**Project Proposal: Vision-Based Lane Keeping Robocar**

**1. Project Overview**

This project aims to develop an autonomous lane-keeping robotic car using computer vision (CV) approaches. The system leverages the Elegoo Robocar hardware platform, enhanced with an ESP32-CAM module for real-time image capture, a vision-based perception and planning algorithm for steering prediction, and Arduino-based motor control. Key components include:

**ESP32-CAM:** Captures live video and streams it to a PC via Wi-Fi.

**CV Algorithm:** Implement lane marking detection and fitting to predict turning commands.

**PC Server:** Runs the CV algorithm for real-time inference and returns results to the ESP32-CAM.

**Arduino:** Executes motor control based on steering predictions.

The system demonstrates a closed-loop workflow: image acquisition (input) → CV algorithm inference (controller) → motor actuation (output & feedback), enabling autonomous navigation along predefined paths.

**2. Hardware and Software Requirements**

Hardware:

Elegoo Robocar chassis with DC motors (actuators).

ESP32-CAM module (microcontroller with a camera sensor for video streaming).

Arduino Uno (microcontroller for motor control).

Wi-Fi-enabled PC (for data communication via Wi-Fi wireless connection).

Software:

ESP32-CAM: Arduino IDE (C++).

Data Collection: Python, OpenCV.

CV Algorithm: Python, OpenCV.

Motor Control: Arduino IDE (C++).

**3. System Architecture**

The system comprises three interconnected modules.

**Image Acquisition & Streaming:** ESP32-CAM captures JPEG frames and a Wi-Fi hotspot streams video to a PC.

**Real-Time Inference:** The CV algorithm processes live video, predicts steering angles, and sends results to ESP32-CAM via TCP.

**Motor Control:** Arduino interprets steering commands via serial and adjusts motor speeds using PWM. Arduino maps steering commands to PWM signals for differential wheel speeds.

**4. Expected Outcomes**

A fully autonomous lane keeping robocar capable of real-time path tracking in expected scenario.

Demonstrable integration of edge computing (ESP32-CAM) and centralized inference (PC).

**5. Development Timeline**

Hardware Setup & Testing 2 days

Completing CV Algorithm 2 days

System Integration 2 days

Testing & Optimization 2 days

Final Project Report & Presentation 1 week

**6. Conclusion**

This project bridges embedded systems and computer vision methods to create an efficient autonomous navigation solution. By combining low-cost hardware with a lane keeping algorithm, it offers a scalable framework for industrial applications, such as automated guided vehicles and lane keeping assistance. Future work may include replacing CV approach with CNN model and apply edge deployment to eliminate PC dependency.