AI Automotive Project

# Requirements Analysis:

## Features:

* Three main features:
  1. Distraction Detection – Head pose and iris
  2. Drowsiness Detection – Blinking frequency, yawning, head nodding
  3. Object Detection – Seat belt usage, phone usage, smoking
* Into two problems:
  + Pose detection
  + Object Detection

## Constraints:

* Accuracy > 90% (? For which dataset under which scenarios and using what metrics)
* Conditions:
  + Light: 20 – 500 lux (Dataset prepared or not)
* Inference Time (CPU only):
  + > 15 fps
* Target hardware:
  + RAM: <= 200 MB
  + CPU: <= 7500 DMIPS - Dhrystone MIPS (Million Instructions per Second) – Should evaluate by DMIPS/MHz instead
  + Some suggested chips: Qualcomm SA6155P, Qualcomm SA8155P, Qualcomm SA8195P, Qualcomm SA8255P SA8775P => snapdragon automotive chips could be using with Snapdragon Neural Processing Engine SDK

## Solutions:

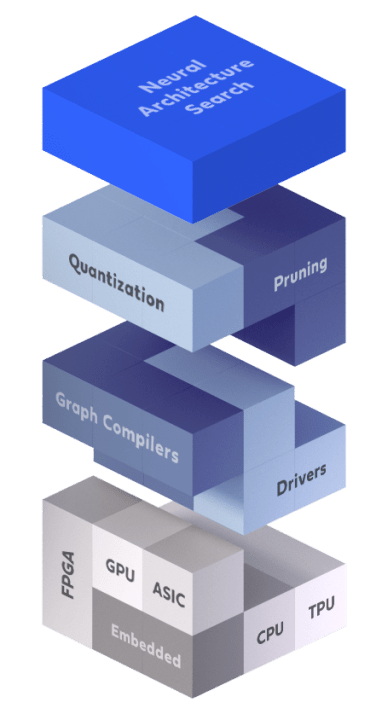
* Using 2 public models one for pose detection (need produces eyes point for references: ML Kit Pose Detection ) and one for object detection, should focus on how to accelerate model (suggest using two main techniques: quantization and graph compilers

A screenshot of a computer screen

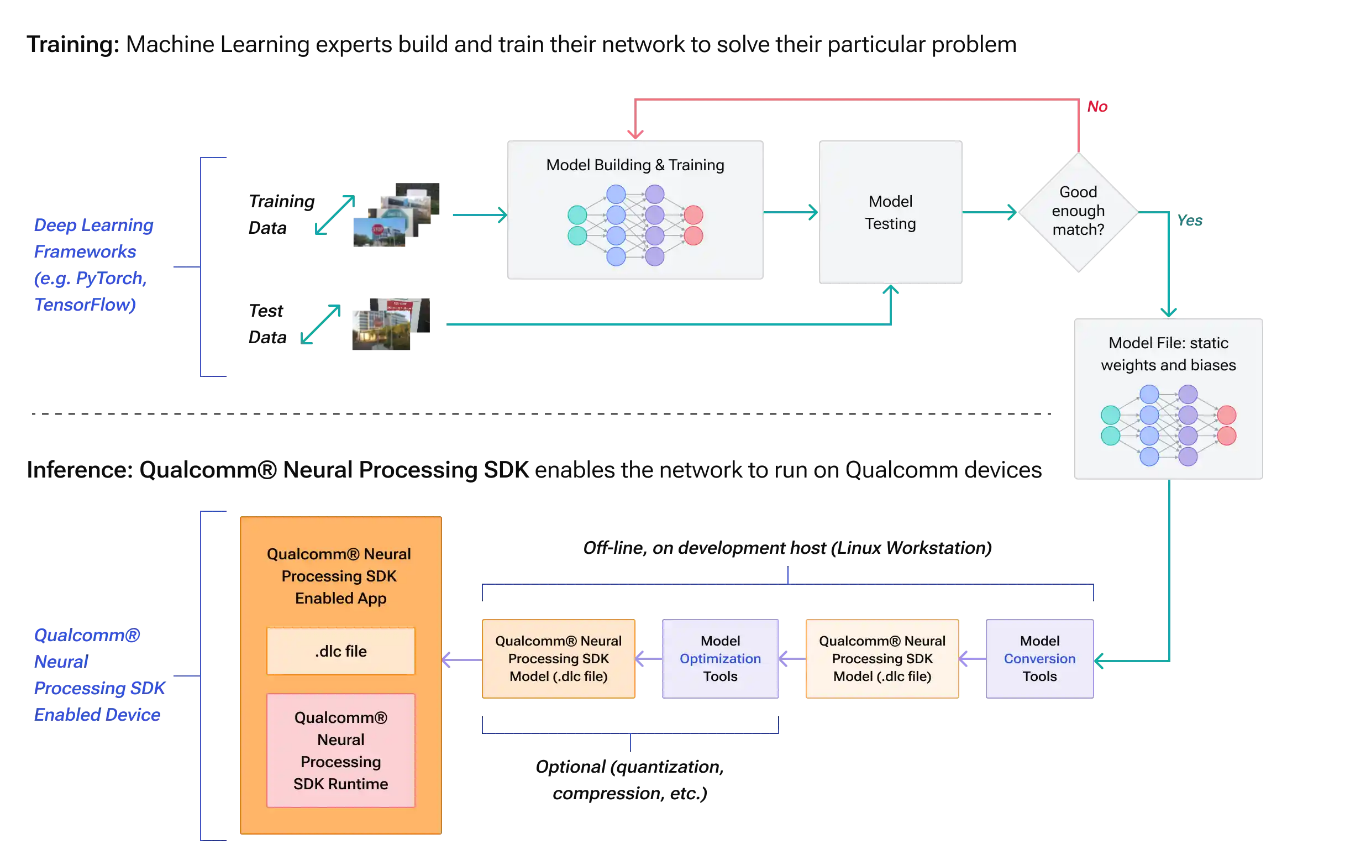
Description automatically generated

ML Kit Pose Detection

* + Pose detection:
    - Should focus on these point marks: eyes, shoulders, Hand, elbow.
    - ML Kit Pose Detection has already been packed into API that can use in Android/ IOS with performance ~ 30FPS (for accurate version) and ~23FPS for normal version with size 10.1~13.3MB
  + Object detection:
    - Object detection models is plenty and easy to find/ custom train
* Acclarate model



* + As mention before, we’re focusing on graph compiler and quatiztion and our’s target hardware is Qualcomm chip which support Qualcomm® Neural Processing SDK
  + Propose Model Workflow (supported by the SDK):



* + - SDK already support quantization and model conversion from vary popular deep learning framework (PyTorch, TFLite, ONNX, and TensorFlow,..)
* **Our’s main job will be:**
  + Inference:
    - Model conversion:
      * directly Pytorch -> DLC
      * take trained weights of Pytorch converts to ONNX then convert to DLC (should do this way because will separate to task/ team training and inference).
    - Implementing on C/C++ (graph compiler) load . DLC file to boost up reference time (take the most effort).
  + Training:
    - At the same time, researching, experimenting models and data to get best performance.

A screenshot of a computer screen

Description automatically generated

* Problems:
  + Need to confirm that our’s target hardware is supports SDK or not. In the [websites](https://docs.qualcomm.com/bundle/publicresource/topics/80-63442-2/overview.html) I can’t find SA6155P in the supported list.