., L., Talebi, S., Fernando, B., Sadler, J., Lary, T. ve Lary D., M., "Autonomous Learning of New Environments with a Robotic Team Employing Hyper-Spectral Remote Sensing, Comprehensive In-Situ Sensing and Machine Learning", Sensors: 20th Anniversary, 21, 6, 2021, 2-11

[3] Rahman M., Q., Corke, P. ve Dayoub F., "Run-Time Monitoring of Machine Learning for Robotic Perception: A Survey of Emerging Trends", IEEE Access, 9, 0, 2021, 20068-20073

[4] Katija, K., Roberts D L, P., Daniels, J., Lapides, A., Barnard, K., Risi, M., Ranaan Y, B., Woodward G, B. ve Takahashi, J., "Visual tracking of deepwater animals using machine learning-controlled robotic underwater vehicles", Winter Conference on Applications of Computer Vision, 0, 0, 2021, 861-867

[5] Hiçdurmaz, A. ve Tuncer A., "Real-Time Obstacle Avoidance Based on Floor Detection for Mobile Robots", Sakarya University Journal of Science, 24, 5, 2020, 847–851.

[6] Ruby, J., Daenke, S., Yuan, Y., Harry, W., Tisa, J., Nedumaan, J., Yung, Y., Lepika, J., Binford, T., Kumar J. S., P. ve Hu, W., "Automatic Speech Recognition and Machine Learning for Robotic Arm in Surgery", The American Journal of Surgery, 4, 0, 2020, 12-16

[7] Kahya E. ve Arın S., "Design of a Robotic Pneumatic Pruner for Robotic Apple Harvesting", European Journal of Engineering and Natural Sciences, 3, 2, 2019, 13–17.

[8] Karakoç A., Doğan D. ve Akıncı T.C., "Robotic Arm Control Using The Brain Waves", The Journal of Cognitive Systems, 2, 2, 2017, 52–54.

[9] Aivaliotis, P., Zampetis, A., Michalos, G. ve Makris, S., "A machine learning approach for visual recognition of complex parts in robotic manipulation", Procedia Manufacturing, 11, 0, 2017,424-429.

[10]Gopalapillai, R., Guptab, D. ve TSB, S., "Pattern Identification of Robotic Environments using Machine Learning Techniques", Procedia Computer Science, 115, 0, 2017, 64-69.

[11]Fard J., M., Ameri, S., Chinnam B., R., Pandya K., A., Klein D., M., ve Ellis D., R., "Machine Learning Approach for Skill Evaluation in Robotic-Assisted Surgery", The World Congress on Engineering and Computer Science, 1, 0, 2016.

[12]Batur D.G. ve Erol S., "Using Simulated Annealing for Flexible Robotic Cell Scheduling", Gazi University Journal of Science, 29, 3, 2016, 574-580.

[13]Kahya E. ve Arın S., "Kivi Meyvesi Hasadında Robot Kullanımı Üzerine Bir Araştırma", Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 18, 3, 2015, 4-8.

[14]Dr. Lukka J., T., Dr. Tossavainen, T., Dr. Kujala V., J., ve Dr. Raiko T., "Robotic Sorting using Machine Learning", ZenRobotics Recycler, 0, 0, 2014, 2–7.

[15]Perkowski, M., Sasao, T., Iseno, A., Wong, U. ve Lukac, M., "Use of Machine Learning Based on Constructive Induction in Dialogs with Robotic Heads", The Eighth International Conference on Rehabilitation Robotics, 0, 0, 2003.