ALLIANCE UNIVERSITY

PROGLINT'S COMPUTER VISION 2K23

NATIONAL HACKATHON

TITLE:: Hand Gesture Recognition using Overhead Cameras to Build Virtual Cart

CONTENT

01 OUR TEAM

02 INTRODUCTION

03 KEY FEATURES

04 SYTEM ARCHITECTURE

05 TECHNOLOGICAL STACKS

06 ALGORITHMS

07 PROJECT OBJECTIVES

08 USER ROLES

09 CUSTOMER WORKFLOW

10 EMPLOYEE WORKFLOW

11 MINIMUM RESOURCE REQUIREMENTS

12 SCALABILITY

OUR TEAM

Rohith Jeevanantham

B.Tech CSE Cybersecurity

Rupin Ajay

Team Lead

B.Tech CSE Cybersecurity

Sansita Karthikeyan

B.Tech AI & DS

Shiv Nadar University Chennai

INTRODUCTION

The Hand Gesture Recognition-Based Shopping System is an innovative and interactive shopping solution that leverages hand gesture recognition technology to enhance the shopping experience for customers. This document provides a comprehensive explanation of the project, its goals, components, and how it functions.

KEY FEATURES

HAND TRACKING

Enabling virtual shopping by tracking and interpreting hand gestures for item selection and management.

OBJECT DETCTION

Enhancing grocery store efficiency by identifying and tracking products for inventory management, shopping list generation, and customer assistance

VIRTUAL CART

Integrates object detection and hand tracking to select items, manage, and generate bills

SYSTEM ARCHITECTURE

1. **Object Detection (YOLOv3)**:
   - YOLOv3 is used for real-time object detection.
   - The model reads weights and configuration from files (**yolov3.weights** and **yolov3.cfg**).
   - It detects objects, primarily focusing on "bottle" objects, and provides their labels and confidences.
2. **Hand Landmark Detection (MediaPipe)**:
   - The program uses MediaPipe's hand landmark detection to track hand gestures.
   - It looks for specific landmarks on the hand to recognize gestures related to picking up and placing objects.
3. **Main Loop**:
   - The main loop continuously captures frames from a webcam (or other video source) using OpenCV.
   - It resizes the frames to a larger size for processing.

The position of a horizontal virtual line is calculated, which serves as a reference for object placement.

- The program performs object detection, checking for "bottle" objects.
- It also detects hand landmarks and analyzes them to recognize gestures, particularly the "Picking up object" gesture.

3. **Object Tracking:**
    - When a "bottle" object is detected, the program tracks its position and trajectory.
    - It determines if the object has crossed the virtual line.
    - It keeps track of whether the object is currently in hand and whether it has been picked up or placed back.
4. **Logging:**
    - The program logs actions to files (action_log.txt and cart.txt) for tracking object-related activities, including additions and removals from the virtual cart.
5. **Display:**
    - The processed frame is displayed in a window using OpenCV.
    - The frame includes visual information, such as the virtual line, detected object labels, bottle count, and cart statistics.

TECH STACKS

1. Python: The core programming language.
2. OpenCV: Handles computer vision, video capture, and object detection.
3. MediaPipe: Performs hand tracking and landmark detection.
4. YOLO: Deep learning model for real-time object detection.
5. PyTorch: Framework for running YOLOv5 model.
6. NumPy: For numerical operations and data manipulation.
7. Threading: Enables concurrent execution for hand tracking and object detection.
8. Time: Measures elapsed time.
9. Matplotlib: Used for creating trajectory plots.
10. Text Files: Logs actions and records cart items.
11. Machine Learning Models: Assumed use of machine learning models for object detection and gesture recognition.

ALGORITHMS

Certainly, here's a short and concise summary of the algorithms used:

1. **YOLOv3 (You Only Look Once, version 3):**
   - Detects objects in real-time by dividing images into a grid and predicting bounding boxes and class probabilities.
2. **MediaPipe Hand Landmark Detection:**
   - Tracks hand landmarks (fingertips, knuckles) for recognizing hand gestures and interactions.
3. **OpenCV (Open Source Computer Vision Library):**
   - Handles video capture, frame processing, drawing, and user interface for visualizing results.
4. **Matplotlib:**
   - Used to create a plot displaying the trajectory of picked-up objects.
5. **Action Logging:**
   - Records timestamps and actions to files for later analysis of user interactions.
6. **Gestures Recognition:**
   - Analyzes hand landmark positions to identify specific gestures, such as object pickup and placement.

PROJECT OBJECTIVES

The primary goal of this project is to create a seamless and efficient shopping experience using hand gesture recognition technology. The key objectives include:

1. **Enhanced Customer Experience**: Provide customers with a unique and user-friendly way to shop by recognizing their hand gestures to select and add items to their shopping cart.

2. **Real-time Cart Management**: Enable customers to manage their shopping carts in real-time, with items automatically added or removed based on their gestures.

3. **Efficient Checkout**: Simplify the checkout process by associating items with a customer's unique identifier (phone number) and generating bills accordingly.

4. **Employee Assistance**: Empower store employees with tools for managing inventory, tracking low-stock items, and assisting customers with any issues related to their shopping carts.

USER ROLES

The system caters to two types of users:

1. **Customer (CUST)**: Shoppers who use hand gestures to select and manage items in their cart.

2. **Employee**: Store employees who can manage inventory, assist customers, and oversee cart management.

CUSTOMER WORKFLOW

1. **Registration:** Customers enter their phone number as a unique identifier when starting their shopping session.

2. **Shopping Process:** Using hand gestures, customers select products they wish to purchase. These items are added to their virtual shopping cart by object detection models which are specifically trained.

3. **Cart Management:** The system automatically updates the cart's contents and total price as customers make hand gestures to add or remove items.

4. **Checkout Process:** To complete their purchase, customers input their phone number again, and the system generates a bill specific to that phone number.

EMPLOYEE WORKFLOW

1. **Login**: Employees log in to the system using their credentials to access their employee dashboard.

2.**Inventory Management**: Employees can view and manage product inventory, including product IDs, names, costs, and quantities. Low-stock items are highlighted for restocking.

3.**Cart Management**: Employees have the ability to assist customers with cart-related issues or inquiries.

MINIMUM RESOURCE REQUIREMENTS

1. **CPU:**
   - A modern dual-core CPU (e.g., Intel Core i3 or equivalent) should suffice for basic usage.
   - For smoother performance, especially when processing higher-resolution video or multiple camera feeds, consider a quad-core CPU (e.g., Intel Core i5 or equivalent).
2. **RAM**:
   - A minimum of 4 GB of RAM is recommended for running the code.
   - If you plan to work with high-definition video or process multiple camera streams, 8 GB or more is advisable for better performance.
3. **Storage:**
   - The code and its dependencies require minimal storage space (less than 1 GB).
   - Additional storage will be needed for recorded action logs and cart contents, depending on usage, but this should not be significant.
4. **Operating System**:
   - The code can run on Windows, Linux, or macOS.
5. **Dependencies:**
   - Ensure you have Python 3.x installed with the required libraries, including OpenCV, MediaPipe, and Matplotlib.

SCALABILITY FOR LARGER RETAIL STORES

Scalability for Large Retail Stores To deploy the system in a large retail store efficiently, follow these strategies:

1. **Distributed System**: Distribute computational load across multiple machines.

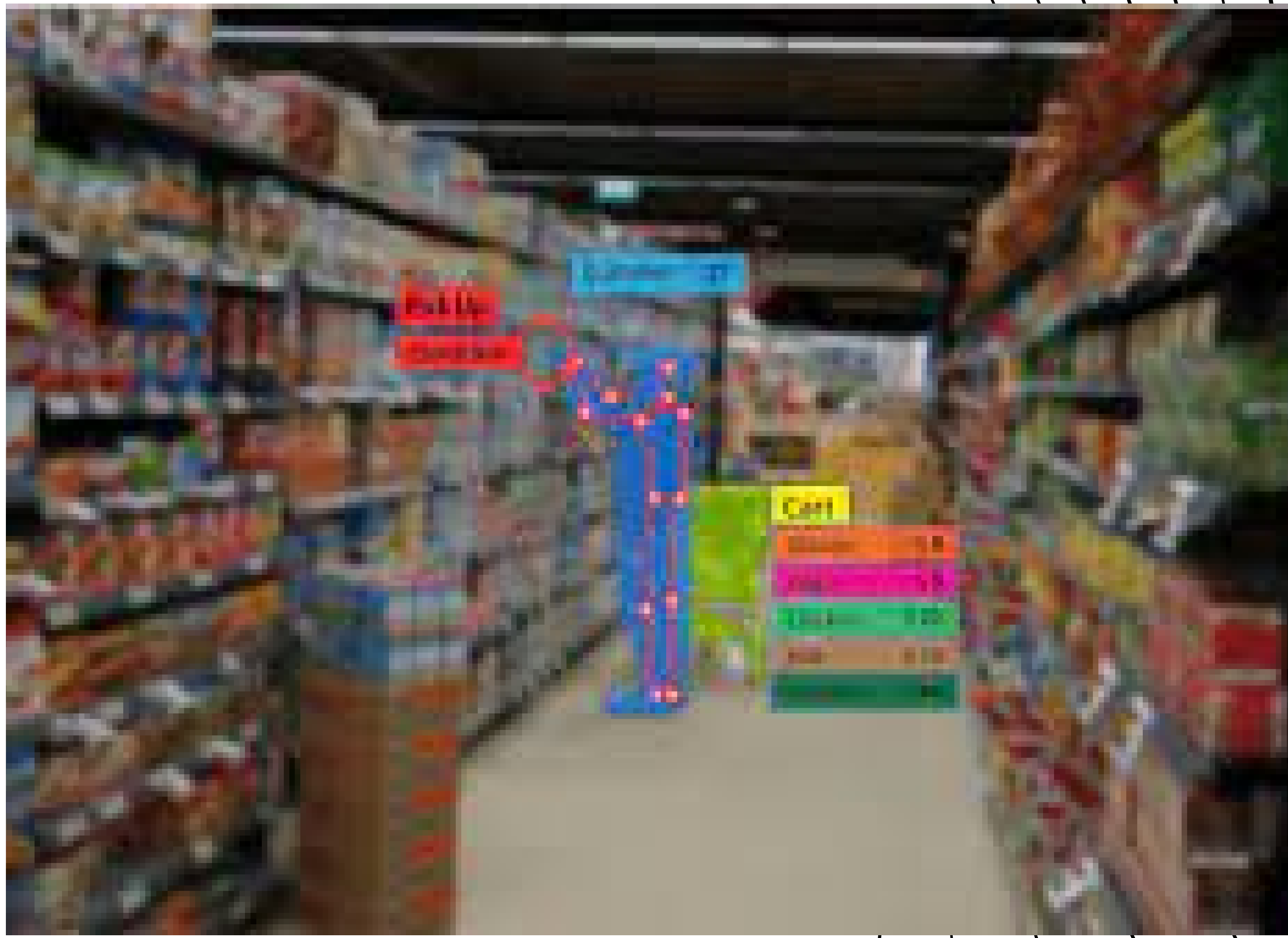
2. **Cloud Deployment:** Utilize cloud services for flexibility and resource scaling.

3. **Containerization**: Use Docker containers and Kubernetes for resource allocation.

4. **Resource Optimization:** Fine-tune resource allocation based on load testing.

CONCLUSION

The Hand Gesture Recognition-Based Shopping System redefines the shopping experience by integrating cutting-edge technology with efficient cart management and inventory control. This document provides an in-depth explanation of the project's architecture, functionality, and future potential, ensuring a clear understanding of its capabilities and significance.



THANK YOU

TEAM : ERR404