# BMC-Artemis

## Overview

This project implements a vehicle counting system using computer vision techniques. It processes video inputs to detect, track, and count vehicles across multiple camera feeds.

## Docker Image Build and Execution

### Build the Docker Image

1. Ensure Docker is installed on your system.
2. Clone this repository and navigate to the project directory.
3. Build the Docker image: docker build -t bmc-artemis .

### Push to Docker Hub (Optional)

1. Tag the image: docker tag bmc-artemis your-dockerhub-username/bmc-artemis:latest
2. Login to Docker Hub: docker login
3. Push the image: docker push your-dockerhub-username/bmc-artemis:latest

### Execute the Docker Image

Run the Docker container with the following command: docker run --rm --runtime=nvidia --gpus all -v /host/path/to/input:/app/input -v /host/path/to/output:/app/output bmc-artemis /app/input/input\_file.json /app/output/output\_file.json

Replace /host/path/to/input and /host/path/to/output with the actual paths on your host system.

## Scripts and Their Descriptions

* app.py: Main entry point for the application.
* counting.py: Implements the vehicle counting logic.
* dataloader.py: Handles data loading and preprocessing.
* distances.py: Calculates distances between tracked objects.
* hausdorff\_dist.py: Implements Hausdorff distance calculation.
* prediction.py: Handles predictions for vehicle detection.
* st\_gen.py: Generates spatiotemporal features.
* tracking.py: Implements object tracking algorithms.

## Requirements

See requirements.txt for a full list of Python dependencies and their versions.

Key libraries include:

* PyTorch (2.2.2)
* OpenCV
* NumPy
* boxmot==10.0.78
* ultralytics==8.2.81
* numba==0.60.0
* ncps==1.0.1
* torch-geometric==2.5.3

## Open-Source Models

This project uses the following open-source models:

* YOLOv10x
* RTDETR100
* BoTSORT (clip-vehicleid weights)

## System Requirements

To run this code, the following specifications are recommended:

* CPU: Core i9
* GPU: NVIDIA RTX 4090
* RAM: 64GB
* Storage: 500GB SATA

Note: These specifications match the evaluation workstation for fair comparison.

## Directory Structure

* cam\_configs/: Camera configuration files
* cam\_masks/: Mask files for each camera
* prediction\_models/: Stored prediction models
* runs/: Output directory for runs
* tracking\_models/: Stored tracking models

## Evaluation

This project will be evaluated on a workstation with the following specifications:

* CPU: Core i9
* GPU: RTX 4090
* RAM: 64GB
* Storage: 500GB SATA

Ensure that your code is optimized to run efficiently on this hardware configuration.