Smart Shopping Cart with Automatic Billing using Deep Learning System: A Review

D Yogesh Kumar
Department of Electrical and Electronics Engineering
Dayananda Sagar Academy of Technology and Management, Bengaluru

Chinmay M R
Department of Electrical and Electronics Engineering
Dayananda Sagar Academy of Technology and Management, Bengaluru

Prashant Dhanmalle
Department of Electrical and Electronics Engineering
Dayananda Sagar Academy of Technology and Management, Bengal

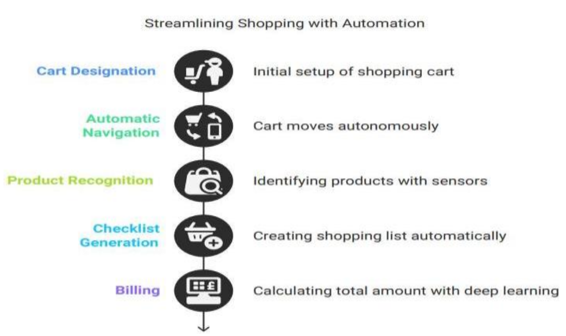
Prajwal S Managuli
Department of Electrical and Electronics Engineering
Dayananda Sagar Academy of Technology and Management Bengaluru

Prof. Monica K.M
Assistant Professor, Dept. Electrical and Electronics Engineering
Dayananda Sagar Academy of Technology and Management, Bengaluru

Abstract—The retail industry is rapidly evolving with automation and AI enhancing the in-store experience. Traditional barcode or RFID-based checkouts cause delays and inefficiencies. This paper introduces a Smart Shopping Cart using deep learning for real-time object detection via a YOLO model on a Raspberry Pi. Items are recognized and billed automatically, removing the need for manual scanning. Weight sensors refine recognition of unpackaged goods, while face-tracking enables autonomous cart following. The system ensures accurate billing, smooth navigation, and supports fresh produce handling. It reduces human intervention and operational costs. This scalable solution advances smart, contactless retail automation.

Keywords-Smart Shopping, Cart billing system, Computer vision, Deep learning, Automation

INTRODUCTION



Technology is transforming our everyday experiences from how we shop to how we secure our vehicles. While digital innovation has reshaped many aspects of modern life, two areas still face persistent challenges: the in-store retail experience and vehicle security. In both cases, conventional systems are often slow, inefficient, and poorly equipped to meet the expectations of today's fast- paced, connected world.

Take the example of physical shopping. What should be a quick and seamless task often turns into a frustrating routine, with long checkout queues, barcode scanning issues, and difficulties managing fresh, unpackaged goods. Although e-commerce offers convenience, physical retail remains irreplaceable, especially for groceries and daily essentials. This calls for a smarter, more intuitive in-store experience.

Similarly, in the automotive space, rising vehicle thefts and unauthorized access incidents have exposed the limitations of traditional locks and alarms. As vehicles become more connected and autonomous, there is a pressing need for intelligent, layered security systems that can respond in real time to potential threats.

Fortunately, the convergence of embedded systems, artificial intelligence, and IoT is enabling innovative solutions in both domains. In retail, smart shopping carts equipped with deep learning and computer vision are transforming the way customers interact with products. These carts can recognize items automatically using YOLO-based object detection, track real-time billing, and even follow users around the store using face-tracking. With additional features like weight sensors for fresh produce and digital payment integration, the entire checkout process becomes seamless and contactless eliminating the need for cashier counters altogether. In parallel, the proposed intelligent vehicle security system integrates GPS, GSM, biometric authentication, and microcontroller-based logic to deliver multi-layered protection. Features like GSM-controlled ignition, fingerprint door access, GPS tracking, and rash driving detection (via an ADXL335 sensor) provide both active and passive safety. A 16x2 LCD ensures the driver stays informed through real-time alerts and status updates.

Both systems reflect a shift toward user-centric, embedded intelligence whether it's in a super-market behind the wheel of a car. By harnessing deep- learning, and microcontroller platforms like Raspberry Pi and Arduino UNO, these projects demonstrate how practical, affordable technologies can solve real-world problems with efficiency and elegance. Together, these innovations underscore the potential of smart embedded systems in enhancing convenience, safety, and user autonomy across different sectors of daily life.

I. Overview

1. Web Interface Integration

To initiate the smart shopping process, a web-based interface is implemented, allowing users to log in or register prior to accessing the smart cart system. The interface captures user credentials, including username, email, and password, and upon successful authentication, redirects the user to the primary shopping dashboard. This dashboard is synchronized with the cart's onboard display unit, enabling real-time interaction as products are added and automatically reflected in the digital bill. The platform also supports user-specific data handling for personalization, along with integrated billing and multiple payment options. This interface functions as a critical entry point to the smart shopping system, enhancing the overall user experience through convenience and automation [1]. A mobile application was developed to streamline user access to the smart cart system. New users register, while existing users log in using their credentials. Upon login, a unique QR code is generated and displayed using Android Studio's Image View. This code is scanned at the store entrance, where cameras capture multi-angle facial images for identity linking. These images support facial recognition during shopping, enabling secure authentication and seamless integration with the user's virtual cart.[10] To initiate the smart shopping experience, a web-based interface was developed, allowing users to log in or sign up before using the smart cart. The interface presents a login window where users enter their credentials such as username, email, and password. Upon successful authentication, they are directed to the main shopping window, which is integrated with the cart's display. This interface is synchronized with the cart's onboard system, enabling real-time interaction as products are scanned and added to the digital bill. The platform ensures a smooth and personalized experience by maintaining user-specific data, and it supports various billing features and payment options after shopping. This web interface acts as the gateway to the smart cart system, combining convenience and automation to streamline the entire shopping process.[3] To enable a smooth and personalized shopping experience, the developers created a user-friendly mobile application that serves as the gateway to the smart cart system. New users are first required to register through the app, and returning users can log in using their credentials. Upon logging in, a QR code is generated for each user using a QR generator, which is then displayed through Android Studio's Image View component. This QR code is scanned at the store entrance, where a set of cameras captures multiple facial images from different angles to associate the user's face with their account. These images are stored and later used for facial recognition during the shopping process. This seamless interface not only authenticates users but also links them directly to a virtual shopping cart, making the entire process intuitive and automated from entry to checkout.[10]

2. Automatic Navigation

To enhance user convenience, the smart shopping cart is designed to move autonomously, eliminating the need formanual pushing. This is achieved by integrating two motors controlled by an Arduino, which allows the cart to follow the shopper. The autonomous movement is facilitated by face detection technology using web cam, enabling the cart to navigate in tandem with the user. This feature aims to provide a hands-free shopping experience[1]. The smart trolley system aims to enhance customer convenience through semi-automated features, laying the groundwork for future automatic navigation. While the design[2] does not include full robotic movement, it proposes integrating sensors and IoT connectivity to support intelligent navigation within store environments. The system envisions using RFID and sensor data to eventually enable the trolley to guide itself or assist users more actively, reducing the need for manual handling and optimizing the overall shopping experience[2]. The smart trolley employs infrared (IR) sensors to achieve autonomous navigation within the retail environment, enabling shoppers to traverse the store efficiently. This obstacle avoidance system allows the trolley to navigate aisles and circumvent obstructions, thus aiming to enhance the shopping experience by reducing navigational burdens. A notable drawback of relying solely on IR sensors, however, is the system's inability to detect transparent objects. This limitation poses a risk of collisions with glass displays or other transparent fixtures commonly found in retail settings, which could impede the trolley's navigation and potentially damage store property[3].Automatic navigation in smart shopping carts is achieved through deep learning-based frame classification and shopping activity recognition.A mounted video camera records shopping behaviors, and image processing models such as Faster R-CNN and YOLOv2 categorize frames into No Hand, Empty Hand, and Holding Item classes. The system segments shopping actions using Dynamic Time Warping (DTW) to match event timelines and determine accurate item placement or removal[8].

3. Product Recognition

- YOLO models

The smart shopping cart system employs deep learning for product recognition to automate the billing process. The system utilizes a camera integrated into the cart to capture images of products as they are placed inside. These captured images are then processed using deep learning models to identify and classify the items[1]. The system relies on YOLO-based object detection, a real-time algorithm that processes entire images in a single forward pass of a neural network, predicting bounding boxes and class probabilities simultaneously.YOLO partitions the image into grids, predicting object presence, bounding box dimensions. The model is trained on annotated datasets where each image is labeled with object coordinates. Despite its advantages, the system faces notable drawbacks. Variability in lighting conditions, occlusions, and shape distortions can result in misclassification, affecting billing accuracy.The system currently supports only a limited number of product categories, necessitating extensive dataset expansion and model refinement for broader applicability[5]. The system employs Faster R-CNN, YOLOv2, and YOLOv2-Tiny for frame classification, distinguishing images based on three classes: No Hand (NH), Empty Hand (EH), and Holding Item (HI). YOLO-based models achieve higher processing speeds, with YOLOv2-Tiny reaching 50 frames per second (fps). Dynamic Time Warping (DTW) is applied to enhance the accuracy of shopping action recognition, segmenting timelines into distinct shopping events such as placing, removing, or swapping items[8]. The system architecture relies on YOLOv8 for high-speed object detection. The neural network undergoes multiple training epochs to improve classification accuracy. Each image frame is divided into grid cells, which the model evaluates to detect products accurately. Training involves iterative loss reduction through cross-validation, enabling the system to generalize effectively across diverse product appearances. The evaluation metrics, including Mean Average Precision (mAP) and Intersection over Union (IoU), measure detection performance and minimize false classifications[20]. The system utilizes YOLOv2 to classify frames into three categories: No-Hand (NH), Empty-Hand (EH), and Holding-Item (HI). A smartphone mounted on the shopping cart captures shopping activity, which is transmitted to a cloud server for processing. YOLOv2 detects objects in video frames, enabling automated recognition of purchased items. Action segmentation follows, identifying shopping events such as item addition, removal[9].

- CNN(convolution neural netwroks)

Product recognition is achieved through a combination of facial recognition and sensor-based item tracking. Upon store entry, customers scan a QR code, and their facial features are recorded through multiple cameras. When an item is picked from a shelf, weight sensors (load cells) detect the change, triggering nearby cameras to identify the customer. A deep learning-based convolutional neural network (CNN) processes these images for face recognition[10]. SmartCart employs convolutional neural networks (CNNs) for image similarity.Additional machine learning models, including Support Vector Machines (SVM), Decision Trees, Random Forests, and K-Nearest Neighbors (KNN), assist in classification and user preference prediction. The system is trained on a dataset of over 5,000 images to improve recognition efficiency[19]. CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layer extracts local features by applying learnable filters to input images, generating feature maps that highlight important visual characteristics. The pooling layer reduces feature dimensions while preserving essential information, employing max or average pooling techniques to enhance computational efficiency. The fully connected layer integrates extracted features to classify products based on learned representations.Popular CNN architectures utilized in product recognition include LeNet-5, AlexNet, VGG16, ResNet, and MobileNet[22]. Product recognition is achieved through a combination of image processing and weight-based validation. A camera mounted on the autonomous trolley captures images of the purchased items, which are then processed using CNN models trained on specific product datasets. The captured image is analyzed for feature extraction, followed by classification into predefined categories. Parallelly, a load cell measures the item's weight, validating product identity and price estimation. The integration of image-based recognition and weight assessment enhances checkout efficiency[13].Product recognition is implemented through a CNN model based on the U-Net architecture, which is trained to predict attention maps highlighting OOS areas. The model processes images captured in retail environments and generates feature representations through multiple convolutional layers. Instead of using raw OOS coordinates, a mid-level representation is created using a binary mask and distance transformation techniques, refining the accuracy of the CNN's predictions. The CNN learns hierarchical patterns, enabling effective recognition of empty shelf spaces, assisting retailers in product replenishment strategies[18].

- Mobile Computing, Mobile Net v2

iCart employs a smartphone-mounted camera to record shopping activities, transmitting video data to a cloud server for processing. The system utilizes YOLOv2 deep learning models for object detection, segmenting frames into No-Hand, Empty-Hand, and Holding-Item categories. Dynamic Time Warping (DTW) is used to evaluate action intervals, determining whether a product is being placed or removed from the cart. Shopping lists are updated in real-time through mobile applications linked to cloud services[9]. Product classification utilizes deep learning models trained on a dataset of 32,000 images across 32 product types. The models tested include MobileNetV2, CNN, VGG16, InceptionV3, and ResNet, with MobileNetV2 achieving the highest accuracy of 92.80%. The classification process involves image preprocessing, neural network training, and cloud storage integration for seamless retail management[17]. The product recognition system is implemented using deep learning models— MobileNetV2 and EfficientNetV2. A smartphone camera captures an image of the product, which is processed using transfer learning techniques to classify the item into one of
51 predefined categories. The classification model is trained on a custom-built dataset containing 153,000 images of fruits and vegetables. To refine accuracy,

ensemble learning combines predictions from both models based on weighted averages. Additionally, a load cell integrated with an Arduino board measures the weight of the identified item, ensuring precise billing calculations via Firebase Real-time Database[11].

- Image Processing

The proposed system utilizes image-based recognition through deep learning-based regression and image retrieval techniques. Cameras mounted on shopping carts capture egocentric images, which are processed for product identification and localization. The image retrieval-based approach enhances accuracy using deep metric learning, retrieving visually similar images from a pre-stored dataset. Feature extraction methods, including VGG16, Fisher Vector encoding of SIFT features, and triplet networks, optimize classification performance. VGG16 extracts hierarchical features for robust image matching, Fisher Vector encoding refines spatial representation, and triplet networks improve latent feature embeddings. These techniques collectively address visual ambiguity, enhancing product recognition and localization efficiency in automated shopping systems[16].This paper involve image-based recognition techniques involving deep learning models such as VGG16, ResNet50, EfficientNet, and InceptionV3, which enhance classification efficiency. Image preprocessing steps include resizing, normalization, feature extraction, and augmentation techniques to improve recognition accuracy and robustness[19]. This system uses Preprocessing techniques such as histogram equalization and noise reduction refine input images for enhanced recognition. Feature visualization, including class activation mapping (CAM), provides interpretability by highlighting relevant areas within images. Probability distribution modeling via Softmax functions improves classification decisions, enhancing system reliability[26]. The proposed system utilizes OpenCV-based image processing algorithms for feature extraction and optimization. CNN architectures enhance hierarchical feature mapping, while edge detection techniques such as Sobel, Laplacian, and Canny operators refine object boundaries. Image augmentation methods, including normalization, contrast enhancement, and rotation, further improve generalization. Additionally, histogram equalization and morphological transformations mitigate noise interference, strengthening recognition accuracy[24].

4. Checklist Generation

The checklist is created based on barcode scanning, where the webcam detects the product and transmits its details to the shopping cart's central processing unit. Raspberry Pi serves as the computational hub, updating the displayed list in real time. The system allows users to remove or modify entries before proceeding to payment. This ensures accuracy in product selection and prevents billing errors[1].The smart trolley system incorporates RFID tags affixed to products, which are scanned by an RFID reader upon placement into the trolley. The scanned data is sent to the Arduino microcontroller, which processes the product information and updates the checklist displayed on an LCD screen. Additionally, the system synchronizes data with a web server for automated billing, reducing the need for manual price verification. The checklist dynamically updates as items are added or removed, providing users with an accurate record of their purchases[2].The proposed system employs YOLO-based object detection and load cell measurement to recognize products and record them automatically. A camera captures images of the item, which are analyzed using deep learning techniques to identify the product class. The detected item is then added to the checklist, which is displayed to the customer. Additionally, a load cell measures the item's weight, ensuring precise billing. The system allows users to modify their checklist through dedicated buttons for resetting, adding new items, and finalizing the purchase[5].The system records shopping activities using a mounted video camera, analyzing each frame to classify hand movements and product interactions. Deep learning models such as Faster R-CNN and YOLOv2segment frames into categories: No-Hand (NH), Empty-Hand (EH), and Holding-Item (HI). Shopping actions are identified using Dynamic Time Warping (DTW), enabling event recognition, including Placing, Removing, and Swapping items. The detected events contribute to checklist updates, ensuring a real-time reflection of the cart's contents[8]. The checklist is generated through shopping activity recognition using YOLOv2-based frame classification and action segmentation. A smartphone mounted on the shopping cart records checkout events, transmitting data to a cloud server where frames are classified into No-Hand, Empty-Hand, and Holding-Item categories. Action segmentation groups classified frames, allowing the system to detect shopping events such as placing, removing, or swapping items. The final checklist is updated via Google Firebase Cloud Messaging (FCM) and displayed on the mobile app in real time[9]. The system employs face recognition and barcode scanning for customer authentication upon store entry. When a customer picks an item from a shelf, weight sensors detect the change, triggering a nearby camera to capture the customer's image. The recorded data is processed using deep learning-based face recognition to associate the product with the respective customer. The identified item is added to the user's virtual checklist, which updates dynamically within the mobile application. If the customer returns the product to the shelf, the system registers the action and removes the item from the checklist[10]. The proposed system integrates MobileNetV2 and EfficientNetV2 deep learning models to classify items placed in the shopping cart. A mobile application captures images of the products using a smartphone camera, and the trained model predicts the item class. Additionally, load cells embedded in the cart measure the item's weight, which is processed by an Arduino microcontroller and transmitted via Bluetooth to the mobile app. Once the product is identified and its weight recorded, the system retrieves pricing details from a cloud-based Firebase real- time database and updates the checklist accordingly. The checklist dynamically adjusts as items are added or removed from the cart, ensuring accuracy in tracking purchases[11].

5. Billing Process

The smart shopping cart system employs a webcam that scans product barcodes upon placement in the cart. The Raspberry Pi processor processes the barcode data, retrieving item details and price from an integrated database. As users shop, an LCD display continuously updates their bill, allowing them to remove or modify purchases. Once shopping is completed, users can initiate payment via QR code or UPI, eliminating the need for manual cash transactions. The final bill is sent via email or SMS, ensuring secure digital transactions[1]. The system employs a camera mounted on the shopping cart to capture images of the selected products, which are processed using YOLOv2, a real-time object detection algorithm. The model classifies the items and determines their identity based on pre-trained datasets. The detected product's weight is then measured using a load cell, which transmits data to an Arduino microcontroller. The microcontroller calculates the total price based on predefined rates and updates the bill dynamically. Customers interact with a user interface that provides options to reset, finalize, or modify the billing details before checkout. The final amount is displayed on an LCD screen for verification, ensuring an efficient transaction process[5]. The smart trolley system integrates RFID technology for automatic product identification. Each item in the supermarket is tagged with an RFID label, which is scanned upon placement into the trolley. The trolley's RFID reader communicates with an Arduino Uno microcontroller, which processes product details such as name, price, and quantity. These details are displayed on an LCD screen, allowing users to track their purchases. Upon completing the shopping process, a submit button finalizes the bill, transmitting all collected data to a central cloud server via the NodeMCU module. The cloud server generates the final invoice, which is then synchronized with the customer's mobile application, providing a seamless checkout experience[6].The proposed billing process involves a mounted video camera that captures shopping activity within the cart.Deep learning models such as Faster R-CNN, YOLOv2, and YOLOv2-Tiny analyze each frame, classifying hand movements and item interactions.These models distinguish between No Hand (NH), Empty Hand (EH), and Holding Item (HI) frames, enabling the system to track purchases dynamically.The shopping timeline is segmented into action intervals, identifying events such as placing, removing, or swapping items.These detected actions update the billing system, which maintains a real-time list of purchased items.The system transmits shopping activity data to the cloud, where billing details are processed and finalized for checkout[8].The billing process in iCart is based on checkout event classification using YOLOv2 deep learning algorithms. A smartphone mounted on the shopping cart records a video of shopping activity, which is transmitted to the cloud server for analysis. The Linux-based server processes the video frames, segmenting actions such as item placement, removal, or swapping. Each detected item is added to the shopping list, updating the billing details displayed on the user's app.The final bill is generated in real time through cloud computation and presented for digital payment, minimizing checkout delays[9].The billing process begins with customer authentication at the store entrance via barcode scanning and face recognition. Cameras capture the user's image and match it with database records to ensure identity verification.As the customer shops, weight sensors on store shelves detect changes in weight when an item is picked up or replaced.A deep learning model processes captured images to associate product selections with the correct user.The system dynamically updates the cart by adding or removing items based on sensor feedback, displaying real- time billing details on a mobile application. Upon exiting the store, the total bill amount is deducted from the user's linked account, eliminating the need for manual checkout[10].

6. Price Display on LCD

The system utilizes Raspberry Pi as the central processor to control the LCD display, barcode scanner, and database management. When a user scans a product using a webcam, Raspberry Pi retrieves item details from a preloaded database and updates the total price dynamically. The LCD screen continuously refreshes to display added or removed items and their respective costs. The billing interface is designed to provide an intuitive user experience, allowing modifications before finalizing the payment[1].The billing system employs an RFID reader, which scans the RFID-tagged products when placed in the trolley.The microcontroller, Arduino Uno, processes the scanned item's details, retrieving product price and description from a pre-configured database.The total price is updated in real time and displayed on an LCD screen embedded within the trolley. Additionally, this information is synchronized with a web server via a Wi-Fi module, enabling remote billing updates and online tracking.This setup reduces the need for manual verification and ensures efficient transaction processing at checkout counters[2].The billing system employs an RFID reader, which scans the RFID-tagged products when placed in the trolley. The microcontroller, Arduino Uno, processes the scanned item's details, retrieving product price and description from a pre-configured database.The total price is updated in real time and displayed on an LCD screen embedded within the trolley. Additionally, this information is synchronized with a web server via a Wi-Fi module, enabling remote billing updates and online tracking.This setup reduces the need for manual verification and ensures efficient transaction processing at checkout counters[6].The system employs a Raspberry Pi- based processing unit integrated with a barcode scanner and cloud-based data retrieval. As users place items into the cart, the barcode scanner captures product details, transmitting them to a central processor.A database fetches the corresponding price and updates the total cost, which is then displayed on an LCD screen mounted on the shopping cart.The system dynamically adjusts the displayed price as new items are added or removed, ensuring real-time accuracy. The auditory feedback mechanism further assists visually impaired shoppers by announcing the updated total cost through a headset or speaker[14].The shopping cart is equipped with a mobile application that classifies grocery items using deep learning models such as MobileNetV2 and EfficientNetV2.The identified product's weight is measured using load cells connected to an Arduino microcontroller, which transmits weight data via a Bluetooth module.The application retrieves the item price from Firebase Real-time Database and calculates the total cost based on the weight.This updated price is then displayed on an LCD screen attached to the cart, ensuring real-time expense tracking.The system continuously updates the total bill as items are added or removed, enhancing checkout efficiency[11].

II. Conclusion

The integration of deep learning in image processing and visualization has significantly enhanced computational efficiency, accuracy, and user interactivity. AI-powered applications, from smart shopping systems to medical imaging, demonstrate improved structural retention, faster response times, and refined noise-reduction techniques. Comparative studies confirm substantial gains in performance over traditional methods, particularly in retail optimization, autonomous navigation, and industrial automation. The balance between precision and speed enables seamless real- time processing, bridging the gap between computational power and practical usability. As technology continues to evolve, deep learning-driven advancements will redefine visual data interpretation across multiple domains.

References

[1] Ramkumar S, Saravanan R, Kanagaraj Venusamy, Rani Jabbar N Jeevith .Smart "Shopping and Cart Billing System using Deep Learning" 2024 Proceedings of the Second International Conference on Intelligent Cyber Physical Systems and Internet of Things(ICoICI2024)IEEE

[2] Uganya G., Nisha A. S. A., Vinoth R., Rajah A. J. L., "Intelligent Smart Trolley System Using Arduino," *2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)*, pp. 1-5, Dec. 2023.

[3] Komnavarapu L., Mangu T., Macherla S. P., Tripathi B. M., "Smart Trolley for Quick Shopping," *2024 IEEE International Conference on Communication Technologies (IC2PCT)*, Vol. 5, pp. 141-144, Feb. 2024.

[4] Gorai K., Santosh S. V. S. C., Skanda S., AS V. M., Prajwala T. R., "Automated Shopping Cart: Reducing Long Queues One Cart at a Time," *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, pp. 1-8, Mar. 2023.

[5] N. Ragesh, B. Giridhar, D. Lingeshwaran, P. Siddharth, K. P. Peeyush, "Deep Learning Based Automated Billing Cart," *2019 International Conference on Communication and Signal Processing (ICCSP) - IEEE*, Chennai, India, 2019.

[6] Maurya S., Sahu G., Yadav A., Shukla B., Agrawal G., Kumar N., "IoT-Based Smart Shopping Trolley System," *2023 International Conference on IoT, Communication and Automation Technology (ICICAT) - IEEE*, 2023.

[7]R. Shetty, D. Pawar, R. Poojari, R. Patel, J. Mali, "Smart Shopping Cart using Machine Vision with Machine Learning," *AIP Conference Proceedings (Vol. 2857, No. 1) - AIP Publishing*, 2021.

[8] Hong-Chuan Chi, Muhammad Atif Sarwar, Yousef- Awwad Daraghmi, Kuan-Wen Liu, Tsı-Uı Ik, Yih-Lang Li, "Smart Self-Checkout Carts Based on Deep Learning for Shopping Activity Recognition," *2020 International Conference on Artificial Intelligence & Machine Learning(AIML)*, 2020.

[9] Muhammad Atif Sarwar, Yousef-Awwad Daraghmi, Kuan-Wen Liu, Hong-Chuan Chi, "Smart Shopping Carts Based on Mobile Computing and Deep Learning Cloud Services," *2020 IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, 2020.

[10] Narendra Shekokar, Anchal Kasat, Sanjay Jain, Pranjal Naringrekar, Monil Shah, "Shop and Go: An Innovative Approach Towards Shopping Using Deep Learning and Computer Vision," *2023 IEEE International Conference on Emerging Smart Computing and Informatics (ESCI)*, 2023.

[11] Kuntal Gorai, S V S C Santosh, Skanda S, Vijay Murugan A S, Prajwala T R, "Automated Shopping□ : Reducing Long Queues One Cart at a Time," *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, 2023.

[12] Joshua Cecil L., Sujitha Juliet, J. Anitha, Simy Mary Kurian, "Intelligent Location-Based Recommendation System for Enhancing Shopping Experience," *Proceedings of the 2024 International Conference on Cognitive Robotics and Intelligent Systems - IEEE*, 2024.

[13] S. Nithya Priya, G. Swadesh, K. M. Thirivikraman, Mohammed Vazeer Ali, M. Ranjith Kumar, "Autonomous Supermarket Robot Assistance Using Machine Learning," *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2021.

[14] Peter Zientara et al., "A Multitask Grocery Assistance System for the Visually Impaired," *IEEE Computer Society Conference on Computer Vision Pattern Recognition*, 2015.

[15] Mark P. Melegrito et al., "Abandoned-Cart-Vision: Abandoned Cart Detection Using a Deep Object Detection Approach in a Shopping Parking Space," *2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, 2021.

[16] Emiliano Spera et al., "Ego Centric Shopping Cart Localization," *24th International Conference on Pattern Recognition (ICPR)*, Beijing, China, 2018.

[17] Panutsaya Rujakom et al., "Retail Management on Mobile Application Using Product Classification," *2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2022.

[18] Dario Allegra et al., "Exploiting Egocentric Vision on Shopping Cart for Out-Of-Stock Detection in Retail Environments," *2021 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2021.

[19] Dharwadkar, R. Electronics, M. S., J. M. G., and P. Kannadaguli, "Smart Cart: Revolutionizing E-Commerce in India with AI-Powered Personalized Product Recommendations Overview," *IEEE 2024 5th International Conference on Circuits, Control, Communication and Computing (I4C)*, Bengaluru, India, 4-

[20] Ranjan S., Ponnalagu R. N., "An Intelligent Shopping Cart with Automatic Product Detection and Secure Payment System," *IEEE INDICON 2019*, 2019.

[21]M. Shalini, C. Sivasaran, and R. Kishore Harshan Kumar ,"Product Mapping Using Deep Learning in Supermarkets," *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) - IEEE*, DOI: 10.1109/ICSES60034.2023.10465534.

[22] Cust2Mate, "Smart Shopping Cart Automates Payments in Grocery Stores," *RFID Journal*, 2023.

[23] Yan ying Zhang and Xiaolin Zheng, "Development of Image Processing Based on Deep Learning Algorithm," *Proceedings of the 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, Dalian, China, Apr. 2022, doi: 10.1109/IPEC54454.2022.9777479.
.

[24] Weijun Wang, Junquan Wang, Zhiqiang Zhang, and Da Shi, "OpenCV Implementation of Image Processing Optimization Architecture of Deep Learning Algorithm Based on Big Data Processing Technology," *Proceedings of the 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS-2022) - IEEE*, ISBN: 978- 1-6654-7884-7, DOI: 10.1109/ICSCDS53736.2022.9760795.

[25]S. Deepan, Dr. Jaideep Gera, A. Pareek,S. Chinchmalatpure, G. S. Nayak, and Dr. J. Bodapati, "Deep Learning Approaches for Signal and Image Processing: State- of-the-Art and Future Directions," *Proceedings of the 2024 7th International Conference on Contemporary Computing and Informatics (IC3I) - IEEE*, DOI: 10.1109/IC3161595.2024.10829368.

[26]Y. Deng, Y. Huang, Z. Chen, and Y. Chen, "Image Processing and Visualization Technology Application of an Improved Deep Learning Algorithm," *Proceedings of the 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON) - IEEE*, DOI: 10.1109/NMITCON62075.2024.10699280.