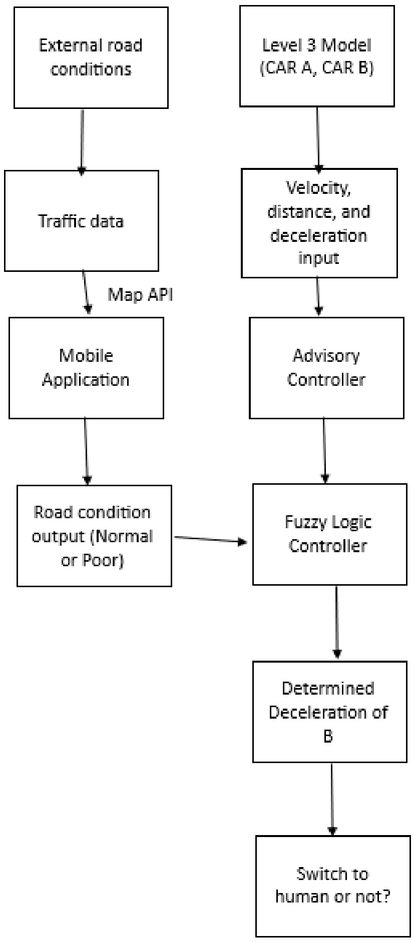
PROJECT 3 ANSWERS

1. Architectural integration of the mobile application and advisory control



2. The video outlines strategies for optimizing a Fuzzy system controller, focusing on a neuro-fuzzy system based on the Mamdani approach. These strategies encompass the tuning of membership functions, the optimization of connecting weights, and efficient training methods.

    1. Optimizing the membership functions, representing input and output variables, is a fundamental step. These gaussian functions can be adjusted in terms of shape and size to enhance the system's performance. This enables a more precise modelling of the input-output relationship.
    2. The connecting weights between layers within the neuro-fuzzy system play a crucial role in controlling the shape of the membership functions. These weights can be fine-tuned to improve the system's accuracy in capturing complex relationships between variables.

    3. Training methodologies for the Fuzzy system involves two primary approaches: the use of the back-propagation algorithm and nature-inspired optimization tools such as genetic algorithms and particle swarm optimization. These methods aim to minimize errors between the desired and actual system outputs, making the system more effective.

Optimizing a Fuzzy system controller involves adjusting membership functions, optimizing connecting weights, and employing efficient training methods. This process allows for the development of a neuro-fuzzy system that accurately models the input-output relationship and enhances the controller's performance in various applications.

3. Deploying a mobile application within a Vehicular Ad-Hoc Network (VANET) involves several key steps for real-time functionality. First, the application must integrate with the VANET infrastructure, enabling communication within vehicles and roadside units. Real-time data exchange is crucial, allowing the app to continuously monitor traffic conditions, including congestion and road conditions.

   Once integrated, the app can leverage the clustering algorithm designed in Task 1 to categorize traffic congestion data into "normal" or "poor" road conditions. Simultaneously, the fuzzy logic advisory control system from Task 2 must be integrated. This system runs the LaneMaintanSystem3Car.slx and assesses Car B's deceleration based on real-time inputs, i.e Car A's velocity, road condition estimates, and distance between car A and car B. It decides whether to switch to human control based on predefined rules.

   The advisory control system should operate continuously, making real-time decisions about switching to human control when necessary. It must record the time of the switch and the required human reaction time. Integration with simulation model "Level3Model.slx" allows for testing and validation of the advisory control system's decisions. A user-friendly interface and clear notifications are essential for keeping the driver informed.

4. Fuzzy logic rules for driver alerts in if-else format.
   Input conditions: driver alertness, traffic conditions, and vehicle proximity. The expected output should be in the form of an alarm or no alarm.

   Fuzzy Logic Rules:

- If Driver alertness=yes, Traffic Condition =Poor and Vehicle Proximity=near, then alarm.
- ELSE If Driver alertness=yes, Traffic Condition =Normal and Vehicle Proximity=near, then no alarm.
- ELSE If Driver alertness=yes, Traffic Condition =Poor and Vehicle Proximity=far, then no alarm.
- ELSE If Driver alertness=yes, Traffic Condition =Normal and Vehicle Proximity=far, then no alarm.
- ELSE If Driver alertness=no, Traffic Condition =Poor and Vehicle Proximity=near, then alarm.
- ELSE If Driver alertness=no, Traffic Condition =Normal and Vehicle Proximity=near, then alarm.
- ELSE If Driver alertness=no, Traffic Condition =Poor and Vehicle Proximity=far, then alarm.
- ELSE If Driver alertness=no, Traffic Condition =Normal and Vehicle Proximity=far, then alarm.