Real-time Object Detection for Autonomous Vehicles Using Intel oneAPI Toolkit

Team NextIn

Team Members:- Hans Upadhyay (Team Leader)
Aishwarya Upadhyay

Guided By:- Dr. Smita Rath (Assistant Professor, CSIT)

Contents

PROBLEM STATEMENT

- The challenge is to develop a real-time object detection model for autonomous vehicles using computer vision techniques and Intel® AI Analytics Toolkits and its libraries, that can accurately detect objects such as pedestrians, vehicles, traffic signs, and traffic signals. The model should perform with high accuracy and low latency, ensuring safe navigation for autonomous vehicles.

- Testing the model on a dataset that includes real-world scenarios, including various weather conditions, lighting conditions, and road environments.

- Provide a comprehensive report detailing the approach, methodology, results, and challenges faced during the development and testing of the model.

Core Components of oneAPI Used

Intel® Extension for PyTorch

Performed the PyTorch optimization of the model which accelerated computation and improved performance in training and inference

Intel® Extension for TensorFlow

Intel® TensorFlow offered an optimized implementation of the YOLO-NAS Algorithm, enabling efficient and fast object detection.

Intel® Distribution for Python

Achieved high-performance numerical and scientific computing.

Intel® Developer Cloud

Tested workloads across Intel® CPUs and GPUs. Intel® devCloud was the prime virtual environment for this prototype development.

YOLO (You Only Look Once)

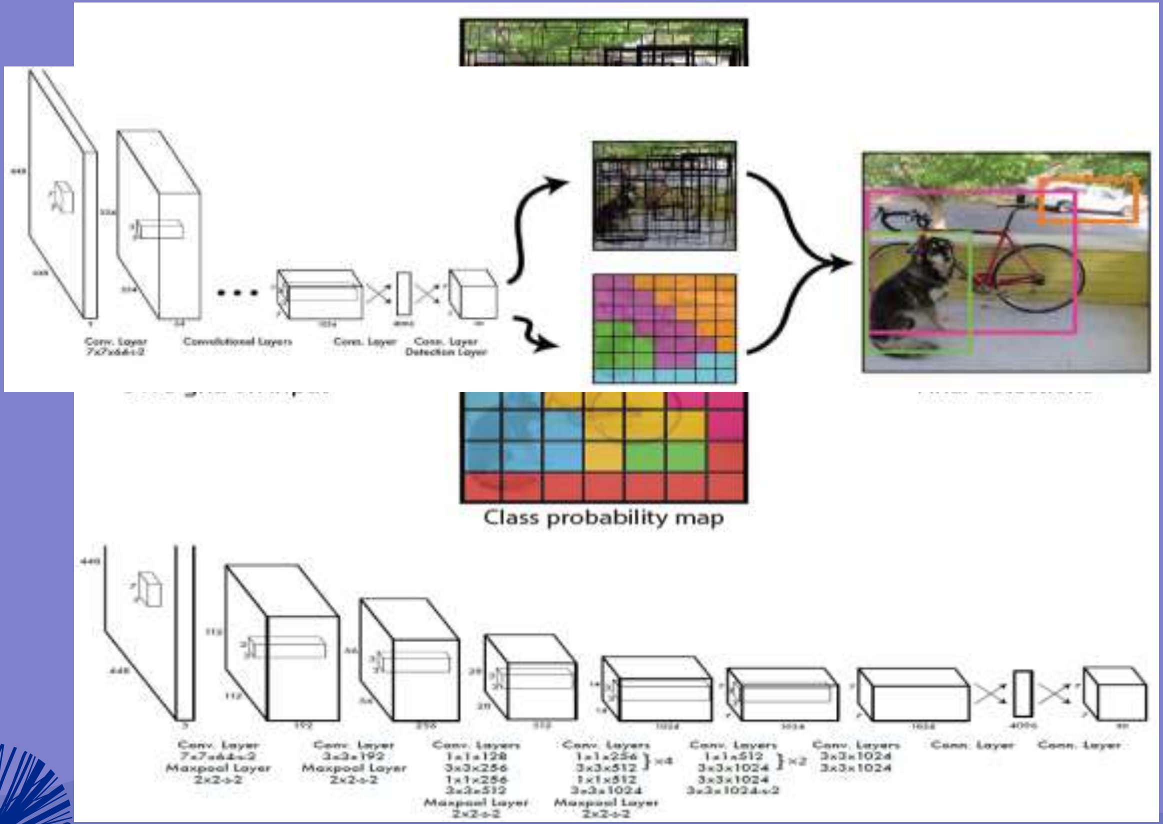
Single end-to-end CNN

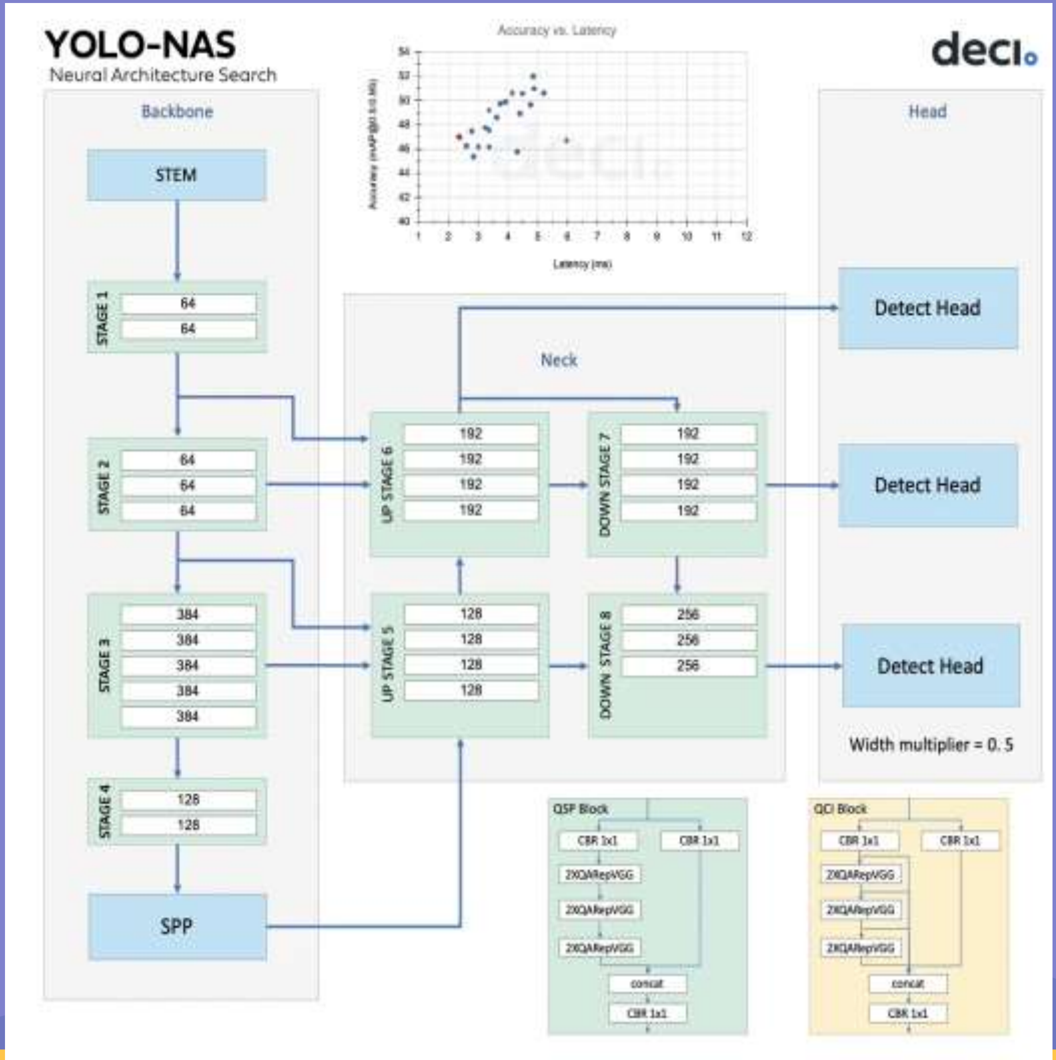
Object Detection as single regression problem

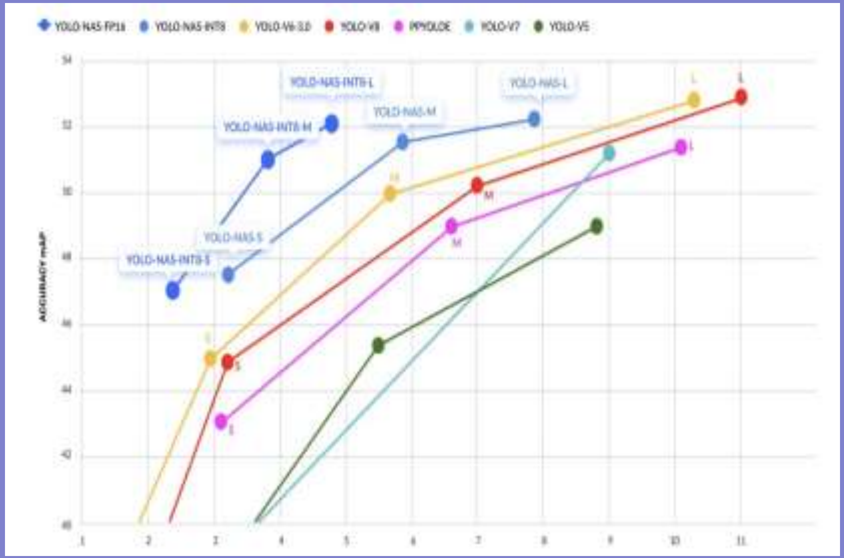
Divides image into S x S grid cells and predicts for each cell:

- B(=2) bounding boxes with 4 coordinates and confidence score
- C class probabilities

Map object to grid cell containing center of object







MODEL*	PRECISION*	mAP^{val*} 0.5:0.95	LATENCY* BS=1 (ms)	PARAMS (M)
YOLO-NAS S	FP16	47.5	3.21	19.0
	INT-8	47.03 (-0.47)	2.36 (+0.85)	
YOLO-NAS M	FP16	51.55	5.85	51.1
	INT-8	51.0 (-0.55)	3.78 (+2.07)	
YOLO-NAS L	FP16	52.22	7.87	66.9
	INT-8	52.1 (-0.12)	4.78 (+3.09)	

IMPORTANT LINKS FOR REFERENCE

- GITHUB REPOSITORY LINK

https://github.com/hansupadhyay007/intel-oneAPI

- LIVE VIDEO/DEMO OF THE PROJECT LINK ON YOUTUBE

https://youtu.be/EqWxqZqPX1M

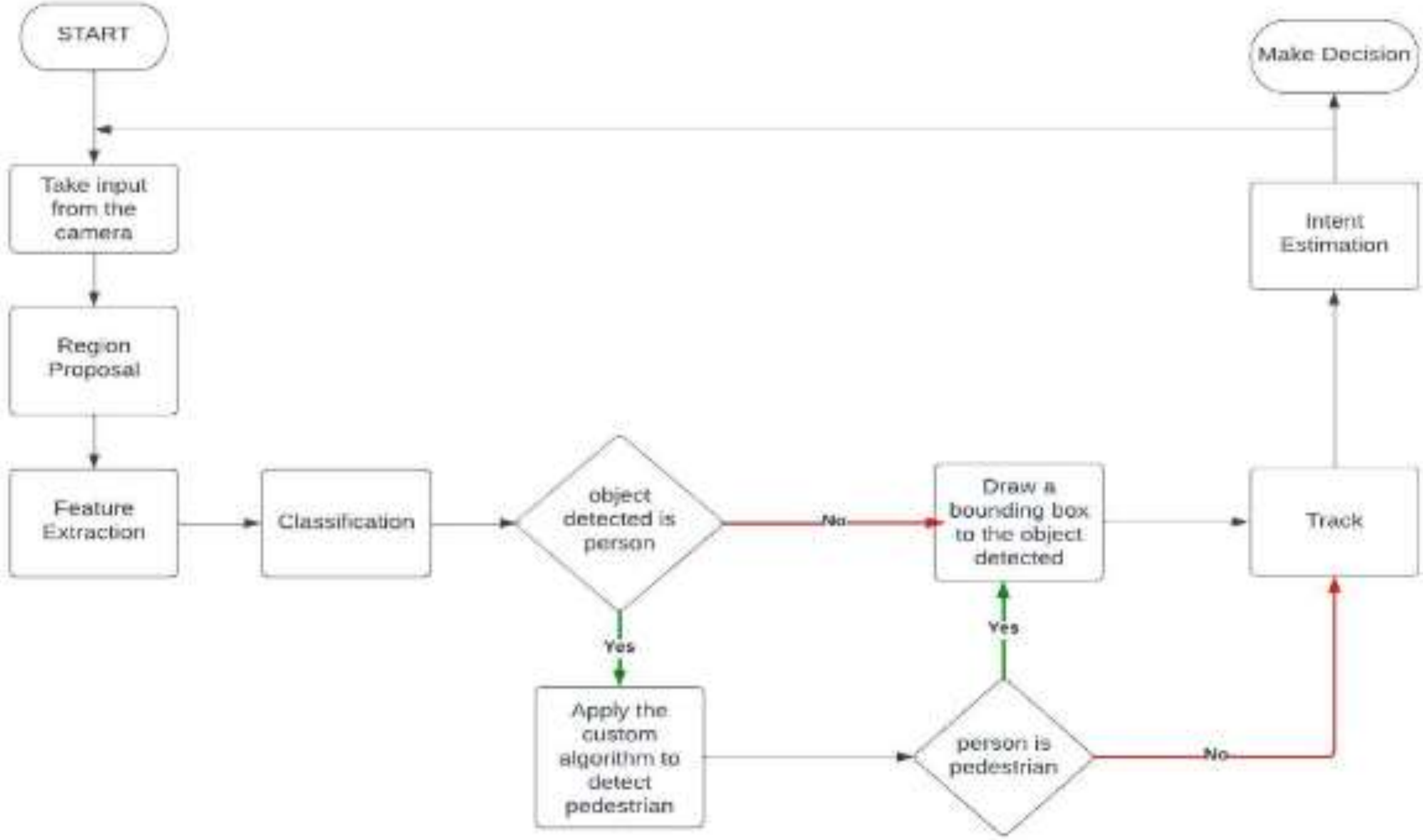
- GOOGLE COLAB LINK FOR COMPARISON REFERENCE WITH INTEL DEVCLOUD

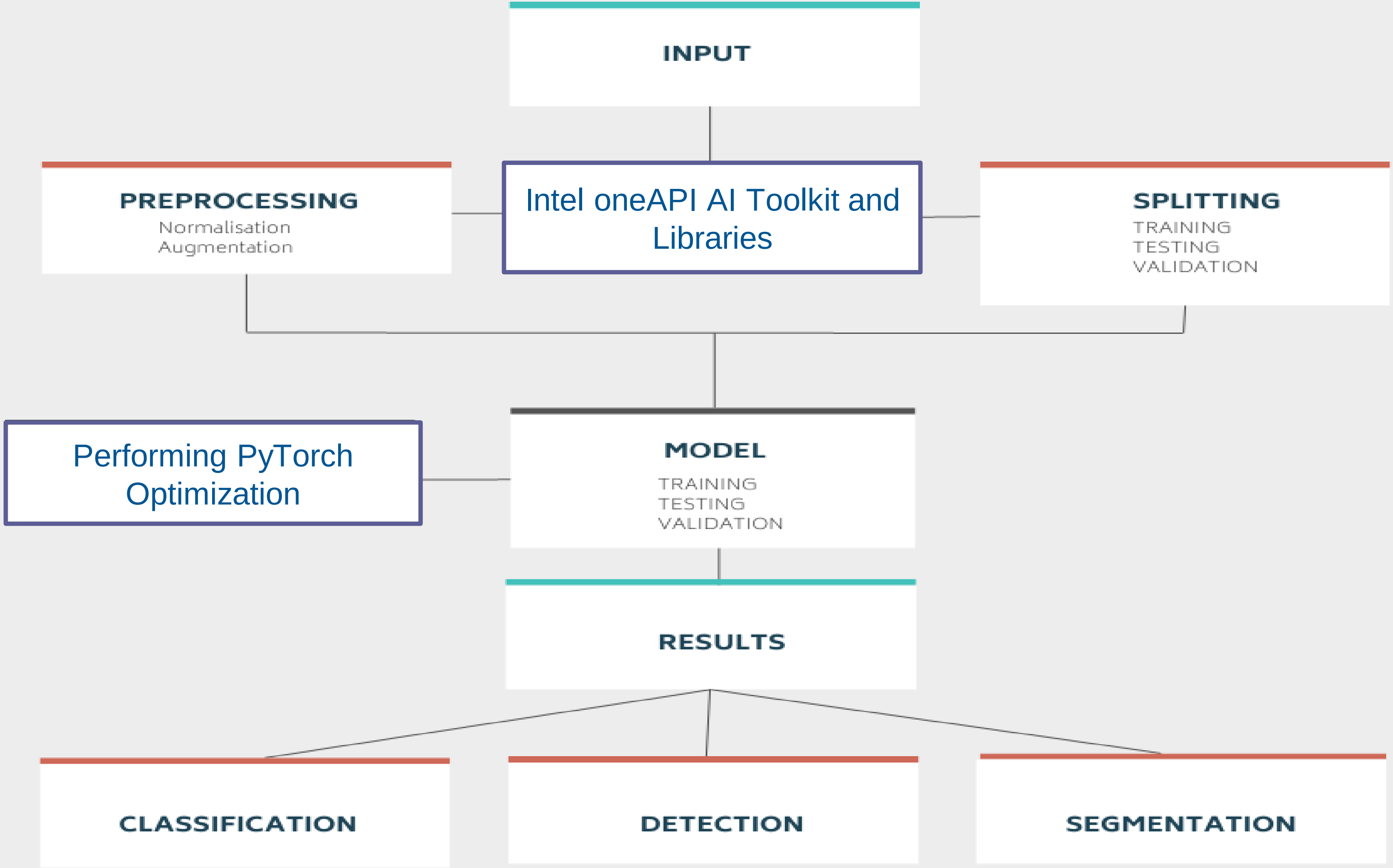
https://colab.research.google.com/drive/1ochHn0XRWrw3bUf5FteJgevc0iR8CTVP?usp=sharing

- WRITE-UP LINK ON MEDIUM

Given in the README File of the Github Repository

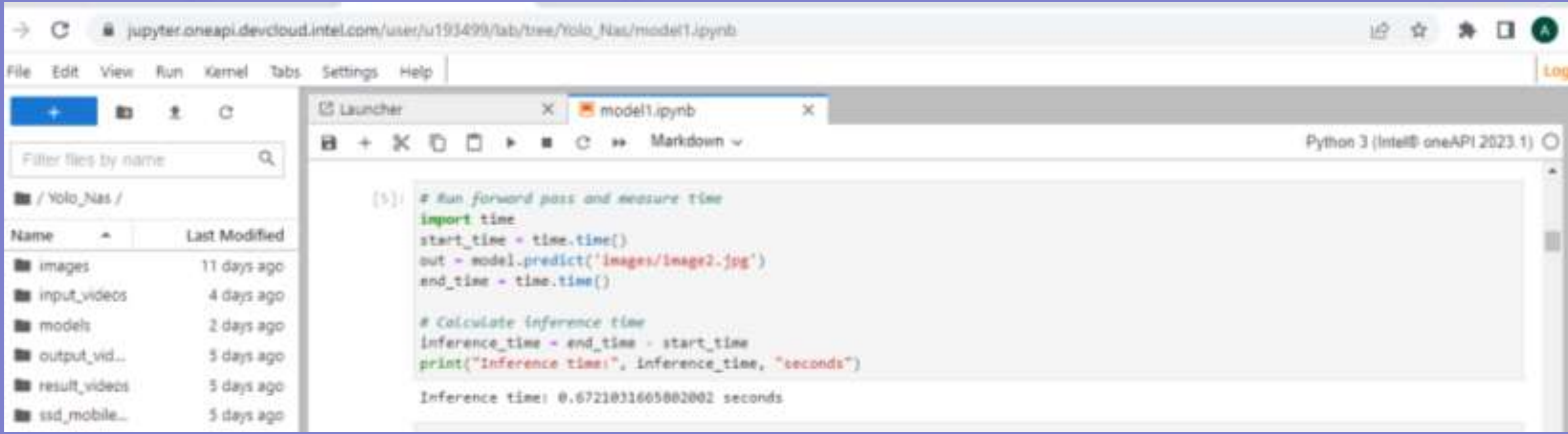
Process Flow Diagram



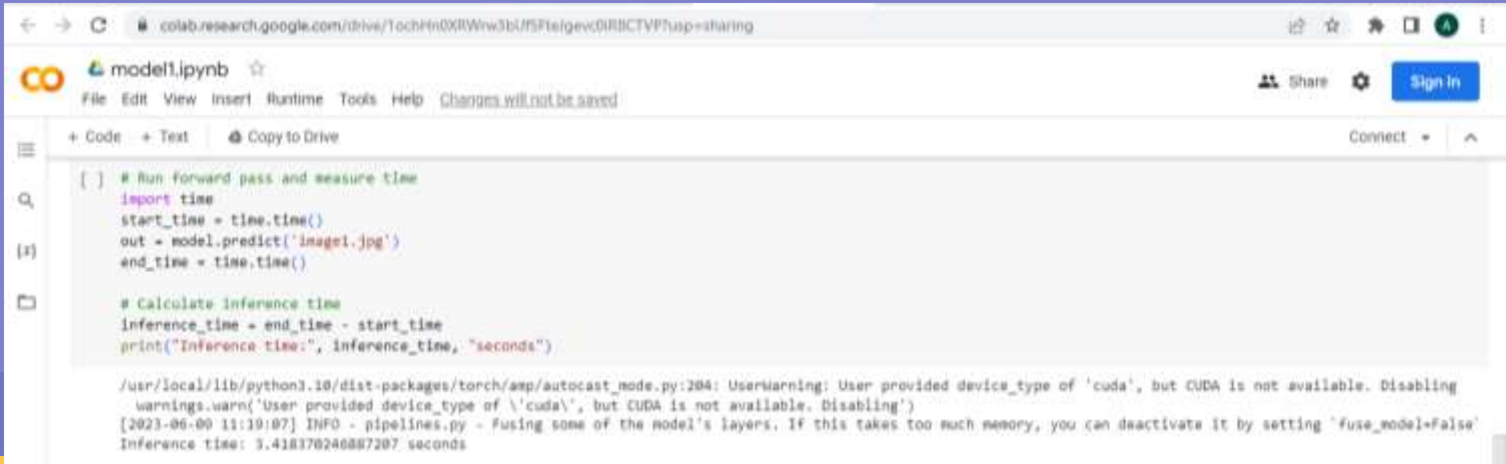

INPUT
PREPROCESSING
Normalisation
Augmentation
Intel oneAPI AI Toolkit and Libraries
SPLITTING
TRAINING
TESTING
VALIDATION
Performing PyTorch Optimization
MODEL
TRAINING
TESTING
VALIDATION
RESULTS
CLASSIFICATION
DETECTION
SEGMENTATION

Benchmarking of Intel® AI Analytics Toolkits and its Libraries

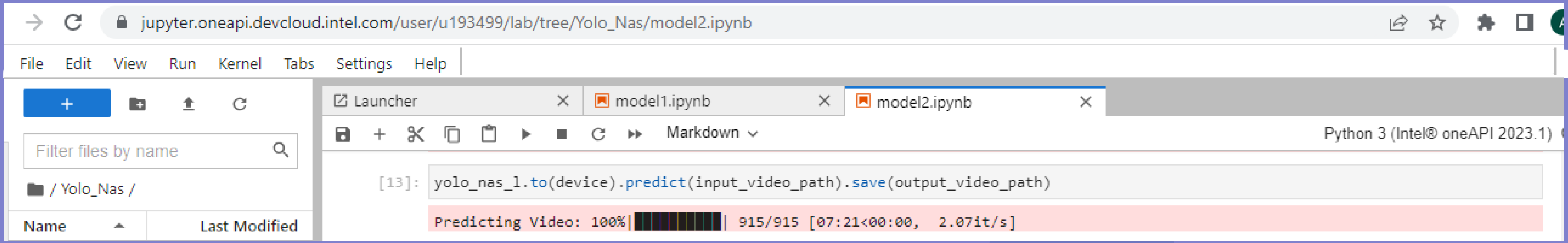
Inference Time in Intel DevCloud is 0.6721 seconds

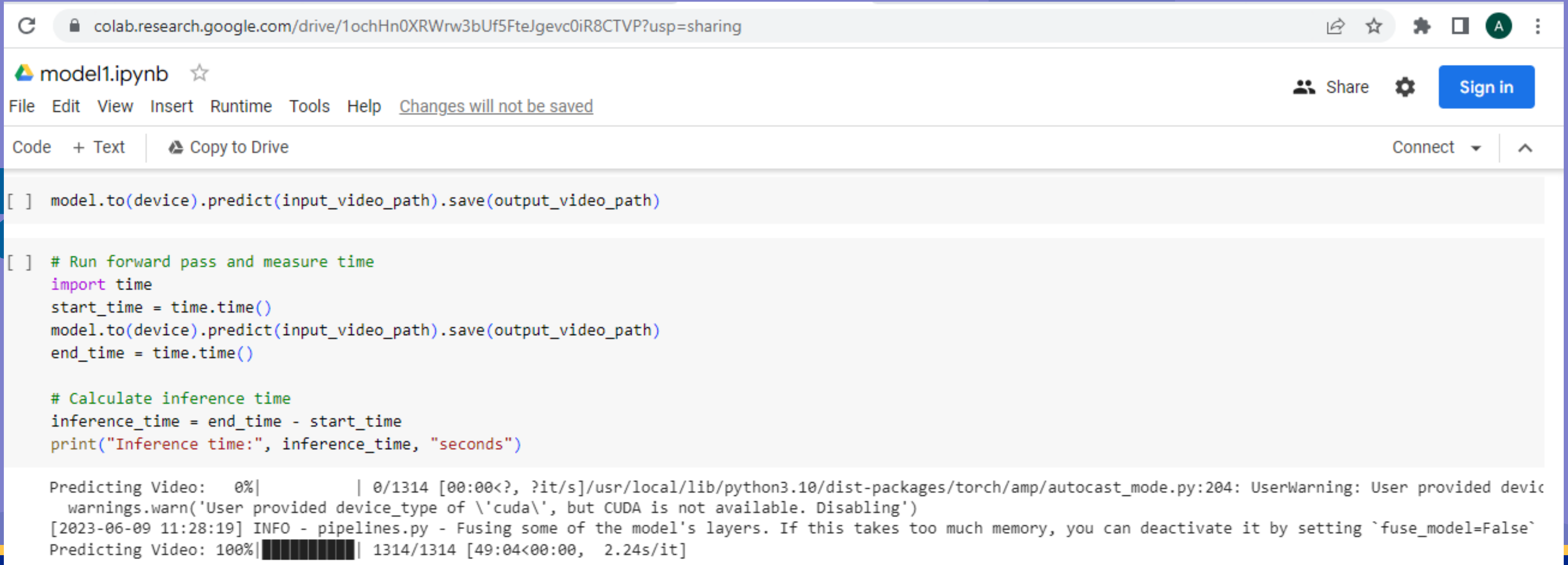


Inference Time in Google Colab is 3.4183 seconds



Benchmarking of Intel® AI Analytics Toolkits and its Libraries

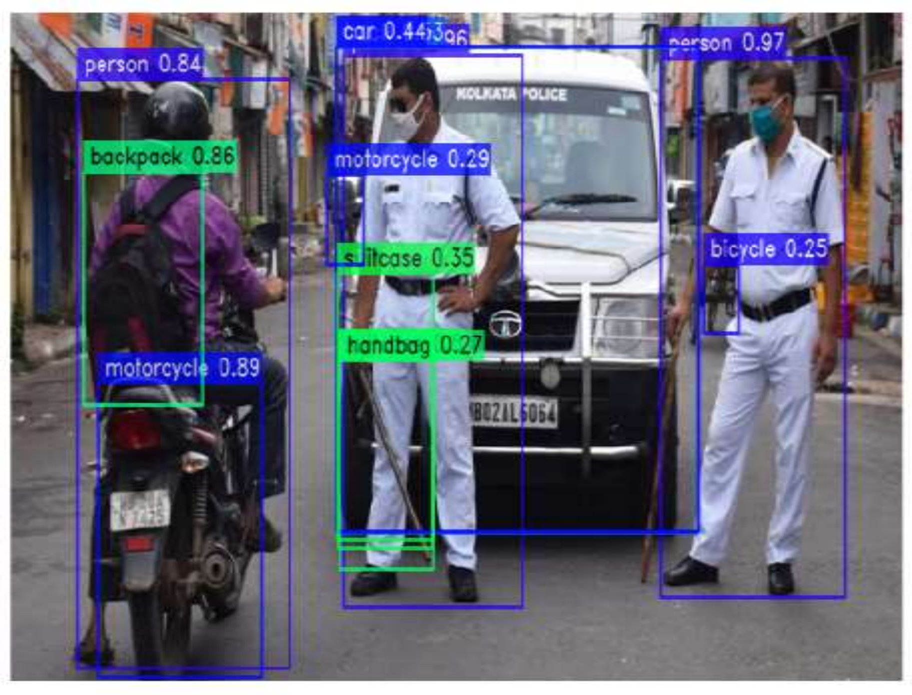




RESULT SUMMARY

1. Multiple Object Detection

The objects detected for autonomous driving are "pedestrian", "traffic light", "traffic sign", "truck", "train", "person", "bus", "car", "rider", "motorcycle", "bicycle“ as present in the COCO Dataset



RESULT SUMMARY

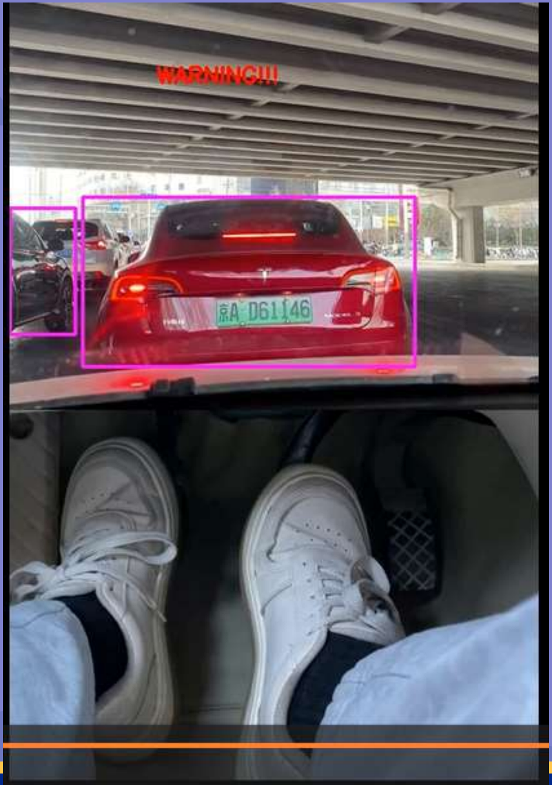
2. Detected Objects in different lightning conditions

The objects were detected in varied lightning conditions such as during Bright Daylight, Night, Foggy Environment, Rain, Smog

RESULT SUMMARY

3. Detected Objects with Distance Mapping

A WARNING alarm is displayed on screen for collision prevention in autonomous vehicles from nearby objects.



RESULT SUMMARY

4. Greater Accuracy achieved using YOLONAS

YOLO-NAS-INT8-M, demonstrates a 50% improvement in inference latency, while at the same time sporting a 1 mAP increase in accuracy.

5. Faster Optimization by using Intel Extension Of PyTorch

Intel PyTorch integrated well with the broader Intel software ecosystem, including the Intel oneAPI AI Analytics Toolkit and Intel DevCloud. Intel PyTorch improved model performance, scalability, and optimization for deep learning algorithms on Intel architecture.

6. Efficient Virtual Environment on Intel DevCloud

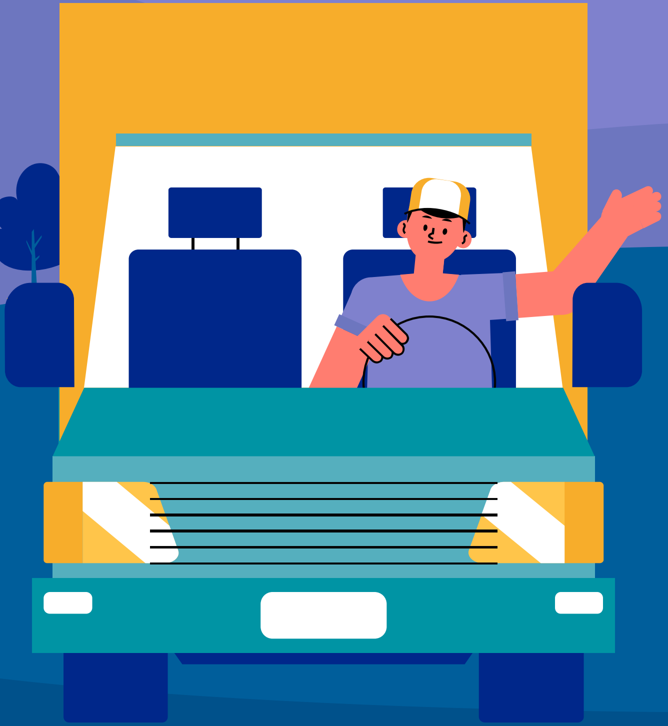
DevCloud comes pre-installed with various software tools and frameworks commonly used in AI and deep learning development, including popular frameworks like TensorFlow, PyTorch, and Caffe. This eliminated the need to set up own software environments and accelerates the development process.

CONCLUSION & FUTURE WORK

- For autonomous vehicles, the detection of objects with every single thing present and with high accuracy as much as possible is of utmost priority. So, in the future, we are going to add more object detections like Traffic Light Colors, Path Detection, and Every Single Traffic Signs which could assist the vehicles to automatically recognize and get an idea of the upcoming objects in their way.

- Voice Assistant Feature is also to be added so that if our project were to be used in any vehicle irrespective of whether autonomous or not, it could easily detect and gives an alarming sound to the drivers to be cautious of the coming object with what type of object in any harsh weather conditions.

- Thus we believe that Intel AI Toolkit and Intel oneAPI will be used for our future work for greater accuracy and speed to help us solve this real-life problem which could decrease the Road-Accidents Rate to a minimal amount in our country and could able to save more precious life of the innocent citizens of India.
- THUS, INTEL IS THE BEST AMONG OTHERS…!!



THANK YOU!

PRESENTED BY :-

Hans Upadhyay
Aishwarya Upadhyay

TEAM NEXTIN