**Traffic Sign Detection App**

**Requirements Model**

**Version 1.2**

**Team Members:**

Hayden Jones

Shreya Komarabattini

Rishigesh Rajendrakumar

Aidan Mao

Jacob Heffelmire

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Name** |
| 10/5/25 | 1.0 | First draft | Hayden Jones |
| 10/30/25 | 1.1 | Update product backlog | Hayden Jones |
| 12/2/25 | 1.2 | Expanded functional/non-functional requirements with updated product/backlog | Rishigesh Rajendrakumar |

**Requirements**

**Functional Requirements:**

The team will develop a fully functional Traffic Sign Recognition (TSR) app for Android devices:

* The app accepts input from a live camera feed or uploaded images/videos.
* The app processes images and detects traffic signs using a CNN-based recognition model.
* The app displays recognized signs visually on-screen with bounding boxes and labels.
* The app provides text-to-speech (voice) feedback announcing recognized signs.
* The app shows confidence percentages for each detected sign.
* Users can select regional sign sets (e.g., U.S., EU, Asia) for detection.
* The app uses offline operation, using preloaded models and datasets.
* A testing mode allows developers or students to evaluate recognition performance using sample images or videos. Metrics such as accuracy, precision, and recall are reported.
* Users can adjust settings for audio alerts, visual overlays, and detection thresholds.
* Critical alerts (e.g., Stop, Yield, Wrong Way) are prioritized to prevent alert congestion when multiple signs appear simultaneously.
* The app is optimized for real-time performance, maintaining at least 15–30 FPS with latency under 200 ms per frame.
* The app logs detection events for debugging and future model improvement.
* The app supports different confidence thresholds that users can modify.

**Non-Functional Requirements**

**Security:**

* The app processes all camera images locally; no cloud upload is required.
* Camera feed and voice input is not saved.

**Efficiency & Performance:**

* The app must maintain real-time recognition without causing phone lag or crashes.
* The model must be lightweight enough to run on mid-range Android devices.

**Usability:**

* The UI must be simple and intuitive, suitable for hands-free driving interaction.
* Large buttons and clear labels must be provided.

**Reliability:**

* Recognition should be consistent across varied lighting, weather, and speed conditions.
* The system should recover gracefully if the camera feed fails.
* The system must handle poor lighting, motion blur, and partial occlusion without crashing.

**Product Backlog**

|  |  |
| --- | --- |
| **Backlog Items** | **Progress** |
| **1.0 Frontend Setup** |  |
| 1.1 Front end design | **Done** |
| 1.2 Create front end UI | **Done** |
| 1.3 Integrate camera functionality | **Done** |
| 1.4 UI-Backend Data Handling | **Done** |
| **2.0 Backend Setup** |  |
| 2.1 CPU Tensor Operations | **Done** |
| 2.2 Logger Class | **Done** |
| 2.3 GPU Tensor operations | **Done** |
| 2.4 OpenCV preprocessing | **Done** |
| **3.0 Model Training** |  |
| 3.1 Dataset Preparation | **In Progress** |
| 3.2 Setup model layout | **In Progress** |
| 3.3 Loss function and gradients | **In Progress** |
| 3.4 Data loading | **In Progress** |
| 3.5 Training loop | **In Progress** |
| 3.6 Parameter Management | **In Progress** |
| 3.7 Model Export | **In Progress** |
| **4.0 Output & feedback** |  |
| 4.1 JNI integration | **Not Started** |
| 4.2 Display Results | **Not Started** |
| 4.3 Text-To-Speech | **Not Started** |
| **5.0 Testing & Optimization** |  |
| 5.1 Unit & integration testing | **Not Started** |
| 5.2 Performance tuning | **Not Started** |
| 5.3 Device Compatibility | **Not Started** |

**User stories**

**Driver / User:**

* As a driver, I want to see recognized traffic signs on my screen so I can respond quickly while driving.
* As a driver, I want the app to announce detected signs via voice so I don’t have to look away from the road.
* As a driver, I want the app to prioritize critical signs like Stop or Yield when multiple signs appear, so I receive the most important alerts first.
* As a driver, I want to select a region’s traffic signs, so the app recognizes signs relevant to where I am driving.
* As a driver, I want the app to work offline, so I can receive alerts even in areas without internet connectivity.
* As a driver, I want to be able to use voice commands to keep my hands free while driving to control settings and start/stop the app.

**Use Case Diagram:**

