Crowd Counting, Object Detection, and Segmentation Model

Objective

Develop a system that:

- Counts the number of people boarding and exiting public transport vehicles in real-time.
- Maintains individual privacy by processing blurred images from entrance/exit cameras.
- Achieves accurate crowd density estimations using specialized models.

Data Sources

Datasets considered for training crowd counting and object detection models:

1. **UCF-QNRF Dataset**: High-resolution images ideal for crowd counting.
    a. **Image Count**: 1,535 images
    b. **Annotations**: 1.25 million person annotations
2. **ShanghaiTech Dataset**: Commonly used for crowd counting, split into Part A (dense crowds) and Part B (sparse crowds).
    a. **Image Count**: 1,198 images
    b. **Annotations**: Over 300,000 person annotations
3. **JHU-CROWD++ Dataset**: Large-scale crowd dataset.
    a. **Image Count**: 4,372 images
    b. **Annotations**: Over 1.5 million head annotations
4. **COCO Dataset**: Standard for object detection and segmentation tasks.

Models Considered

1. **CSRNet (Convolutional Neural Network)**: Suitable for high-density crowd counting through density map generation.
2. **YOLO / Faster R-CNN**: For detecting individual persons in less dense crowd conditions.
3. **Mask R-CNN**: For object segmentation and detailed scene analysis.

Cloud-based Model Training & Tuning

1. **Training**:
    a. Leverage **cloud resources** (e.g., AWS, GCP) to train large models using GPUs/TPUs.
    b. Utilize **distributed training** to reduce training time and enhance scalability.
2. **Hyperparameter Tuning**:
    a. Use cloud-based automated tuning tools (e.g., AWS SageMaker Tuning, GCP Hyperparameter Tuning) for efficient model optimization.
3. **Continuous Training & Updates**:
    a. Implement **CI/CD** for periodic retraining as new data arrives.
    b. Monitor and evaluate models post-deployment to ensure optimal performance.

Real-time Deployment on Edge Devices

1. **Edge-Cloud Integration**:
    a. Deploy models on edge devices installed in vehicles, enabling offline processing.
    b. Use cloud storage for centralized monitoring and data backup.
2. **Edge Device Data Handling**:
    a. For cases of **network loss** or **error occurrences**, store temporary data on the edge device.
    b. Use lightweight storage such as SSDs, with periodic cloud syncing for data persistence.
3. **Privacy Compliance**:
    a. Blur or anonymize input images to maintain passenger privacy.

Data Pipeline Workflow

1. **Data Collection**:
    a. Acquire data from pre-existing datasets or live camera feeds.
2. **Preprocessing**:
    a. Resize, normalize, and augment images to improve robustness.
    b. Generate **density maps** for crowd counting tasks.
3. **Model Training & Optimization**:
    a. Train crowd counting models with density maps and fine-tune object detection models for specific scenarios (e.g., entrance cameras).

    b. Optimize models (e.g., pruning, quantization) for real-time inference on edge devices.
4. **Real-time Inference**:
    a. Process camera feeds for passenger counting and segmentation in near real-time.
    b. Implement temporal smoothing to refine count consistency across frames.
5. **Post-Processing & Output**:
    a. Generate passenger counts, detect entry/exit events, and upload results periodically to cloud storage.