AUTOMATED OBJECT SORTING ROBOT USING MACHINE LEARNING

Yashas MA\textsuperscript{1}, Amogh CR\textsuperscript{2}, Chetan S\textsuperscript{3}, Harshitha B\textsuperscript{4}, Jagadeesh BN\textsuperscript{5}

\textsuperscript{1,2,3,4} Student, Dept. of CS&E, East West Institute of Technology, Bangalore, Karnataka, India
\textsuperscript{5} Assistant Professor, Department of CS & E, East West Institute of Technology, Bangalore, Karnataka, India.

ABSTRACT

The following publication describes a solution to object sorting with a robotic arm with the aid of Machine learning to identify and differentiate objects. We have used a cost effective approach in illustrating a prototype for Automated Object Sorting Robot, one of the major components of this project is Raspberry Pi 4 which is easy to use as it runs on Raspbian OS which is Open source Operating system allows developers to do modifications required for their respective projects, Freeduino USB 1.1 microcontroller which is a clone of Arduino UNO but contains bare essential functionalities required to us, it is mainly used to control the servo motors that drive the robotic arm and is also serially connected to the Raspberry Pi 4. Pi camera is used for capturing the image of the object, once captured the Raspberry Pi processes the image with the help of algorithm and generates labels if object that is identified this label is checked with the target list (list containing objects that we want) if found then Pi sends the signal to the Freeduino microcontroller to actuate the Arm and Sort the respective object in predetermined location.


I. INTRODUCTION

For a long time now the role of automations and machines or robots have become one of the most important innovations in removing the need for human labor. The usage of such robots in industries have become essential for the stability of economy and meeting the demands of clients who directly or indirectly depend on such machines for their needs. Modern machines when built right can outperform humans in many tasks with better accuracy, speed and more efficiently. They are usually less expensive than paying the salaries of multiple workers and its ability to manufacture/process a huge volume of entities are much better than that of humans. One of the most common application for automations in the industry is to sort and classify objects. Most of the solution for automated object sorting applications that involve robotic arms make use of simple solutions for identifying objects, these involve extracting very simple features from the object such as general shape or color of the object. However, this solution cannot be applied to every scenario or all objects. A better way to identify the objects involves the usage of Machine Learning. This publication describes an "Automated Object Sorting Robot Using Machine Learning" that is capable of sorting different objects based on the classifications made by the machine learning model. The basic operation covers identifying the object using a machine learning model, a robotic arm then picks up the arm and then places it in a destination that depends on the output of the machine learning model.

II. METHODOLOGY

The main modules that cover every major necessary operation can be generalized into 3:

- The Object Identification Module.
- The Arm and Arm Control Module.
- The Communication Module

a) The Object Identification Module

The object identification module is responsible for identifying an object that needs to be identified. To achieve this we have used Yolo V3 Algorithm (You Only Look Once, is a popular and fast convolution neural network based object identification algorithm) which runs on a Raspberry Pi 4 System. The model has been trained on the Coco dataset. The input for the model needs to be an image, this is taken by a Pi camera attached to Raspberry Pi 4. We have a target list which is a list of specific items we are looking for, and if the resulting label from the image of the object is a part of the target list then the arm will move to pick and place the object in a predetermined location, else the object will not be picked up. To decide on the resulting destination location the index of the object found in the target list will be used to decide where the picked up object will be placed.
b) The Arm and Arm Control Module

The Arm itself is made out of cardboard and a set of 4 servo motors. Among the 4 motors, 3 of them are MG995 motors used for base, shoulder and elbow rotation, and the last one is an SG90 motor for the gripper. Controlling the Arm is done with the help of Freeduino USB which contains the Arduino Code for performing the Operation of Picking Object and placing it in a different location. So for this purpose we are using 5 different states in the code which are Reset Position, Pickup Position, Lifting Position, Base Turning and lastly dropping position. The following sequence depicts the procedure followed by the arm when it is tasked with picking and placing an object. The destination of the object depends on the index of the object in the target list, based on this index the angle of base rotation is changed in step 4 as shown below.

1. Enters the Reset Position, checks whether it’s in reset position or not if not then comes to reset position.
2. Enters the Pickup Position, moves to pick up the object.
3. Enters the Lifting Position, to lift object to certain height.
4. Enters Base Turning Position, where Base of the Arm rotates and along with the object being lifted.
5. Then it Enters Dropping Position, where the elbow moves down and loosens up the gripper and drops the object in particular destined place.
6. Finally goes back to the Default position or Reset Position.
The Communication Module

To enable communication between two boards like Raspberry Pi and Freeduino USB the simplest way is to establish a serial connection.

This can be done in 2 ways –

- With 4 GPIO pin connections
- With a single USB connection

The single USB connection is preferred due to there being only one connection needed, making it less of a hassle to add or remove. The protocol for exchanging information here that was used here is called UART [universal asynchronous reception and transmission]. This is an Asynchronous Multimaster Protocol, meaning there are no master slave configurations and both boards can send data to each other at any time as long as they are connected. 1 bit of data is sent at a time and this is stored in the USB UART buffer on the other side. This data can be extracted whenever needed on either side easily as there are libraries in python and the Arduino environment that allow for this. In python the library “serial” is used and "Serial" is used in Arduino. Both libraries come with functions that check the number of bytes available in the buffers and based on this we can extract data. The actual data that will be sent from the raspberry pi will be the index where the detected label has been found in the target list. After the Arduino receives this data it can decide where to drop off the object

III. MODELING AND ANALYSIS

a) Algorithm showing overall working procedure with all 3 modules:

Step 1: Take an image from the Pi-camera
Step 2: Preprocess the image [reduce its size to 40% of its original width and height, convert to blob]
Step 3: Pass the blob to the input layer of the network
Step 4: Extract the output vectors from the output layer
Step 5: Find the highest confidence value within the list of confidences [hcv]
Step 6: Find the label name [dl] that corresponds to the highest confidence value (hvc)
Step 7: Check if dl is a part of the target list if it is a part of the target list, go to
Step 8 Else, go to Step 1
Step 8: Retrieve the index of the string dl from the target list
Step 9: Prepare a message to be sent to the Freeduino USB board that contains this index and send it serially via USB cable to Freeduino USB and go to a waiting state
Step 10: In Freeduino, parse the message sent by the raspberry pi
Step 11: Set the appropriate destination of the object based on the received input
Step 12: Pick up the object, rotate the base motor over to the appropriate destination and drop the object there
Step 13: Reset angles of all the motors in the arm to the position it was before picking the object up
Step 14: Send an acknowledge string to raspberry pi stating it’s completion of work
Step 15: In raspberry pi, receive the acknowledgement and go to step 1

b) Architecture of the Apparatus:

From the following figure which shows the Architecture of the following proposed system as how the connections are made between the components. Some of the major components used are Raspberry Pi Model 4, and Freeduino USB 1.1 microcontroller. And the components such as Camera module, Servo Motors which include MG995 servos of 3 quantity and SG90 Servo of 1 quantity. So from the figure it is clear that Pi-Camera module is directly attached to the Raspberry pi’s 2 Lane MIPI CSI port which is built on it for that purpose
Fig-3: Architecture of the Apparatus.

c) Work Flow between the two boards:

Fig-4: Work Flow between the boards.
IV. RESULTS AND DISCUSSION

The above figure shows the top-view of the entire setup, depicting how the components have been connected with each other.

Testing the apparatus:

This is the main test where all the modules are functioning together and are given 2 different classes to differentiate. There were 2 positions for the arm to drop them off, one at 110 degrees and the other at 160 degrees in code for the base rotation motor. The default position if the base rotation motor is set at 90 degrees. The classes used for the trial were "cats" and "cakes". The set destination for cats was 110 (slight left from the perspective of the arm) and 160 for the cake (far left in the perspective of the arm). The objects were placed one at a time and the program was run. Observation: The arm successfully classified the two classes of objects and placed the cat object at slightly to the left of the arm (110) and the cake object to the far left (160). The test was repeated multiple times and the same result was observed. The video of System Test can be seen by the following link attached below.

![Figure 5: The complete view of apparatus.](image)

![Figure 6: Target List.](image)
Fig-7: Pictures taken by Pi camera during Test.
V. CONCLUSION

In this paper we have concluded that “Automated Object Sorting Robot using Machine Learning” project was able to successfully overcome the drawbacks of previous approaches and can be implemented for a lot of applications for industrial purposes. Currently the process of identifying an object from the time taken to take an image to process it is about 0.25 seconds on the Raspberry Pi 4, this may be improved by using a dedicated microcontroller more suitable for the process. Further optimizations will be made to speed up the entire process and also replacement of the MG995 motors with a more reliable Servo motor will yield better accuracy with determining the angles exactly and also improve repeatability. Replacing the cardboard parts will also make the system more rigid and less prone to failure also it increases the amount of weight that it could lift based on material used. By using a small touch based display and attach it for 2 lane MIPI DIS port we can make our robot portable and convenient to use, and we could also provide mobility feature for this robot which could actual help it to pick objects and travel by itself and place the object in desired destinations which enlarges the field of applications where it could be used.

VI. REFERENCES


