

# AI Research Intern - Technical Assessment Task

## Overview

This task evaluates your ability to work with computer vision, machine learning, and retrieval-augmented generation (RAG) in a robotics context. You have **4 days** to complete this assignment.

## Task: Intelligent Object Recognition and Query System for Warehouse Robotics

### Scenario

You're developing a prototype system for a warehouse robot that needs to:

1. Identify and locate objects using its camera
2. Understand natural language queries about objects
3. Retrieve relevant technical documentation about handling specific items

### Required Deliverables

#### Part 1: Computer Vision Module (OpenCV)

**Objective:** Build an object detection and tracking system

#### Requirements:

- Create a Python script that processes images/video feed to detect common warehouse objects (boxes, packages, pallets)
- Implement at least ONE of the following:
  - Edge detection and contour analysis to identify package boundaries
  - Color-based segmentation to differentiate object types
  - Template matching or feature detection for specific object recognition
- Calculate and display:
  - Bounding boxes around detected objects
  - Approximate dimensions (in pixels)
  - Center coordinates of each object
- **Bonus:** Track objects across multiple frames in a video

#### Deliverable:

- Python code with clear comments
- Sample images/video showing input and output
- Brief explanation (200-300 words) of your approach

## **Part 2: Machine Learning Model (ML)**

**Objective:** Train a classifier to categorize objects

### **Requirements:**

- Use a small dataset (you can use public datasets like CIFAR-10, COCO subset, or create synthetic data)
- Train a model to classify objects into at least 3 categories relevant to warehousing (e.g., "fragile", "heavy", "hazardous")
- You may use:
  - Traditional ML (SVM, Random Forest with hand-crafted features)
  - Deep Learning (CNN with transfer learning recommended)
- Evaluate model performance with appropriate metrics
- Demonstrate inference on new images

### **Deliverable:**

- Training code with documentation
- Model weights or saved model file
- Performance report including:
  - Accuracy, precision, recall
  - Confusion matrix
  - Discussion of limitations (150-200 words)

## **Part 3: RAG System**

**Objective:** Build a retrieval system for robotics documentation

### **Requirements:**

- Create a small knowledge base (10-15 documents) containing:
  - Object handling instructions
  - Safety protocols
  - Equipment specifications
  - Troubleshooting guides
- Implement a RAG pipeline that:
  - Chunks and embeds documents
  - Retrieves relevant context based on user queries
  - Generates responses using retrieved information

### **Example queries your system should handle:**

- "How should the robot handle fragile items?"
- "What's the maximum weight capacity for gripper arm?"
- "What safety checks are needed before moving hazardous materials?"

**Deliverable:**

- RAG implementation code
- Sample documents used (can be synthetic)
- Demo showing 3-5 query-response examples
- Short write-up (200-300 words) explaining your architecture choices

**Part 4: Integration (Optional - Highly Valued)**

**Objective:** Connect the three components

Create a simple demo where:

1. The system detects an object via OpenCV
2. Classifies it using your ML model
3. Answers questions about how to handle it using RAG

**Example workflow:**

- Input: Image of a box with "FRAGILE" label
- System detects box, classifies as "fragile"
- User asks: "How should this be handled?"
- System retrieves and presents relevant safety protocols

## Submission Guidelines

**Required Files**

1. All source code (well-commented and organized)
2. README.md with:
  - Setup instructions
  - Dependencies (requirements.txt)
  - How to run each component
  - Challenges faced and how you solved them
3. Results folder with:
  - Sample outputs
  - Performance metrics
  - Screenshots/videos of your system working

**Submission**

- Push your code to a GitHub repository (public or provide access)
- Include a 5-minute video demo (optional but recommended)

**Note:** You are welcome to use ChatGPT or other AI agents to assist you with this task. However, you should be fully aware of the logic and implementation details, as you may be asked to explain your approach.