Team name: Scoops Troops

Problem statement : Object Detection For Autonomous Vehicles

Brief about the idea

Our idea is to develop an object detection solution for autonomous vehicles using a combination of computer vision , machine learning techniques and intel oneAPI tools.
Our solution will enable the vehicles to detect and classify objects in its immediate surroundings as well as in the blindspots where human eyes can't reach using LiDAR point clouds.
This solution detects and classify objects such as other vehicles, pedestrians, traffic signals, and road signs,vehicles in the other lanes to ensure safe and efficient navigation.
Since most of the times the accidents on the roads are caused due to overtaking other vehicles in the wrong time, Our solution predicts where will the vehicle in front, behind and in sides of us is going to be using it's behavior and the way it's being driven in it's past few seconds. This helps to predict the perfect time for our vehicle whether to overtake or stay along and make accurate decisions which enhances safety as well as performance.
The solution will integrate various data sources, including visual data from cameras, LIDAR point clouds, and radar signals, to build a comprehensive dataset for object detection and navigation. This data will be preprocessed and fed into a feature extraction model, such as a deep learning model, to extract relevant features and improve the accuracy of object detection and navigation of vehicles more safely and efficiently.

opportunity

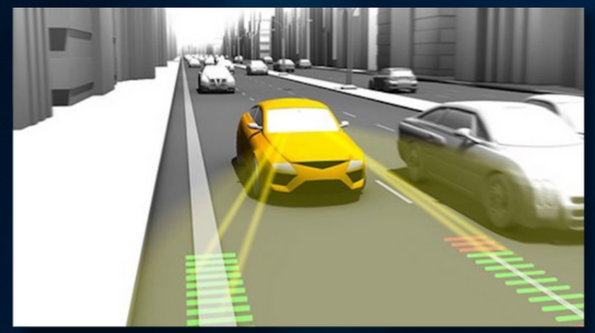
Ability to solve the problem

- Enhanced Safety: By enabling the vehicle to detect and classify objects in its immediate surroundings and in it's blind spots, the solution can significantly enhance the safety of autonomous vehicles.
- Robustness and Adaptability: By incorporating reinforcement learning techniques, the solution can continuously learn and adapt to changes in the environment, improving its overall robustness and reliability.
- Real-Time Object Detection: The solution can perform real-time object detection, enabling the vehicle to react quickly to changes in the environment.
- Improved Efficiency: The solution can improve the efficiency of autonomous vehicles by enabling them to navigate through complex and dynamic environments with greater accuracy and reliability.
-

Innovation over other ideas

- Integration of Multiple Data Sources: The solution integrates visual data from cameras, LIDAR point clouds, and radar signals, to build a comprehensive dataset for object detection. This enables the system to obtain a more accurate and reliable understanding of the environment than solutions that rely on a single data source.
- OneAPI Toolkit Optimization: The solution is optimized using the OneAPI toolkit to ensure maximum performance across a wide range of hardware architectures, which leads to improvements in the efficiency of the system, enabling us to perform real-time object detection more quickly and accurately.
- Predicts the next activity of other vehicles: Apart from just object detection and image recognition , this solution also predicts the behavior of the vehicles near to it and takes decision that is more safe and efficient.
- Detecting vehicles in blindspots: Most of the time accidents occur because of not knowing about the vehicles in our blindspots, this solution used LiDAR point clouds to alert our system about the vehicles in out blindspots.

Features



It can sense lanes

The lane changes and the other vehicle activity can be seen on the dashboard of the car

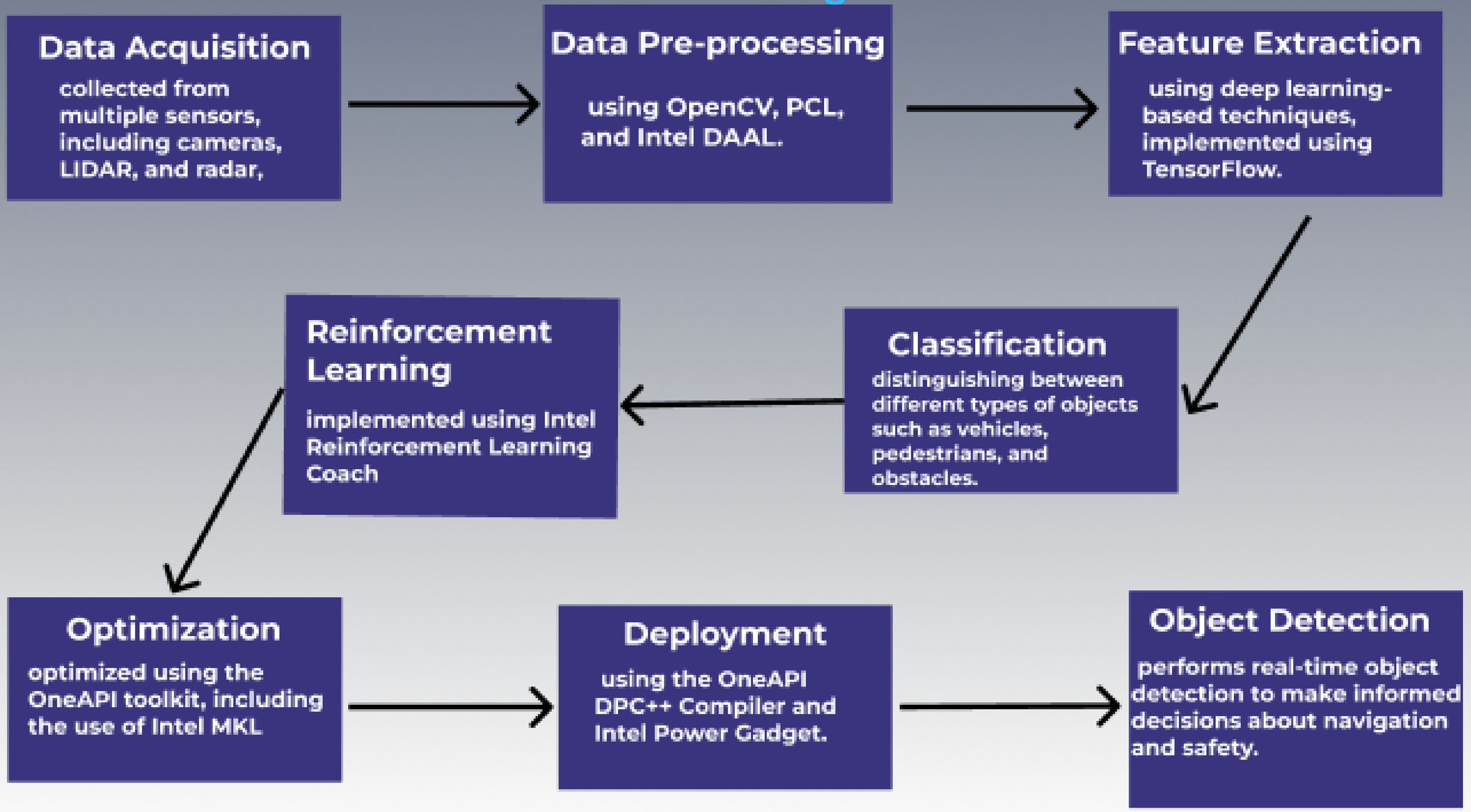
It can sense sign boards , other vechicles activities
and it can indicate the lane with green where you can go safely , red with a warning

As you can see a pedestrian crossing the road without proper knowledge of what coming beside but we can deduct the pedestrian with the help of surface wave radars and Lidar which can help you to slow with a prior knowledge which can avoid collision

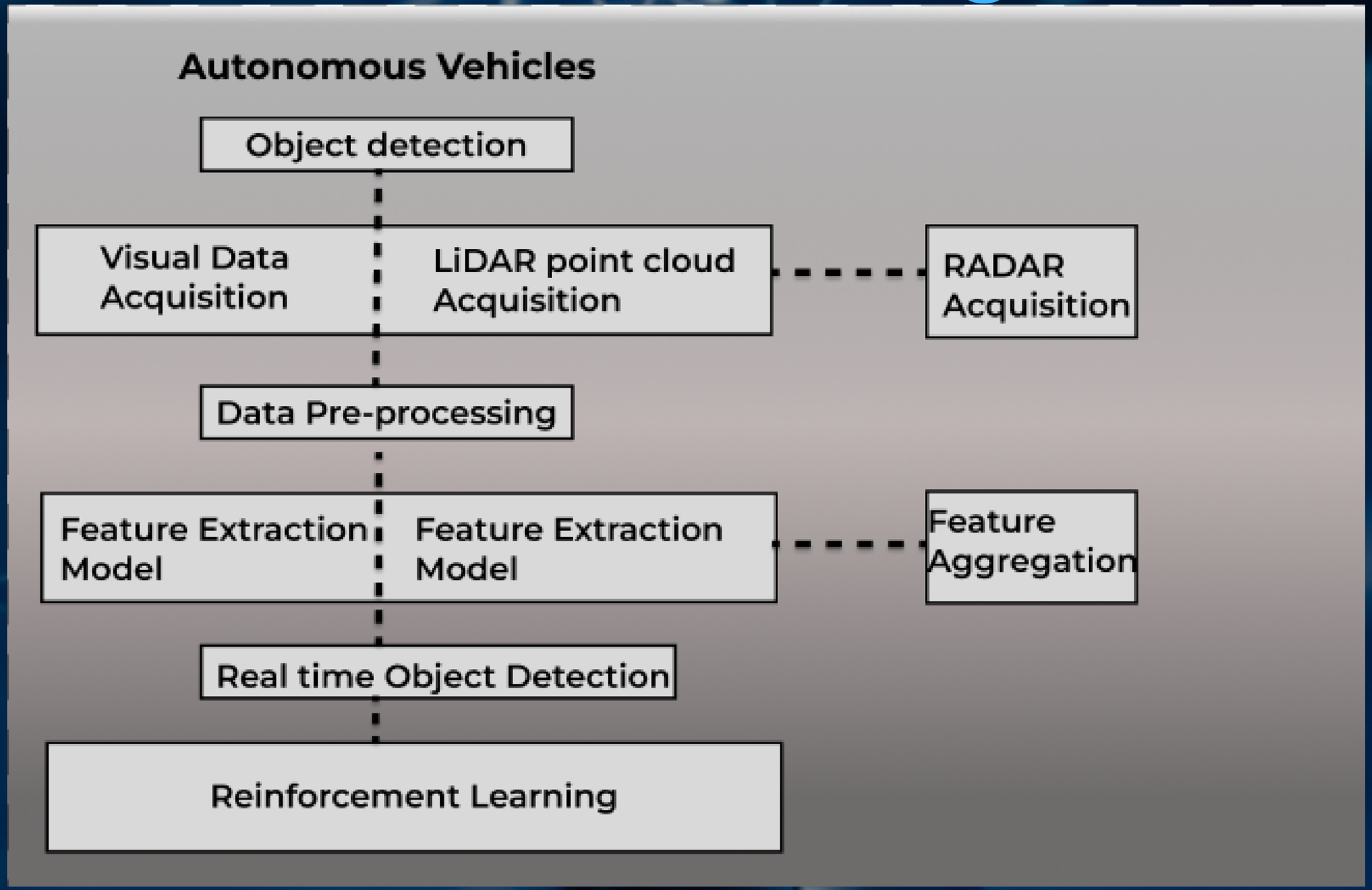
List of oneAPI Ai Analytics Toolkits, its libraries and the SYCL/DPC++ Libraries used

OpenCV is used to preprocess our dataset. **OpenCV** has integration with the Intel Distribution of **OpenVINO** toolkit, allowing us to accelerate the inference process on Intel hardware. It also provides a wide range of functions for image and video processing, including object detection and recognition.
Using of pre-trained models such as **YOLOv5**, **Faster R-CNN**, or **SSD** to detect objects in our dataset.
After selecting a pre-trained model we now have to train it with our pre-processed dataset, We use **Intel oneAPI Deep Neural Network Library (oneDNN)** and **Intel Distribution of OpenVINO toolkit** to accelerate the training process and optimize the model's performance.
We use the **oneAPI Video Processing Library (oneVPL)**, which provides video processing tools to help with hardware-accelerated video encoding, decoding, and processing.
The **oneAPI Video Analytics Library (oneVAN)** is also used to predict the output with the help of input videos. oneVAN provides us with a set of high-level APIs and pre-built models for video analytics tasks such as object detection, tracking, and classification.
The **SYCL library and DPC++ library** provide a unified programming model for heterogeneous computing, allowing us to write code that can run on a variety of devices, including CPUs, GPUs, and FPGAs.

Process flow diagram



Architectural Diagram



Technologies used

- **Programming Languages**: Python, C++.
- **Machine Learning Frameworks**: Popular machine learning frameworks such as TensorFlow, PyTorch, and Keras can be used for building and training machine learning models.
- **Computer Vision Libraries**: Libraries such as OpenCV can be used for image processing and computer vision tasks.
- **OneAPI Toolkit**: Intel Distribution of OpenVINO toolkit , oneAPI Video Processing Library (oneVPL),oneAPI Video Analytics Library (oneVAN), Intel oneAPI Deep Neural Network Library (oneDNN), The SYCL library and DPC++ library.
- **Development Tools**: Intel Devcloud, version control systems such as Git, and project management tools such as JIRA will be used.
- **Testing Tools**: Unit testing frameworks such as PyTest will be used for testing the prototype's functionality.
- **Visualization Tools**: Visualization tools such as Matplotlib and TensorBoard will be used.

Estimated cost

- Intel Realsecse LiDAR Camera L515

  Cost : 60,999 Rs

- Pixy 2.1 smart vision sensor

  Cost : 8549.00 Rs

- Raspberry Pi 3

  Cost : 4543.00 Rs

Total : 74,091.00

Team Members:

1.Anto jerish S

2.Swathi Subramani