ROAD SIGNS DETECTION FOR AUTONOMOUS VEHICLES BY USING CNN AND KERAS

### ****1. Introduction****

Road sign detection is a critical component in autonomous vehicles, enabling them to interpret and respond to various traffic signs on the road. This process involves the use of Convolutional Neural Networks (CNNs) to classify and detect road signs from real-time images. By leveraging advanced computer vision techniques, the system can accurately identify and categorize road signs, ensuring safer navigation and compliance with traffic regulations.

### ****2. Product Goal****

The product goal of a road sign detection system using CNNs and Keras is to develop a robust and efficient tool that can automatically recognize and classify road signs from vehicle camera feeds. This system aims to achieve high accuracy in sign recognition, provide real-time performance for autonomous driving, and integrate seamlessly with vehicle navigation systems for improved road safety and compliance.

**3. Demography (Users, Location)**

**Users**

* **Target Users**: Autonomous vehicle manufacturers, automotive safety regulators, transportation agencies, technology developers, and research institutions.
* **User Characteristics**: Users range from automotive engineers and AI developers to traffic safety experts and regulatory bodies, with varying levels of expertise in machine learning and computer vision.

**Location**

* **Target Location**: Global, with particular focus on regions with advanced autonomous vehicle testing, urban and suburban areas with diverse traffic signs, and institutions involved in automotive research and development.

**4. Business Processes**

**Key Business Processes**

1. **Data Collection**:
   * Collect a dataset of road sign images, such as those from the German Traffic Sign Recognition Benchmark (GTSRB) or similar sources.
2. **Data Preprocessing**:
   * **Explore the Data**: Analyze the dataset to understand variations and identify any issues.
   * **Resize Images**: Standardize image dimensions (e.g., 32x32 pixels) for uniform processing.
   * **Normalize Pixel Values**: Scale pixel values to the range [0, 1] for improved model performance.
   * **Data Augmentation**: Apply transformations such as rotation and flipping to increase data variability.
3. **Data Splitting**:
   * Split the dataset into training, validation, and testing sets to develop and evaluate the model effectively.
4. **Model Selection**:
   * Choose appropriate CNN architectures (e.g., LeNet, VGG16) for detecting and classifying road signs.
   * **Model Components**:
     + Convolutional Layers: Extract features from images.
     + Pooling Layers: Reduce spatial dimensions.
     + Dense Layers: Perform classification based on extracted features.
5. **Model Evaluation**:
   * Evaluate model performance using metrics such as accuracy, precision, recall, and F1-score.
   * Use confusion matrices and ROC-AUC curves to assess classification performance.
6. **Visualization**:
   * Visualize the results by plotting model performance metrics and analyzing misclassified images to improve model accuracy

6. Authorization Matrix

|  |  |
| --- | --- |
| Role | Access Level |
| User | |  | | --- | | Access to real-time road sign classification and feedback features. |  |  | | --- | |  | |
| Admin | |  | | --- | | Manage model training, performance monitoring, and updates with new datasets. |  |  | | --- | |  | |
| Developer | Access to backend settings for modifying model architecture and training pipelines. |

**6. Assumptions**

* The dataset is comprehensive and representative of various road sign types and environmental conditions.
* The preprocessing steps effectively standardize image data and enhance model learning.
* The chosen CNN architecture is suitable for the road sign detection task and performs well on real-time video streams.