Computer Vision Engineer Assignment

Instructions: Please submit your answers in the most efficient manner possible. The solutions should ideally be compatible with x86_64 or ARM platforms. You are encouraged to use any tools or libraries necessary for implementation. Provide a GitHub link containing all solutions along with the dependencies. Attach screen recordings of all the tasks in the github link provided.

1. Custom Object Detection with Model Training from Scratch: Implement a complete object detection pipeline for a custom dataset. Your task is to build and train a detection model from scratch (no pre-trained weights) such as Faster R-CNN, or a custom CNN-based detector. Train on a custom dataset of at least 3-5 object classes (you can use publicly available datasets like PASCAL VOC subset, or create your own). Evaluate the models based on mAP (mean Average Precision), inference speed (FPS), and model size. Provide a detailed report including architecture design choices, data augmentation strategies, training methodology, results comparison, and a discussion on the trade-offs between accuracy and speed. Also attach videos/GIFs showing real-time detection results with the github link.

2. Automated Quality Inspection System for Manufacturing: Background: A manufacturing facility produces some components and needs automated visual inspection to detect defects before packaging. **Task:** Develop a computer vision solution to identify and classify defects on items of your choice. **Requirements:**

1. Choose any manufactured item with visible features (PCBs, fabric, printed materials, etc.) for your prototype.
2. Capture or source actual images containing at least 3 types of defects (scratches, misalignment, missing components, discoloration, etc.) along with defect-free samples.
3. Write a script that:
    1. Analyzes an input image of the product
    2. Detects and localizes defect regions (bounding boxes or segmentation masks)
    3. Classifies each defect type with confidence scores
    4. Outputs the (x, y) pixel coordinates of defect centers and severity assessment
4. Submit sample images (defective and non-defective) with annotations in the github repository

3. Custom VLM Design for Industrial Quality Inspection

Scenario: A semiconductor manufacturer needs an offline AI system for PCB inspection where inspectors ask natural language questions about defects and receive structured responses with locations and confidence scores (<2s inference). Available: 50,000 PCB images with defect bounding boxes (no QA pairs). Generic VLMs hallucinate.

Task: Given this scenario, design a custom VLM solution addressing:

(A) **Model Selection** - Which VLM would you choose (LLaVA, BLIP-2, Qwen-VL, or custom architecture) and why? What factors influence your choice (model size, architecture design, inference speed, fine-tuning flexibility, licensing)? What architectural modifications are needed for precise localization?

(B) **Design Strategy** - How would you design the VLM architecture to handle PCB-specific requirements? What components would you modify (vision encoder, language decoder, fusion mechanism)?

(C) **Optimization** - How would you optimize the VLM for <2s inference and offline deployment? What techniques would you use (quantization, pruning, distillation, LoRA)?

(D) **Hallucination Mitigation** - How would you tune the VLM to reduce hallucinations? What training strategies, loss functions, and architectural changes would minimize false information?

(E) **Training Plan** - Design a multi-stage training approach with QA pair generation strategy, data augmentation, and evaluation metrics.

(F) **Validation** - How would you validate counting accuracy, localization precision, and hallucination rates?

Deliverables: Design a document addressing the above questions