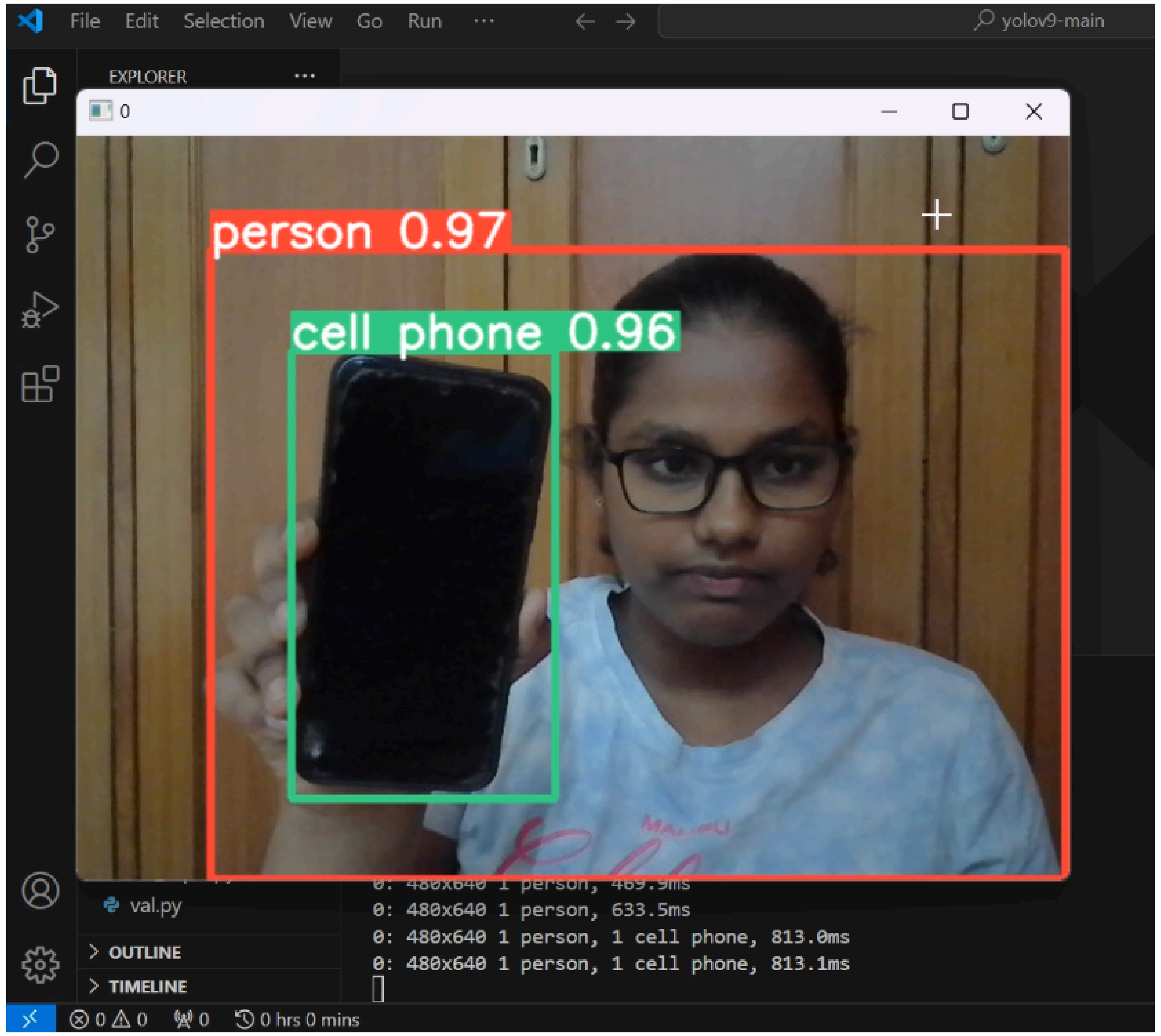
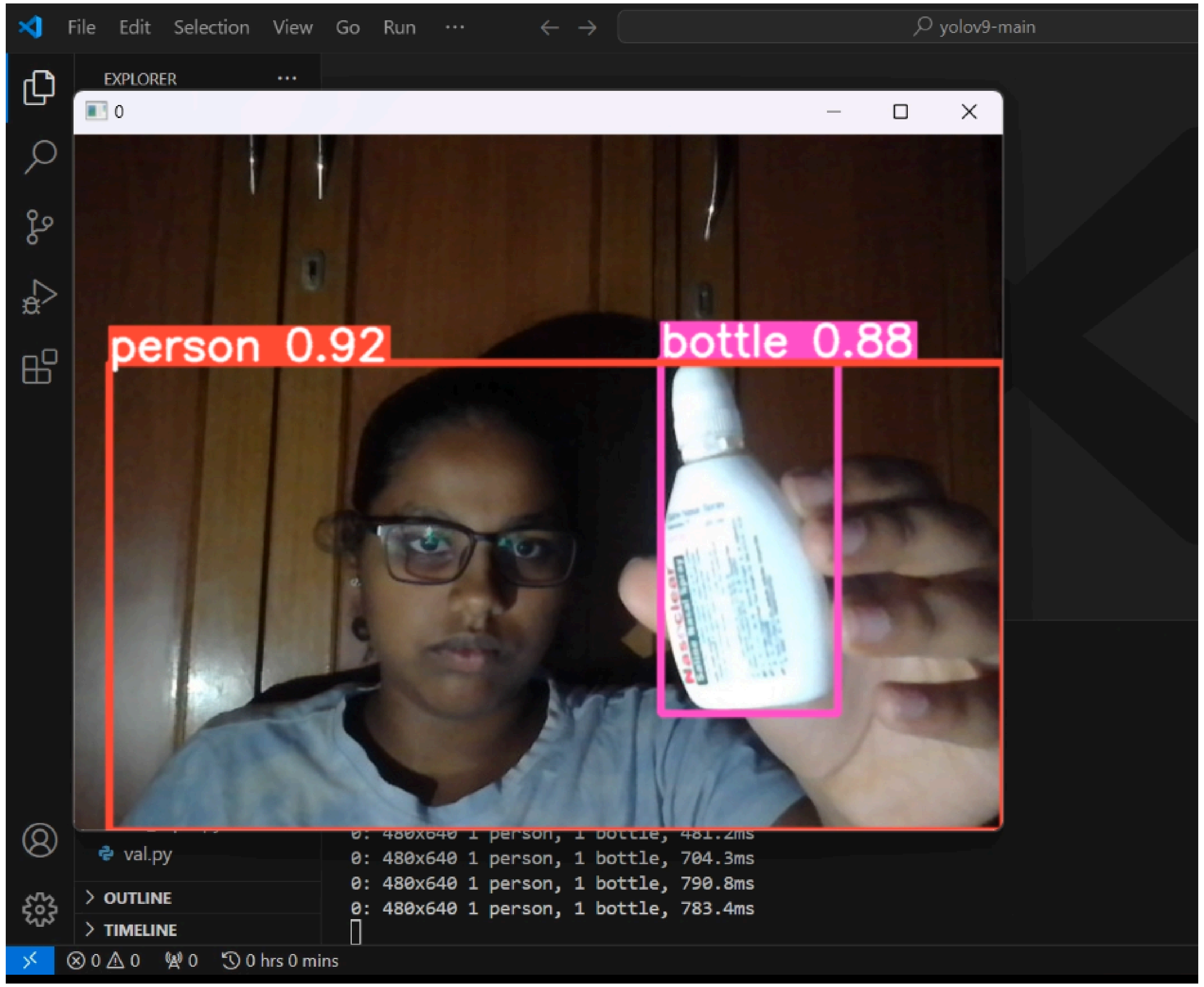
Training Pipeline for Real-Time, High-Precision Object Detection using YOLOv9

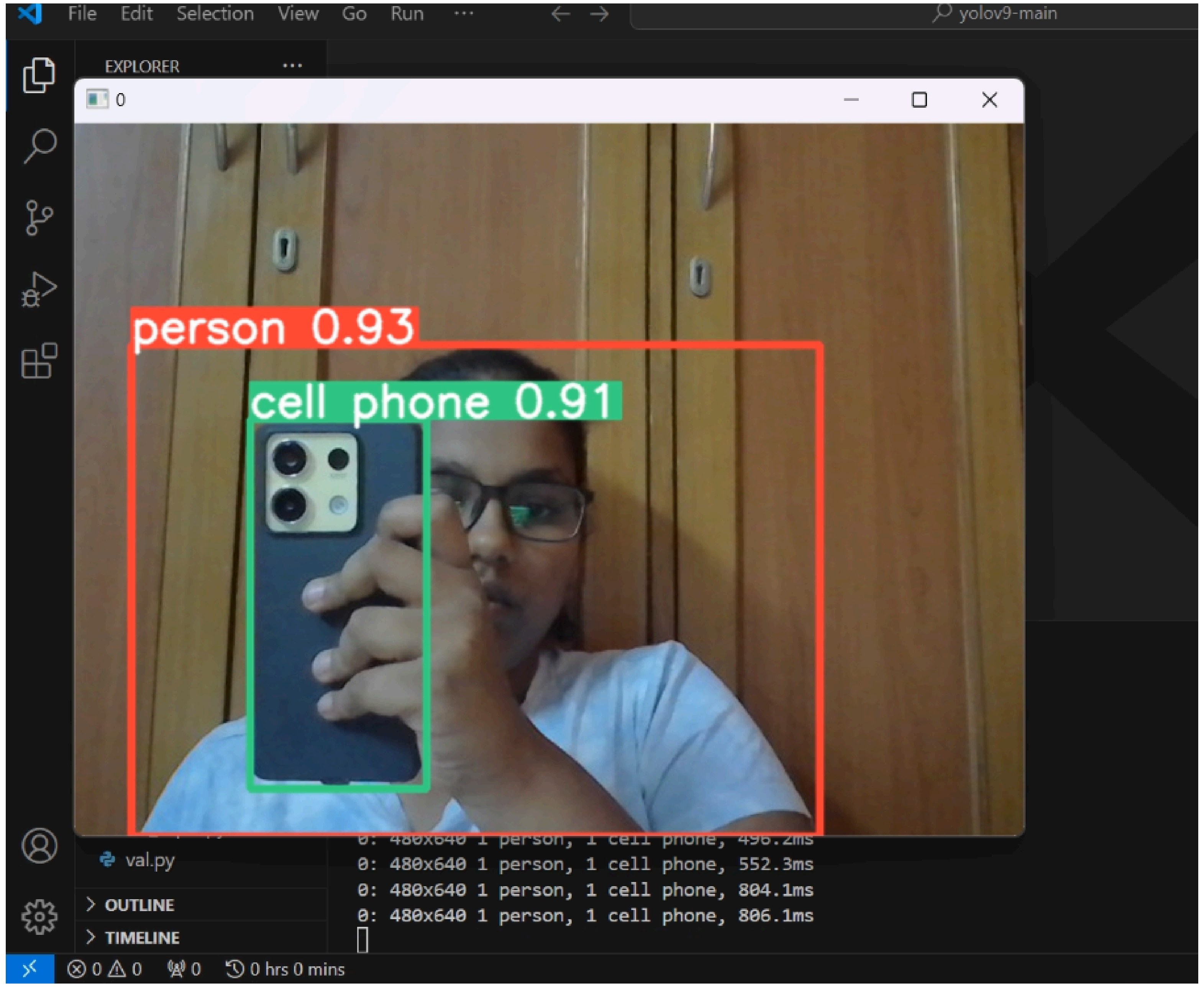


Sindhoora Raja,
National Public School - Indiranagar,
Bangalore

Accuracy in low lighting & tracking

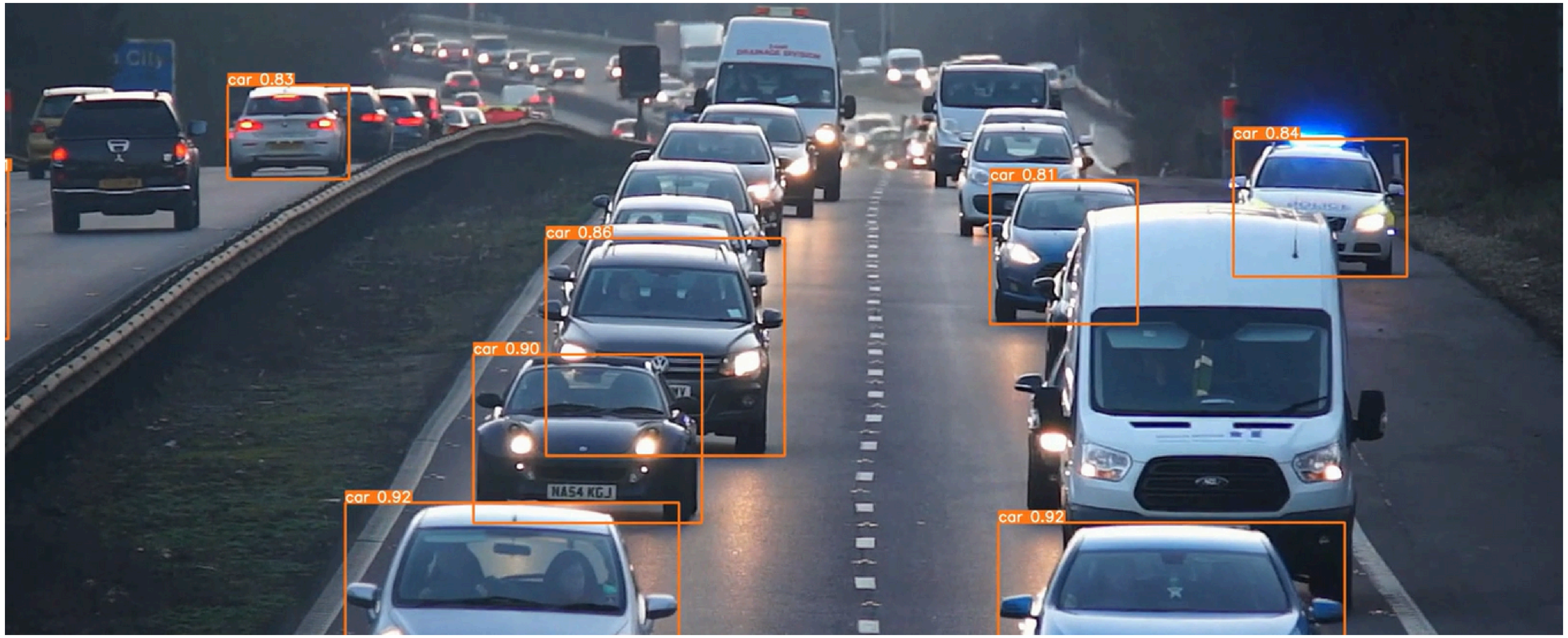


Low-light (please refer to the video)



Tracking (please refer to the video)

Video Sample



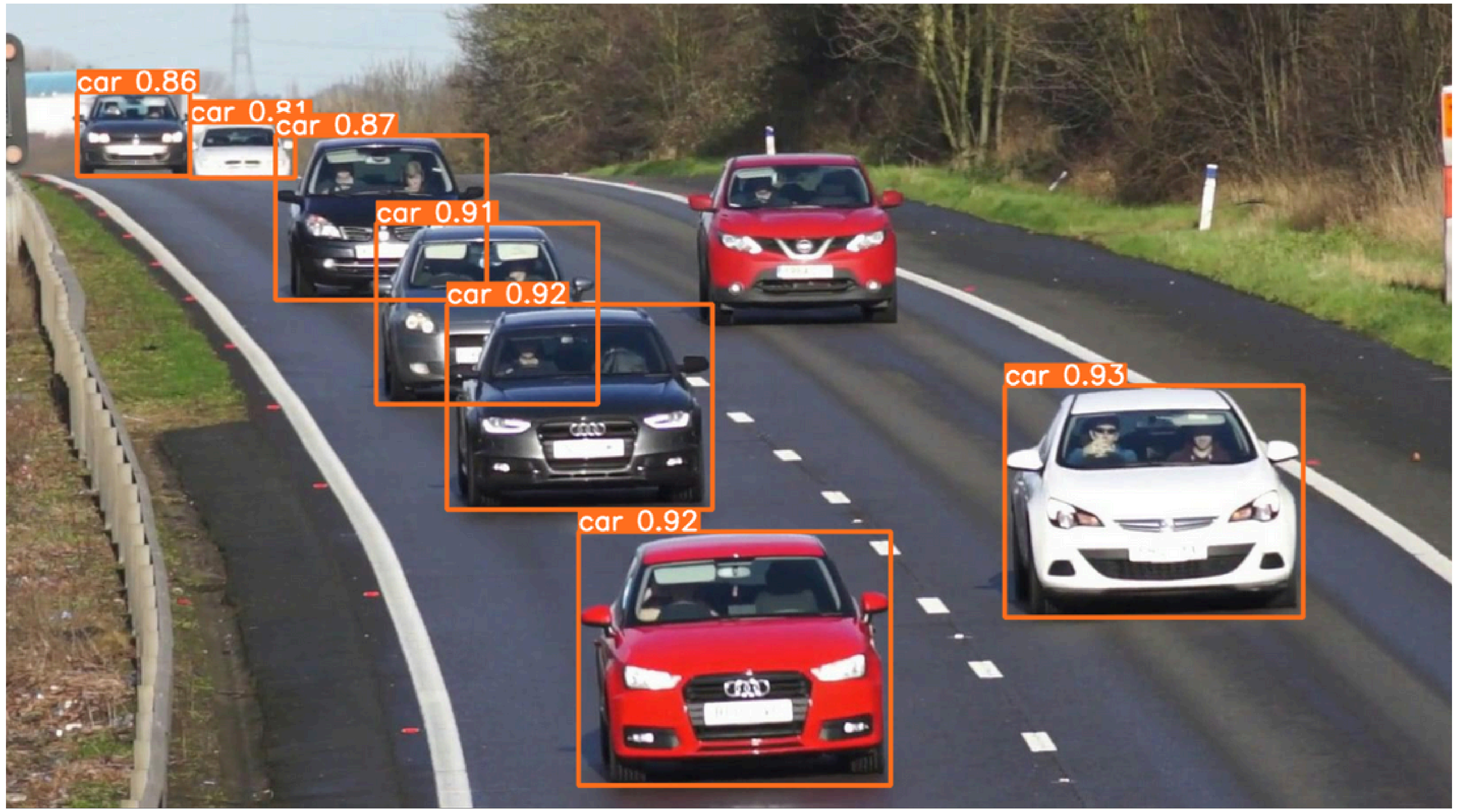
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
video 1/1 (39/1800) C:\Users\rajas\Documents\iit-b techfest\yolov9-main\vid.mp4: 384x640 10 cars, 497.1ms
video 1/1 (40/1800) C:\Users\rajas\Documents\iit-b techfest\yolov9-main\vid.mp4: 384x640 10 cars, 457.7ms
video 1/1 (41/1800) C:\Users\rajas\Documents\iit-b techfest\yolov9-main\vid.mp4: 384x640 10 cars, 448.8ms
video 1/1 (42/1800) C:\Users\rajas\Documents\iit-b techfest\yolov9-main\vid.mp4: 384x640 10 cars, 454.8ms
```

Individual frames are taken and processed (please refer to the demo)

Introduction

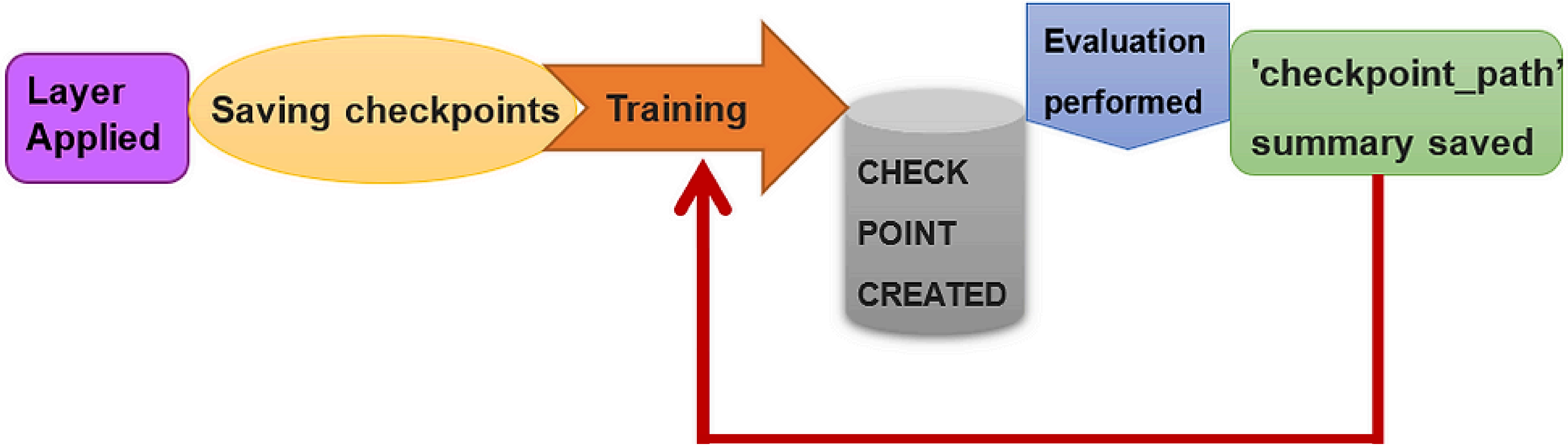
- This model is designed for real-time applications, requiring accuracy.
- The focus here is minimising false positives and negatives, which are important for applications like surveillance or autonomous driving. The model must process data in real-time, handle various environments (lighting changes, occlusions, etc.), and provide object tracking across multiple video frames for better situational awareness.



Ran on the present trained model

Environment Setup & Checkpoint Management

- Distributed training was implemented with communication enabled across GPUs, allowing parallel processing and faster training.
- GPU acceleration is critical for real-time model training and inference. However, the model supports both CPU and GPU setups, but scales best with GPUs for large-scale applications.
- The program uses a checkpoint strategy that includes saving the latest, best-performing, and periodic checkpoints to ensure minimal loss of progress and stripping implemented reduces file size which is rather useful for cloud storage and deployment.



Model

- The YOLOv9 model is used in the context of this project
- The system supports loading pre-trained models for transfer learning or initializing new models and specific layers can be frozen based on user-defined indices to optimize fine-tuning. The batch statistics are synchronizes across GPUs, improving model stability in distributed setups.
- Various optimizers (SGD, Adam, AdamW, LION) are available to adapt to different model requirements and learning rate schedulers (linear, cosine, flat cosine) help manage learning rate adjustments for smoother convergence. Learning rate and momentum warmup strategies accelerate early training, reducing initial instability.
- Exponential Moving Average (EMA) is also used to smooth model weights, enhancing stability during inference and improving overall performance and these EMA updates are applied at each iteration to maintain smooth transitions in model learning.

Summary

This presentation covers the setup and training of a high-precision, real-time object detection YOLOv9 model tailored for applications in surveillance, autonomous driving, and retail automation. It includes an environment setup for distributed GPU training, organized checkpoint management, and flexible hyperparameter loading. Key features involve logging and callbacks to track training, data loading with augmentation for robustness, and model handling with options for layer freezing and distributed SyncBatchNom. Optimizer and scheduler configurations, along with Exponential Moving Average (EMA), enhance training stability and performance. The training loop incorporates gradient clipping and mixed precision, while validation ensures model quality with metrics like precision, recall, and mAP. Additional features include early stopping, command-line argument parsing, and a main entry point to streamline the process, with potential future improvements aimed at enhancing tracking and optimizing for edge deployment.