IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CREATION AND IMPLEMENTATION OF E-TROLLEY: AN AFFORDABLE INTELLIGENT SELF-GUIDED CART FEATURING ANTI-THEFT SAFEGUARDS

[1] Dr. Dinesh D. Patil, [2]Dimpal Liladhar Narkhede

[1]Head Of Department, [2] Student

[1] Department of Computer Science and Engineering,

[1] Shree Sant Gadge baba Engineering College, Bhusawal, India

Abstract: Shopping and buying are an integral part of our daily lives. Big Supermarkets have a wide variety of items and different stores can have different varieties of goods. It is difficult for many consumers to stand in long queues, just for billing of the goods purchased. This leads to a waste of time and a poor experience for the customer. Trolleys are used in supermarkets & and grocery stores to make shopping simpler. However, it is difficult for customers to handle the trolley while shopping.

***Index Terms* - Raspberry-pi, QR Scanner, Image processing.**

I. Introduction

In the modern world, every supermarket and hypermarket employs shopping baskets and shopping trolleys to aid customers in selecting and storing the products which they intend to purchase it. The customers have to carry their trolleys everywhere and drop every product that they wish to purchase into the shopping cart and then proceed to checkout. The billing counter. This process is quite tedious and highly time-consuming and has created the need for shops to employ increasingly more human resources in the billing section and wait times remain considerably high at most times. Our Project Design and Development: A Low-Cost Intelligent Autonomous Trolley with an Anti-Theft Mechanism" Aims to remove the efforts required by the customers to move the trolley and reduce the Total waiting time of customers. Lower the total workforce requirement and expenses for Supermarkets and increase efficiency overall, as well as provide a contactless billing process to the customer.

Shopping and buying are an integral part of our daily lives, and trolleys are used as tools to make shopping simpler. It is usually a tiresome process for many consumers to shop for everything and then stand in long queues just for billing of the goods purchased leading to wastage in time and poor experience for the customer. However, it is often not easy for customers to handle the trolley while shopping. Also, considering the COVID situation - customers might even hesitate to touch the trolley. Our project work intends to solve this problem.

To summarize the complete shopping experience, the customer scans a QR code on the trolley from the Smartphone App, which uniquely identifies the trolley and a communication is established over Bluetooth between the trolley and the App. After capturing an image of the customer, the trolley can identify the human silhouette and can start following the customer.

Using an RFID (Radio Frequency Identification) scanner on the trolley, the items are scanned & billed concurrently in the Smartphone App. If a customer/meddler removes/adds a certain item from/to the trolley without scanning, a Vibration, along with a short message is displayed on the Smartphone, to indicate tamper/theft. The customer can finally Checkout using any Digital payment method to pay for the billed amount.

II. LITERATURE REVIEW

Paper [1]: "Automated Billing Smart Trolley and Stock Monitoring", 2021 by M. K. Dev et al. describes a design with Radio Frequency Identification (RFID) Technology for billing with Arduino Uno and an Liquid Crystal Display (LCD) Screen & also devised a stock monitoring IoT platform. They have provided for online or offline payment methods. No kind of theft / tamper detection and no autonomous movement devised.

Paper [2]: "Automated Shopping Trolley for Super Market Billing System", 2014 by J. Thangakumar et al. demonstrates a design with a Raspberry Pi as the main computing device with barcode scanners employed on the trolley to bill products with a barcode. Used an LCD monitor, increasing the overall cost and similar to the previous paper, no kind of theft / tamper detection and no autonomous movement devised.

Paper [3]: "Human Friendly Smart Trolley with Automatic Billing System", 2020 by Hanooja T et al. is using an elaborative design using RFID reader with a Raspberry Pi and using individual buttons to add/remove products, which is tedious for multiple products. Furthermore, no tamper protection is provided. They have also implemented a following mechanism using a Color Tag, which would be given to the user, such that the camera detects a specific color to track and line follower to keep trolley in track. The line follower is useful to prevent collisions, but color detection is highly prone to false negatives, and thus is susceptible to false positives.

Paper [4]: "Travelmate Robot: Smart Touchless Trolley" by D. S. Radhika Shetty proposes a model using the concept of patterned color tags to follow a user and a camera to detect the color code. Color code detecting of the trolley for following customers is not ideal in a supermarket where the density of customers is high and light could affect the working.

III. BLOCK DIAGRAM AND WORKING PRINCIPLE

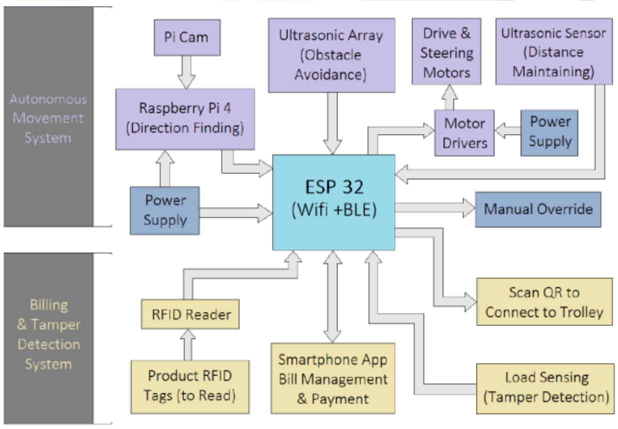

fig. 1: block diagram of the proposed methodology

The whole project is divided into 2 parts: an Autonomous Movement System and a Billing and tamper Detection System For the Autonomous following of the trolley, a TensorFlow Lite Model - 'Pose Estimation Model' has been used to detect the user's Silhouette. This machine learning model has been chosen as it best outlines the crucial landmark features of the human body accurately (even from behind) and is decent for use on our Image Processing Microcontroller of Choice – Raspberry Pi 4.

This data is used to steer the trolley to keep the person centered i.e., whenever the user moves, the camera tries to keep the user centered and the steering information is conveyed to the steering motors through the ESP32.

When no person of interest is detected on the frame of the camera, the trolley detects everything as an obstacle and by using one of the left and right ultrasonic sensors, avoids any obstacle while moving. The distance finding between the trolley and customer is taken care of by an ultrasonic sensor on the trolley.

Further, two ultrasonic sensors (left and right) are used to find out possible paths to move when an obstacle is faced in front of the trolley or when the user's Silhouette is blocked. Smart Billing is accomplished by using Radio Frequency Identification (RFID) Technology, maintained by the ESP32 microcontroller chip (MCU) Since the ESP32 powers and communicates with the RFID Module & Load Cell directly, the communication between the smartphone App and the ESP32 is handled using Bluetooth Low Energy (BLE) on the SoC.

The proposed model uses a unique QR Code on each trolley, which is the MAC Hardware address of the Bluetooth adapter on the ESP32. By scanning the QR Code on the App, the user gets connected to the trolley. Whenever 'RFID Product Tags are read', the ProductID is sent along with the simultaneous values of load sensor readings over a single Bluetooth Service with two separate Bluetooth Characteristics, for Tag Data and Load Cell Reading.

For our project, we have used RFID Cards, which are written manually with a unique ProductID as the data for each product. When the RFID Tag-affixed products are scanned on the RFID Reader, the data is sent to the smartphone app.

The app then processes this ProductID data to add/remove the product in the cart and calculate the total bill thereafter. Regarding Theft Detection, Load Cell values that are constantly being sent over the channel are processed and detect any weight changes. This enables us to quickly detect any weight changes (increase/decrease) and alert immediately on the Smartphone app with a continuous Vibration and a short message and prevents any attempts at tampering / theft by any meddler

IV. CIRCUIT DIAGRAM

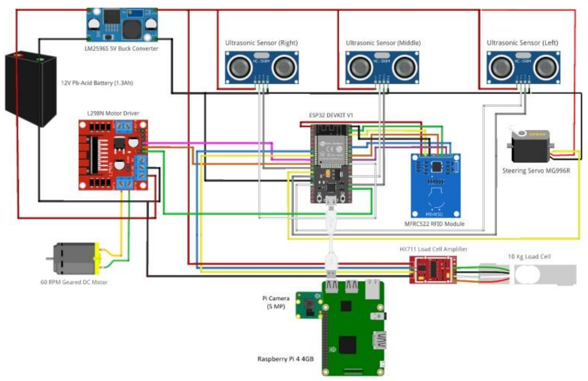

Fig. 2: Circuit Diagram

The above circuit connection diagram shows the complete connections between various components in the fabricated trolley design.

1. Autonomous Movement System: Consists of the complete movement mechanism of the trolley, which further consists of the following components.

- Raspberry Pi 4: Raspberry Pi 4 Model B – 4GB.
- Pi cam: Raspberry Pi 5MP Camera Module. Supports 1080p30, 720p60 video.
- Ultrasonic Sensor Array (Left, Middle & Right)
- Driving Motor: Johnson 12V 60-RPM DC Motor • Motor Driver: L298N Motor Driver (supports up to 2A)
- Wheel Size (diameter): Front – 7 cm. Rear - 9.5 cm • Steering Servo: MG996R 10Kg-cm torque

2. Billing & Tamper Detection System: Consists of the complete billing mechanism for the user and the anti-theft system, which further consists of the following components:

- RFID Reader: MFRC 522, Operating Range 13.56 Mhz.
- Product RFID Tags
- Load Sensor: 10 Kg Load Cell, for anti-theft mechanism.
- HX711 Load Cell Amplifier
- Smartphone App: Connects the user to the shopping cart, built using Flutter and Dart (a programming language)

3. Power Supplies Used:

- 12V 1.3Ah Pb-Acid Battery: for Motors, Servo, Load Cell & Ultrasonic Sensors.
- LM2596S DC-DC Buck Converter, adjusted for 5V DC Output for Servo and Ultrasonic Sensors.
- 10,000 mAh Power bank: for Raspberry Pi & ESP32

V. FLOWCHART

1. Flow- Chart for the working of the e-Trolley App:

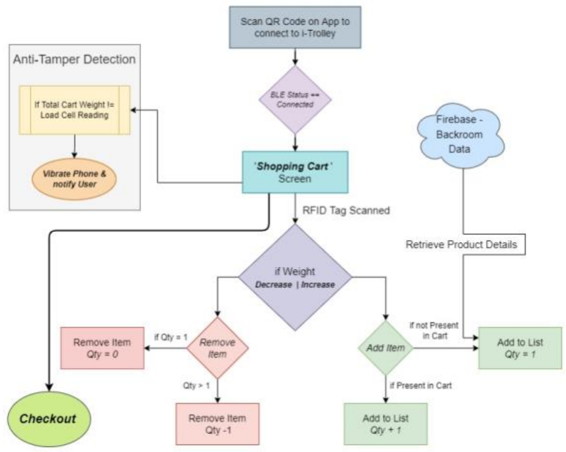

Fig. 3: Flowchart of e-Trolley App Flow

Table 1: Pose Detection Landmarks with id & locations

This chart provides a succinct overview of the App's functioning, from scanning a QR Code to completing the Checkout. Additionally, it explains how the app amends the cart based on weight. The App also provides a "Subtotal" and "Total" of the items before and after any discounts or surcharges are applied. The App displays a summary of all the items in the cart, including the subtotal and total cost. This summary is then used to complete the checkout process. Thanks to the App, customers can now experience a modern and streamlined shopping experience.

2. Flow- Chart for the working of the Image Processing for Autonomous Steering

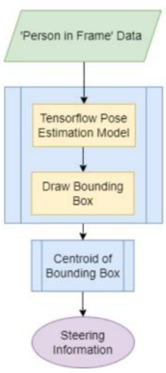

Fig. 4: Flow-chart of Autonomous Following Processing

VI. Working oF the Complete Project Module

The solution that was recognized as the most beneficial was to use a Pose Detection Model provided by TensorFlow Lite, which was originally built to detect different poses and classify them. However, we have employed the model to detect a human silhouette.

This idea was empowered by poor, inaccurate, and false-negative-ridden – object detection algorithms that can track objects but fail after a person goes out of frame or gets stuck on other elements in the environment rather than the person.

The pose estimation model accurately identifies several landmarks on the human body and joins them to create a human silhouette. This especially works well from behind, i.e., back faced towards the camera, since the model tries to estimate the human landmarks based on body structure. Other algorithms usually fail in this regard, making the 'literal following 'part failure-prone

Future scope could include merging interval-based Object Detection and pose Estimation Model-based Human detection together for faster processing

TensorFlow's Pose Estimation Algorithm:

Pose estimation is using an ML model to estimate the pose of a person from an image or video. You estimate the spatial locations of key body joints (key points).

How the model works:

Pose estimation is important in computer vision. It identifies where humans are in pictures and videos, like where a person's elbow is. It's not accurate enough to know who's in an image or video, but it does estimate where the bodies are.

The pose estimation models take a processed camera image as the input and output information about key points. The part ID numbers index the key points found, and a confidence score from 0.0 to 1.0 is associated with each point. This score reflects the chance that a key point is present at that location.

TensorFlow provides a reference implementation of two TensorFlow Lite pose estimation models:

MoveNet: the latest pose estimation model available in two versions: Lighting and Thunder.

PoseNet: the previous generation pose estimation model released in 2017

The table below shows the different body joints that are picked up by the pose estimation model.

ID	PART
0	**NOSE**
1	**LEFTEYE**
2	**RIGHTEYE**
3	**LEFTEAR**
4	**RIGHTEAR**
5	**LEFTSHOULDER**
6	**RIGHTSHOULDER**
7	**LEFTELBOW**
8	**RIGHTELBOW**
9	**LEFTWRIST**
10	**RIGHTWRIST**
11	**LEFTHIP**
12	**RIGHTHIP**
13	**LEFTKNEE**
14	**RIGHTKNEE**
15	**LEFTANKLE**
16	**RIGHTANKLE**

Table 1: Pose Detection Landmarks with id & locations

MoveNet is available in two versions:

MoveNet.Lightning is more compact, quicker, but not as precise as the Thunder version. It can carry out operations in real-time on contemporary smartphones.

MoveNet.Thunder is more precise than its smaller and quicker counterpart, Lightning, but it is also slower. It is useful for use cases that require higher accuracy.

MoveNet demonstrates higher accuracy than PoseNet on a range of datasets, particularly with respect to images depicting physical activities. Therefore, MoveNet is recommended over PoseNet.

VII. Limitations:

Some of our current limitations include:

- RFID Tagging every product
- Autonomous systems are not as good at tracking a single person. Retracking the person may induce false positives when more people are present in the frame.

VIII. ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my research guide Dr. Dinesh D. Patil for their invaluable guidance, unwavering support, and expert mentorship throughout this research project. Their dedication and insightful feedback have been instrumental in shaping the outcome of this work.

I am also thankful to my college Shri Sant Gadge Baba College, Bhusawal, Maharashtra,India for their contributions and support during this research endeavor. Their assistance has been greatly appreciated.

IX. REFERENCES

[1] M. Kabil Dev, R. Kannan, M. Agarshan, S. Karthik, and K. Lakshmi, "Automated Billing Smart Trolley and Stock Monitoring", 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), doi: 10.1109/ICCMC51019.2021.9418043, pp. 500-505, 2021.

[2] J. Thangakumar, S. Sainath, K. Surender, and V. Vikram Arvind, "Automated Shopping Trolley for Super Market Billing System", International Conference on Communication, Computing, and Information Technology (ICCCMIT-2014), pp. 7-9, 2015.

[3] T. Hanooja, C. G. Raji, M. Sreelekha, J. Koniyath, V. Muhammed Ameen, and M. Mohammed Noufal, "Human-Friendly Smart Trolley with Automatic Billing System", 2020 4th International Conference on Electronics, Communication, and Aerospace Technology (ICECA), ISBN: 978-1-7281-6387-1, doi: 10.1109/ICECA49313.2020.9297439, Conf. Paper id No., pp. 1614-1619, 2020.

[4] D. S. Radhika Shetty, "Travel mate Robot: Smart Touchless Trolley", IJRESM, vol. 4, no. 8, pp.

[5] http://www.neilkolban.com/esp32/docs/cpp_utils/html/class_m_f_r_c522.html

[6] https://how2electronics.com/iot-weighing-scale-hx711-load-cell-esp8266

[7] https://randomnerdtutorials.com/esp32-hc-sr04-ultrasonic-arduino