

***Digital Twin Technologies Towards Understanding
the Interactions between Transportation and other
Civil Infrastructure Systems: Phase 2***

Vehicle Detection and Tracking on Edge Device
Document

*Sensors and Sensor Algorithms Relating to the Implementation
Process of the Project's Detection Technology*

PI: Ruimin Ke, Ph.D.

The University of Texas at El Paso
0000-0001-9139-6765

Co-PI: Jeffrey Weidner, Ph.D.

University of Texas at El Paso
0000-0003-0424-665X

Co-PI: Adeeba Raheem, Ph.D.

University of Texas at El Paso
0000-0002-2226-5582

Co-PI: Ruey Long Cheu, Ph.D.

University of Texas at El Paso
0000-0002-0791-2972

Vehicle Detection and Tracking on Edge Device

The process is implemented based on the Nvidia DeepStream SDK [1] which provides a complete pipeline for developing computer vision algorithms and deploying them in edge computing devices. Developers only need to customize the content of each module, such as setting the object detection algorithm to yolov5, DeepStream SDK will complete a series of tasks from image capture to output results, which greatly improves the development efficiency of developers. Figure 1 is the complete flowchart of DeepStream SDK.

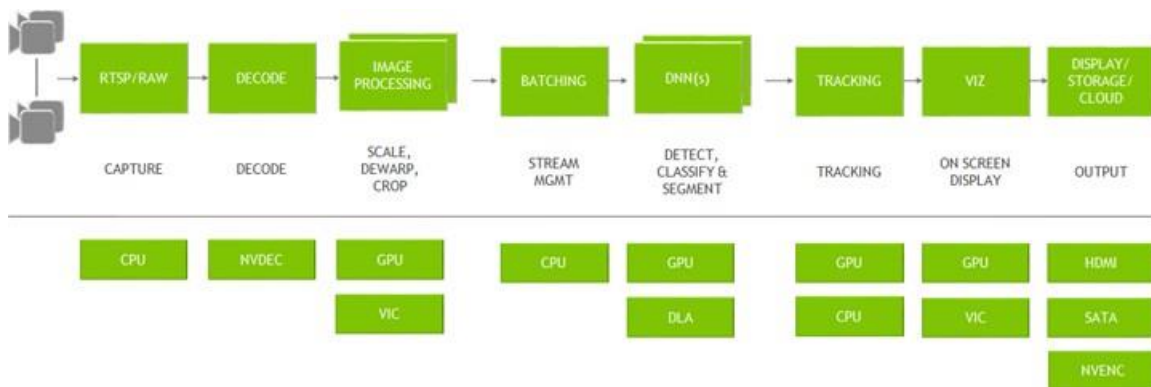


Figure. 1 Flowchart of DeepStream SDK [1]

The Nvidia TAO Toolkit was originally the Nvidia Transfer Learning Toolkit, its main function is to customize the DNN module in the DeepStream SDK and give it different functions. Its input is pre-trained model provided by Nvidia from Nvidia NGC (<https://catalog.ngc.nvidia.com/>), and the developer's customized data. The TAO Toolkit [2] performs transfer learning to improve accuracy and generates models that can be deployed on edge computing devices with the Nvidia DeepStream SDK. Figure 2 is the flowchart of TAO Toolkit.

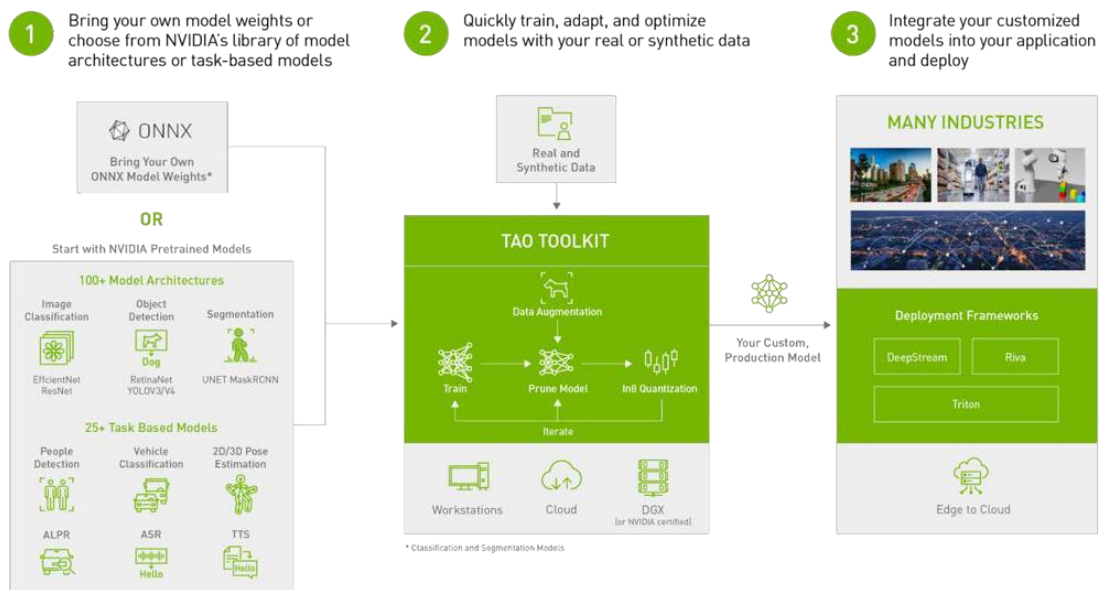


Figure. 2 Flowchart of TAO Toolkit [2]

The purpose of this project is to place edge computing devices Jetson Xavier NX and cameras on the fourth and fifth floors of the Schuster Garage on the UTEP campus, then use Yolov5 [3] and DeepSORT [4] to complete object detection and tracking tasks at the roundabout near the UTEP campus and get real-time traffic flow information in all directions including the trajectory of each vehicle. Among them, Yolov5 is responsible for the target detection task, and DeepSORT is responsible for the tracking task.

The project is currently in the stage of testing target detection and tracking algorithms and counting the number of vehicles for every type. Some videos shot from Schuster Garage at different times and from different angles are being tested using yolov5 and DeepSORT algorithms. Algorithms for calculating traffic flow in each direction have not yet been implemented.

Some issues have been observed in the experimentation. The first is the problem of detection accuracy. For example, some cars will be detected as trucks. The current solution is to use the Nvidia TAO Toolkit to further train the model to improve the detection accuracy. The second problem is that there is a parking lot next to the roundabout, and many stationary vehicles are also detected. The current solution to this problem is to use a background subtraction algorithm to classify stationary vehicles as background. The third problem is that vehicles traveling on the I-10 highway far away from the roundabout will also be detected. Our solution to this problem is that vehicles whose detection area is outside the area will not be logged. The last problem is that there is an obstruction in the middle of the roundabout, which will partially occlude the vehicle, so the tracking algorithm will lose the target and cause the same vehicle to have different IDs. The tentative plan is to use vehicle recognition and image smoothing technology based on [5].

As for the output, the original data could only be displayed on the monitor, and after some modifications, the data has now been stored in the data.txt file. Now the process is first real-time data will be written to internal_memory.txt and running client.py will write real-time data to data.txt.

Prerequisites and Device Environment

- Jetson Xavier NX (Advantech MIC-AILX-00B1)
- Jetpack = 4.6
- Keyboard
- USB camera
- Mouse
- Monitor

The original method was obtained from a github repository [6], and changes were made to log the data.

Implementation Process:

1. Download images
`$ docker pull gjy123456/cv-detect-robot:CDR-jetson-v4.18`
2. Give docker permission to display the graphical interface on the display
`$ sudo apt-get install x11-xserver-utils`
`$ xhost +`

3. Start container

```
$ sudo docker run -it --net=host --device=/dev/video0 -e QT_X11_NO_MITSHM=1 --  
gpus ""device=0"" --name="guo18" --privileged=true -v /tmp/.X11-unix:/tmp/.X11-unix -e  
DISPLAY=$DISPLAY -w /opt/nvidia/deepstream/deepstream-6.0 gjy123456/cv-detect-  
robot:CDR-jetson-v4.18 /bin/bash
```

4. Enter the specified directory

```
$ cd /opt/nvidia/deepstream/deepstream/sources/deepstream_python_apps/deepstream-  
yolov5/
```

5. Perform detection and tracking tasks. The tracking algorithm is DeepSORT.

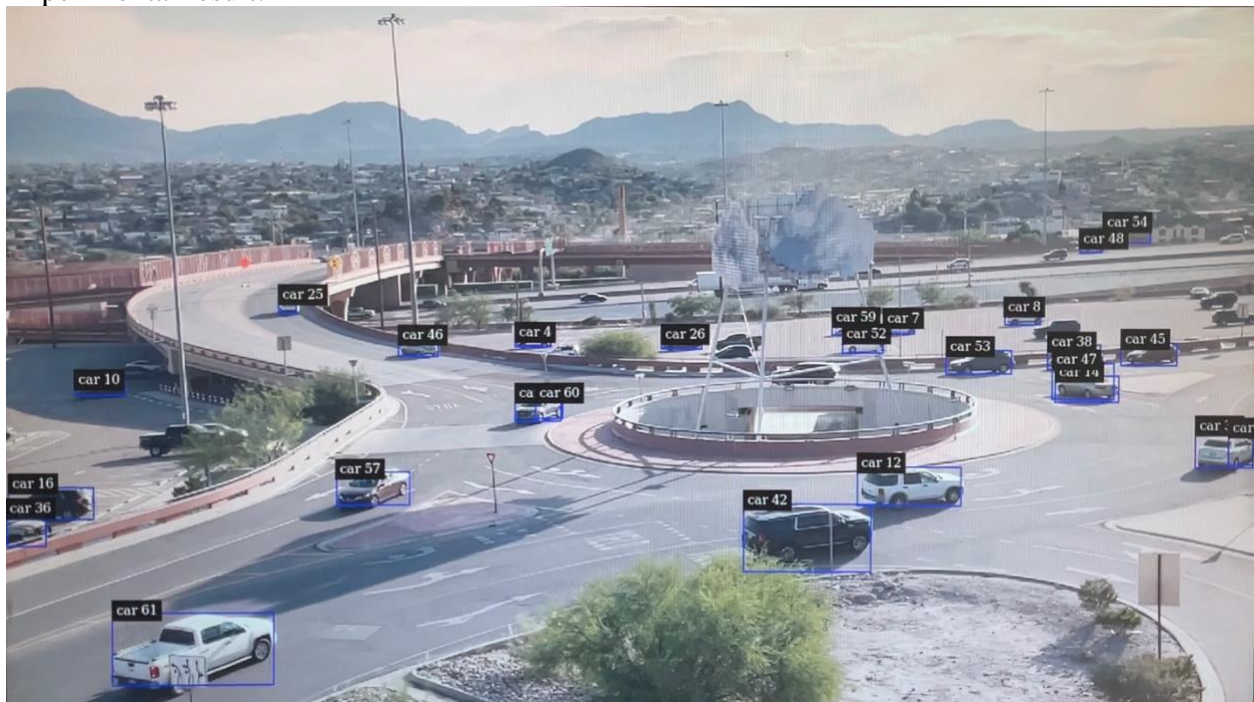
5.1 Detection and tracking tasks on video files

```
$ python3 deepstream_yolov5_file.py file:///opt/nvidia/deepstream/deepstream-  
6.0/samples/streams/sample_qHD.h
```

5.2 Detection and tracking tasks on usb camera live video

```
$ python3 deepstream_yolov5_cam.py /dev/video0
```

Experimental result:



Some useful links:

<https://developer.nvidia.com/deepstream-sdk>

https://docs.nvidia.com/metropolis/deepstream/dev-guide/text/DS_Overview.html#

<https://docs.nvidia.com/tao/tao-toolkit/text/overview.html>

<https://catalog.ngc.nvidia.com/>

References:

1. Nvidia DeepStream SDK. NVIDIA Developer. (August 25, 2022). Retrieved August 30, 2022, from <https://developer.nvidia.com/deepstream-sdk>

2. TAO Toolkit, NVIDIA, (Jul 13, 2022), Retrieved August 30, 2022, from <https://docs.nvidia.com/tao/tao-toolkit/text/overview.html>
3. Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Kalen Michael, Jiacong Fang, imyhxy, Lorna, Colin Wong, Zeng Yifu, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar (2022). ultralytics/yolov5: v6.2 - YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations (v6.2). Zenodo. <https://doi.org/10.5281/zenodo.7002879>
4. Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In 2017 IEEE international conference on image processing (ICIP) (pp. 3645-3649). IEEE.
5. Yang, Z. M., & Kong, C. C. (2020, December). A Video Based Method of Occlusion Vehicle Completion. In 2020 IEEE 6th International Conference on Computer and Communications (ICCC) (pp. 1368-1372). IEEE.
6. Guo, J. (2022). cv-detect-robot (Version CDR-v4.0) [Computer software]. https://github.com/guojianyang/cv-detect-robot/tree/main/CDR-docker_main_file/deepstream-yolov5