VisionSmart - AI-Powered Object Recognition and Interaction System

23.03.2025

Aiman Gohar
aimeegohar@gmail.com

VisionSmart: AI-Powered Object Recognition and Interaction System

1- Introduction

1.1 Overview of the Project

VisionSmart is an AI-powered object detection and interaction system that enhances user experience by identifying objects, detecting colors, and providing real-time assistance. Utilizing **YOLOv12**, the system can detect various room objects, including furniture, stationery, gadgets, appliances, and hazardous materials. With an integrated color detection module and advanced user interaction features, VisionSmart offers shopping assistance, and safety alerts.

1.2 Objectives

The primary goal of VisionSmart is to develop an intelligent object detection and interaction system that enhances user experience, improves safety, and facilitates smart decision-making. The specific objectives include:

1. Accurate Object Detection
    - Train and fine-tune YOLOv12 on a custom dataset to identify room objects, including furniture, stationery, gadgets, appliances, and hazardous materials.
2. Real-Time Color Identification
    - Utilize OpenCV's HSV color space to determine dominant object colors and detect background wall paint for home design applications.
3. Smart Shopping Assistance
    - Integrate web scraping and APIs (Amazon) to fetch product listings and provide users with real-time purchase recommendations.
4. Enhanced Safety Mechanism
    - Detect hazardous objects (e.g., knives, chemicals) and issue alerts through email, SMS, or smart home automation to prevent accidents.
5. Augmented Reality (AR) Integration
    - Overlay object details, pricing, and alternative products using AR visualization for an interactive experience.

1.3 Significance of the Study

VisionSmart is a system that enhances everyday interactions with objects through AI and automation. This study holds significant value due to its wide range of applications. One of its key benefits is enhanced user convenience, as it provides real-time object identification and smart shopping suggestions, streamlining the decision-making process. Additionally, improved safety measures are a crucial aspect, as the system can detect hazardous objects and trigger instant safety alerts, making homes and workplaces safer. Moreover, its augmented reality (AR) integration provides a visual and interactive experience, enabling users to view object details, and alternative options through an AR overlay.

Beyond personal use, VisionSmart has commercial and industrial applications, with potential implementations in retail stores, warehouses, and smart homes. By enhancing automation, security, and customer experience, this technology can revolutionize multiple industries, making AI-driven object detection an essential tool for the future

2- Methodology

2.1 Object Detection with YOLOv12

For object detection, a pre-trained YOLOv12 model is fine-tuned on a custom dataset containing furniture, stationery, gadgets, appliances, and hazardous objects. The dataset is annotated using Roboflow and converted to the YOLO format. The model is trained using Ultralytics YOLO framework, with optimizations such as data augmentation, and transfer learning to improve accuracy.

2.2 Color Detection using OpenCV

To identify object colors, images are processed using OpenCV's HSV color space. The image is converted from BGR to HSV, and dominant colors are extracted using K-Means clustering. For detecting wall paint color, a region without objects is selected, and the average pixel values are computed to determine the most prominent color.

2.3 Enhancing User Interaction

2.3.1 Shopping Assistance

Once an object is detected, the system queries e-commerce platform (Amazon) using web scraping (Selenium/BeautifulSoup) or APIs to fetch real-time product listings, prices, and

discounts. Users receive recommendations and links to purchase the detected item directly.

2.3.2 Danger Detection & Safety Alerts

If a hazardous object (e.g., a knife, cleaning chemicals, or sharp tools) is detected, the system issues a safety alert via an on-screen warning. If integrated with a smart home system, it can trigger preventive actions such as locking cabinets or drawers.

2.5 Augmented Reality Overlay Implementation

To enhance user experience, an AR interface is developed using OpenCV and AR.js/WebAR. This overlay displays object details, and alternative products in real time.

3- Implementation

3.1 Dataset Preparation and Preprocessing

The dataset for object detection is created by collecting images of commonly found bedroom objects for different demographics (babies, women, men, and elderly individuals). Initially, images were gathered using web scraping with Selenium from Google Images, but irrelevant images led to a shift toward manual data collection. Each object class contains at least five images, and the dataset comprises 115 object classes. Images are annotated using the Roboflow platform, where bounding boxes are manually drawn to label multiple objects in each image. The dataset is splitted into training (70%), validation (20%), and testing (10%) data and then converted into the YOLO format for training. Data augmentation techniques have been applied on the training data while training to overcome the disadvantage of a small dataset.

3.2 Fine-Tuning YOLOv12 for Object Detection

A pre-trained YOLOv12 model is fine-tuned on the custom dataset to enhance object detection accuracy. The model is trained using the Ultralytics YOLO framework on Google Colab with GPU acceleration. Initially, training was attempted with batch size = 16 and image size = 640, but the session crashed after the first epoch due to memory limitations. To resolve this, the settings are adjusted to batch size = 4 and image size = 512, ensuring stable training.

The training process included several optimizations:

- Data augmentation (flipping, copy_paste) to improve generalization.
- Transfer learning to leverage pretrained weights and accelerate convergence.
- Hyperparameter tuning, including batch size and learning rate adjustments.

The model's performance was evaluated using key metrics like mAP (mean Average Precision) and IoU (Intersection over Union) to ensure accurate and efficient object detection.

HYPERPARAMETER	VALUE	DESCRIPTION
epochs	20	Number of training epochs
batch	4	Batch size (reduced from 16 due to memory constraints)
imgsz	512	Image size reduced to 512 from 640 to overcome session crashing.
mosaic	1	Enables mosaic augmentation (combining four images)
mixup	1	Applying mixup augmentation (blending two images)
copy_paste	1	Enables copy-paste augmentation (copying objects into new images)
hsv_h	0.015	Hue augmentation factor
hsv_s	0.7	Saturation augmentation factor
hsv_v	0.4	Value (brightness) augmentation factor
fliplr	0.5	Probability of horizontal flipping
flipud	0.1	Probability of vertical flipping

3.3 Color Extraction and Classification

Object color detection is implemented using OpenCV's HSV color space. The process involves:

1. Converting images from BGR to HSV for better color representation.
2. Applying K-Means clustering to extract dominant colors from objects.
3. Mapping dominant colors to object labels (e.g., "Red Chair", "Blue Lamp").
4. For wall color detection, a background region with no objects is selected, and the average pixel values determine the dominant wall color.

3.4 Web Scraping for Shopping Assistance

To assist users in purchasing detected objects, web scraping with Selenium and BeautifulSoup was implemented. The system:

- Extracts product listings, prices, and discounts from Amazon.
- Filters and ranks results based on relevance, price, and customer ratings.
- Displays a buying link for users to purchase the detected object directly.
  For real-time data, integration with Amazon's API is also considered for fetching updated product details.

3.5 Hazardous Object Detection and Alert System

To improve safety, hazardous objects (e.g., knives, sharp tools, chemicals) are identified using the trained YOLOv12 model. Upon detection, the system triggers:

- On-screen warnings (e.g., "Caution! Knife detected!").
- Smart home integration, where detected threats can trigger automated actions (e.g., locking a drawer if a knife is detected).

3.6 AR-Based Object Information Display

To enhance interaction, an Augmented Reality (AR) overlay is developed using OpenCV and AR.js/WebAR. The system:

- Displays floating labels over detected objects with details like name, price, and alternative options.
- Enables users to interact with objects in real time through an AR interface.

4- Results and Analysis

4.1 Object Detection Performance Evaluation

The YOLOv12 model's performance is assessed using Mean Average Precision (mAP) and Intersection over Union (IoU) scores. The fine-tuned model achieved high accuracy in detecting

objects commonly found in bedrooms. The model effectively identifies multiple objects per image, showcasing robust bounding box precision.

4.2 Accuracy of Color Classification

Color detection is evaluated by comparing predicted colors with ground truth labels. Using OpenCV's HSV color space and K-Means clustering, the system accurately classifies dominant object colors. Minor inaccuracies occurred in low-light conditions or when objects had mixed colors.

4.3 Effectiveness of Shopping Assistance Feature

The web scraping module retrieves real-time product listings from platform, Amazon. The system successfully provides relevant product links, pricing, and discounts, improving shopping convenience. However, occasional mismatches occur when product descriptions are ambiguous or lack sufficient keywords.

4.4 Response Time of Safety Alerts

The hazardous object detection system triggers real-time alerts when knives, chemicals, or sharp objects are detected, ensuring prompt warnings. Performance is affected by network latency and object occlusion in some cases.

4.5 User Engagement with AR Interface

The AR overlay displays real-time object details, pricing, and alternative suggestions. However, performance depends on lighting conditions and camera calibration, affecting object positioning accuracy in AR mode.

4.6 Overall System Efficiency

The end-to-end system performs efficiently, balancing object detection accuracy, real-time processing, and user interaction. Future improvements could include optimizing model inference speed, reducing API response delays, and enhancing multi-object tracking accuracy.

5- Future Enhancements

5.1 Improving Object Detection Accuracy

To enhance detection accuracy, additional data collection efforts can be made to include a wider variety of household objects. Expanding the dataset with more images, diverse lighting

conditions, and occluded objects will improve the model's robustness. Further hyperparameter tuning, model pruning, and advanced augmentation techniques can also boost performance.

5.2 Expanding Product Recommendation Sources

Currently, shopping assistance is integrated with Amazon, but future improvements can involve scraping data from multiple e-commerce platforms such as eBay, Walmart, and AliExpress. This expansion will provide users with more competitive pricing, diverse product options, and localized shopping experiences.

5.3 Enhancing AR Experience with 3D Models

The current AR overlay provides text-based object details, but future iterations can integrate 3D models for a more immersive experience. By incorporating depth estimation, interactive object manipulation, and virtual try-on features, users can engage with objects in a more intuitive way.

5.4 Extending the System to Other Environments

At present, the system focuses on bedroom objects, but it can be extended to other home environments such as kitchens, living rooms, and offices. Future advancements may include real-time storage of previously detected objects to track missing items and improve household organization. A database system can be implemented to maintain logs of identified objects, allowing the system to recognize changes in the environment over time.

5.5 Database Management for Object Logs

Currently, the system does not store detected objects, but future advancements can integrate a database to maintain logs of previously identified items. This will enable features such as:

- Tracking Object Presence – The system can compare newly detected objects with previous logs to identify missing or newly added items.
- Personalized Insights – Users can receive reports on frequently used items or misplaced objects over time.
- Smart Alerts – If an object is detected in an unusual location or a hazardous item remains in a room for too long, the system can trigger notifications.
- Cloud or Local Storage Integration – Data storage can be implemented using SQLite for local storage or Firebase/AWS for cloud-based logging, ensuring real-time object history tracking across devices.

7- Conclusion

VisionSmart is an innovative AI-powered system that enhances everyday interactions by combining object detection, color recognition, shopping assistance, safety alerts, and augmented reality. By fine-tuning the YOLOv12 model on a custom dataset, the system effectively detects and classifies various household objects, ensuring accurate recognition. Additionally, OpenCV-based color detection improves the system's ability to provide descriptive object labels.

The shopping assistance feature simplifies purchasing decisions by fetching real-time product listings, while safety alerts help prevent potential hazards by identifying dangerous objects. The system also lays the groundwork for future enhancements, such as expanding product recommendations, improving AR with 3D models, and integrating a database for object logs to track missing or frequently used items.

Overall, VisionSmart demonstrates how AI and automation can enhance convenience, safety, and efficiency in both personal and commercial settings. With further advancements, it has the potential to become a comprehensive smart assistant for homes, retail spaces, and workplaces.

8- References

YOLOv12: https://github.com/sunsmarterjie/yolov12