KTP recruitment technical challenge (0183-24)
Scene change detection report

Chintan Patel
pchints7@gmail.com

Introduction:

This project detects changes in objects between base and test image sequences.

Solution Overview:

The solution tackles the problem of detecting changes in objects between base and test images using YOLO models. A custom YOLO model is fine-tuned to detect specific objects in the base images (Stop Sign, Box, Traffic Cone). At the start script processes all base images to detect these objects, storing the results for later comparison. Non-Maximum Suppression (NMS) is applied to filter redundant detections.

For test images, detections are performed using both the fine-tuned (our custom model) and a pre-trained YOLO model. The pre-trained model helps detect any additional objects that might not be present in base (e.g. Person in current). Detections from both models (custom and pre-trained) are combined, ensuring no duplicates based on Intersection over Union (IoU). The change detection involves comparing objects in base and test images: objects not found in the same location are tagged as "appeared," and those missing in the test images are tagged as "disappeared." New objects not present in the base images are tagged as "unknown."

Rationale behind the design:

Classical computer vision techniques such as background subtraction, contour detection, as well as feature-based methods proved ineffective for our task due to their sensitivity to pixel-level differences in the simulated environment. These methods resulted in numerous false positives and failed to consistently identify objects accurately. To address these issues, we adopted a deep learning-based approach using YOLOv8. By training a custom YOLO model on our dataset, we achieved reliable detection of known objects. Additionally, for the unknown new object we employed a YOLOv8 pre-trained model.

We chose YOLOv8 over other models due to its superior balance between speed and accuracy, making it well-suited for real-time applications. To refine the detections, we implemented Non-Maximum Suppression (NMS) to eliminate redundant bounding boxes, and Intersection over Union (IoU) to measure the overlap between detected boxes. These techniques helped in reducing false positives and ensuring that each object is detected precisely once.

Future Improvements:

Given six months to improve the prototype, I would focus on expanding and diversifying the dataset (varied viewpoints, lighting conditions, and environmental scenarios) to enhance model robustness. Implementing self-supervised learning techniques like contrastive learning or clustering-based methods would allow identifying new objects not present in the initial training set. I would develop a heuristic function leveraging the GPS data to quantify object movements in test scenarios with respect to the base. In addition, optimizing the pipeline through model compression and acceleration techniques would ensure real-time performance on the NVIDIA Jetson AGX Orin. Extensive testing, including challenging edge cases in simulated and real-world environments, will need to be performed to validate the system's reliability and robustness before deployment.

Appendix: Custom YOLO Model Training

The custom YOLO model was fine-tuned on pre-trained YOLOv8 architecture with a diverse dataset comprising 56 training images and 14 validation images. These images were extracted from base (choosing every 25th frame and then manually discarding unnecessary images) and annotated using the 'labelimg' tool. The dataset included three classes: Stop Sign, Box, and Traffic Cone. Training parameters were carefully chosen to optimize performance, including 30 epochs, a batch size of 8, and an initial learning rate of 0.001 with the Adam optimizer. Augmentation techniques such as HSV adjustment, translation, scaling, and mosaic augmentation were employed to improve model robustness and generalization. The model gave an average accuracy of 0.98 mAP for a confidence score of 0.5.