Traffic Sign Detection App

Risk Assessment

Version 1.1

Team Members:

Hayden Jones

Shreya Komarabattini

Rishigesh Rajendrakumar

Aidan Mao

Jacob Heffelmire

Title: Traffic Sign Recognition Application	
Version: 1.1	Date: 10/5/25

Revision History

Date	Version	Description	Name
10/5/25	1.0	First draft	Hayden Jones
10/31/25	1.1	Added risk	Hayden Jones

Risks Assessment:

1. Model performance not meeting real-time requirements
   The CNN model may process frames too slowly for live detection on mobile devices.

   - **Project Impact:** Delays in recognition could make alerts inaccurate or unsafe for drivers.
   - **Mitigation Strategy:** Optimize model using Yolo algorithm and other optimization strategies for CNNs.
   - **Contingency Plan:** Fall back to lower-resolution input or a simplified model version if FPS drops below a usable amount.

2. Low detection accuracy under poor lighting or weather
   The system may fail to correctly identify signs in glare, shadows, or rain.

   - **Project Impact:** Reduced trust and usability in real-world driving conditions.
   - **Mitigation Strategy:** Use data augmentation during training to simulate various lighting/weather. Implement preprocessing filters (contrast enhancement, normalization).
   - **Contingency Plan:** Notify the user when visibility is poor and pause alerts until stable detection resumes.

3. Team coordination and time constraints
   Balancing model training, UI design, and testing may be difficult within the designated timeline.

   - **Project Impact:** Delayed milestones and incomplete application.
   - **Mitigation Strategy:** Use Jira to create a project timeline and divide tasks between group members. Use Git for version control and to see each other's progress. We will also hold weekly meetings every Friday.
   - **Contingency Plan:** Reallocate tasks when needed and prioritize only the core functionality. We may have to omit low-medium priority features if we are short on time.

4. Unplanned sickness/family emergences
   Unplanned sickness and emergencies can cause significant delays in development.

   - **Project Impact:** Group members being unavailable due to emergencies can cause delays in development, since they are unable to complete the work they were assigned.
   - **Mitigation Strategy:** We will mitigate this risk by having clear communication between group members. By communicating, we can plan around a group member's absence and divide the workload between the remaining members.
   - **Contingency Plan:** If the mitigation strategy fails, we can talk to our project sponsor regarding our expectations for the final project.

5. Scope Creep
   The project scope may expand beyond the original requirements.

   - **Project Impact:** Expanding scope can lead to missed deadlines, incomplete features, or reduced app quality due to divided focus. It can also overextend team resources and delay testing or integration.
   - **Mitigation Strategy:** Clearly define project deliverables early. Regularly review scope against initial objectives and timeline.
   - **Contingency Plan:** If scope creep occurs, prioritize core features. Defer optional features to future versions or documentation as future work.
6. Training Taking Very Long
   Training a large CNN can take a very long time, and we may not have enough time to train the model
   - **Project Impact:** Not being able to train the model fully can make the accuracy low.
   - **Mitigation Strategy:** Use CUDA to speed up training time by performing operations on the GPU. We can also ensure that we finish the training module first so that we can start training as soon as possible.
   - **Contingency Plan:** We could use a more simplified model that is faster to train, or not train on as many images.