TEST PLAN FOR

DISTRACTED DRIVER DETECTION

Made By:
Utkarsh Mishra 2000290120179
Sarthak Tyagi 2000290120137
Shruti Gupta 2000290120156

Test Manager:
Prof. Shreela Pareek

Project Coordinator:
Prof. Puneet Goyal

Submitted To:
Prof. Neha Shukla

ChangeLog

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made
001	05/11/2023	Sarthak Tyagi	Initial Draft

1 Introduction

The pursuit of generating this system was cultivated after comparative observations of the real-life events that are occurring in and around the globe. While driving the vehicle, drivers frequently perform secondary activities that distract driving. A decrease in driver distraction is a critical aspect of the smart transportation system.

In recent years, the number of road accidents has steadily increased throughout the globe. According to a study by the National Highway Traffic Safety Administration, 45% of vehicle collisions are the result of a distracted motorist in the immediate vicinity of the incident. We seek to provide a system that is accurate and reliable for identifying diverted drivers. Drivers regularly engage in activities that distract them while operating the vehicle. A reduction in motorist distraction is an essential component of the intelligent transportation system. Various convolutional networks had been trained on images by omitting the last layer to obtain their feature vectors. Using the stacking ensemble technique, we stacked all the feature vectors to train a convolutional network.

This project, "Detecting Distracted Drivers with CNN and ML," aims to contribute to road safety by leveraging these cutting-edge technologies. We will employ CNN to analyze real-time images and video feeds from vehicle-mounted cameras, identifying signs of distracted driving such as texting, using handheld devices, or engaging in other risky activities. Machine Learning algorithms will be used to train the system, enabling it to recognize various forms of distraction with high accuracy.

1.1 Scope

1.1.1 In Scope

1. Data Collection and Preprocessing:
   - Gather a diverse and extensive dataset of images and videos capturing various instances of distracted driving.
   - Preprocess the data to enhance its quality and standardize the format for model training.
2. CNN Model Development:
   - Develop a Convolutional Neural Network (CNN) architecture capable of analyzing images and videos in real-time.
   - Train the CNN to recognize signs of distracted driving, including activities like texting, talking on the phone, eating, and more.
3. Machine Learning Algorithm:
   - Implement machine learning algorithms to fine-tune the CNN model's ability to identify different forms of distraction.
   - Optimize the model's accuracy and efficiency through iterative training and testing.
4. Real-Time Image Analysis:

- Create a system that can process and analyze images and videos in real-time, either from in-vehicle cameras or external sources.
- Ensure the system can identify and classify distracted driving behaviors promptly and accurately.

1.2 Out of Scope

1. Autonomous Driving:
    - This project does not involve the development of autonomous driving technology, self-driving vehicles, or full automation of vehicles. Its primary focus is on distracted driver detection within the context of human-operated vehicles.
2. Hardware Development:
    - The project does not encompass the design or development of specialized hardware, such as vehicle-mounted cameras, sensors, or onboard computer systems. It relies on existing hardware for data input.
3. Regulatory Changes:
    - While the project may address legal and ethical considerations, it does not aim to influence or initiate regulatory changes in traffic laws or standards. Compliance with existing regulations is assumed.
4. Distracted Pedestrian Detection:
    - The project's primary focus is on identifying distracted drivers. Detection of distracted pedestrians or other road users is not within its scope.

1.3 Quality Objective

Here make a mention of the overall objevtie that you plan to achive withou your testing

Some objectives of your testing project could be

- Ensure the system exhibits a high level of reliability, minimizing false positives and false negatives in distraction detection.
- Develop a system that can analyze and respond to potential distracted driving events in real-time, with a minimal latency of no more than 0.5 seconds.
- Consider accessibility standards to make the system inclusive and usable for a wide range of users, including individuals with disabilities.

1.4 Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like

- **QA Analyst** : Shruti Gupta
  The Quality Assurance (QA) Analyst conducted testing on software, websites, and other technical products to identify and resolve bugs, defects, and other potential issue.
- **Test Manager :** Ms. Shreela Pareek
  Managed all test processes, including test plans, resources, costs, timescales, test deliverables and traceability.
- **Configuration Manager :** Mrs. Neha Shukla
- **Developers :** Sarthak Tyagi, Utkarsh Mishra , Shruti Gupta
  Developed the model and trained it.
- **Installation Team :** Sarthak Tyagi, Utkarsh Mishra , Shruti Gupta , Shreela Pareek, Neha Shukla
  Responsible for smooth execution of the program

2 Test Methodology

2.1 Overview

The decision to adopt a Waterfall methodology for a project is typically based on specific project requirements, constraints, and organizational factors. Here are some common reasons for choosing the Waterfall methodology:

- Well-Defined Requirements: When the project has clearly defined and stable requirements that are unlikely to change significantly throughout the project's lifecycle. Waterfall is suitable when you can gather and document all the requirements up front.
- Low Uncertainty: If there is a high level of confidence in the project scope and objectives, and the technology and processes to be used are well-understood, Waterfall can be a good choice. It is less adaptable to uncertainty and change.
- Regulatory Compliance: In cases where the project needs to adhere to strict regulatory or compliance standards, Waterfall provides a structured and documented approach that can help meet these requirements.
- Large-Scale and Complex Projects: Waterfall can be beneficial for large-scale, complex projects where a comprehensive and detailed project plan is essential for successful execution.

2.2 Test Levels

Testing a Web Application Firewall (WAF) typically involves multiple test levels to ensure comprehensive coverage of its security features and effectiveness. These test levels can be organized as follows:

1. **Unit Testing:**
    - **Rule Validation:** Verify that individual security rules within the WAF are correctly configured and accurately detect or block specific types of attacks.
    - **Logging and Alerting:** Test that the WAF generates appropriate logs and alerts for specific rule violations.
2. **Integration Testing:**
    - **Rule Interaction:** Assess how different security rules interact when multiple rules are applied to the same request or response. Ensure they do not conflict or produce unintended outcomes.
    - **Communication with Other Security Components:** Test the WAF's ability to integrate with other security components in your infrastructure, such as intrusion detection systems (IDS) or load balancers.
3. **System Testing:**
    - **Rule Coverage:** Validate that the WAF provides comprehensive coverage for known vulnerabilities and attacks, including SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and other common web application threats.
    - **Custom Rule Testing**: Ensure that any custom rules configured to protect application-specific vulnerabilities are working as intended.

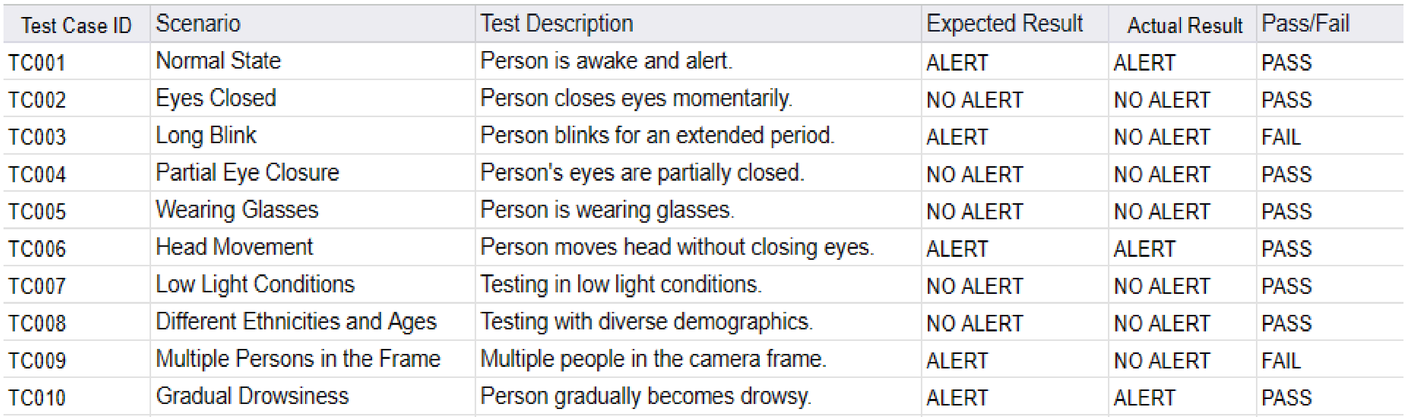
2.3 Test Completeness

Few criteria to check Test Completeness are:

- 100% test coverage
- All open bugs are fixed or will be fixed in next release

3.Test Deliverables

- Test Cases :



Test Case ID	Scenario	Test Description	Expected Result	Actual Result	Pass/Fail
TC001	Normal State	Person is awake and alert.	ALERT	ALERT	PASS
TC002	Eyes Closed	Person closes eyes momentarily.	NO ALERT	NO ALERT	PASS
TC003	Long Blink	Person blinks for an extended period.	ALERT	NO ALERT	FAIL
TC004	Partial Eye Closure	Person's eyes are partially closed.	NO ALERT	NO ALERT	PASS
TC005	Wearing Glasses	Person is wearing glasses.	NO ALERT	NO ALERT	PASS
TC006	Head Movement	Person moves head without closing eyes.	ALERT	ALERT	PASS
TC007	Low Light Conditions	Testing in low light conditions.	NO ALERT	NO ALERT	PASS
TC008	Different Ethnicities and Ages	Testing with diverse demographics.	NO ALERT	NO ALERT	PASS
TC009	Multiple Persons in the Frame	Multiple people in the camera frame.	ALERT	NO ALERT	FAIL
TC010	Gradual Drowsiness	Person gradually becomes drowsy.	ALERT	ALERT	PASS

4. BOUNDARY VALUES

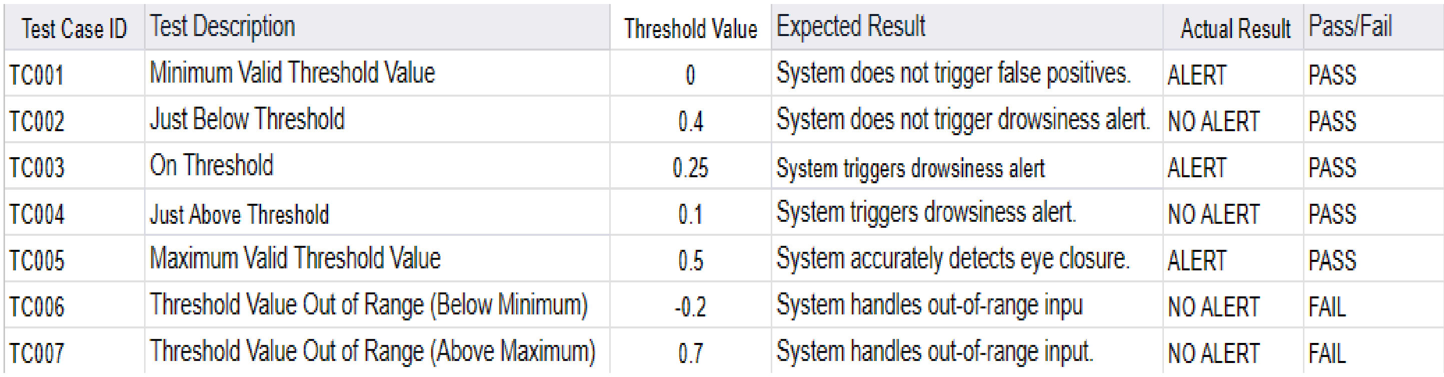
Minimum Value=0.0

Maximum Value=0.5

Norminal Value=(min+max)/2=0.25

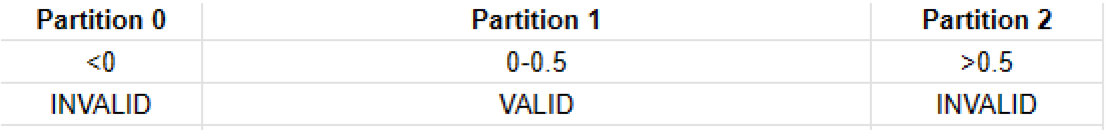
Minimum +0.1 = 0.1

Maximum – 0.1 =0.4



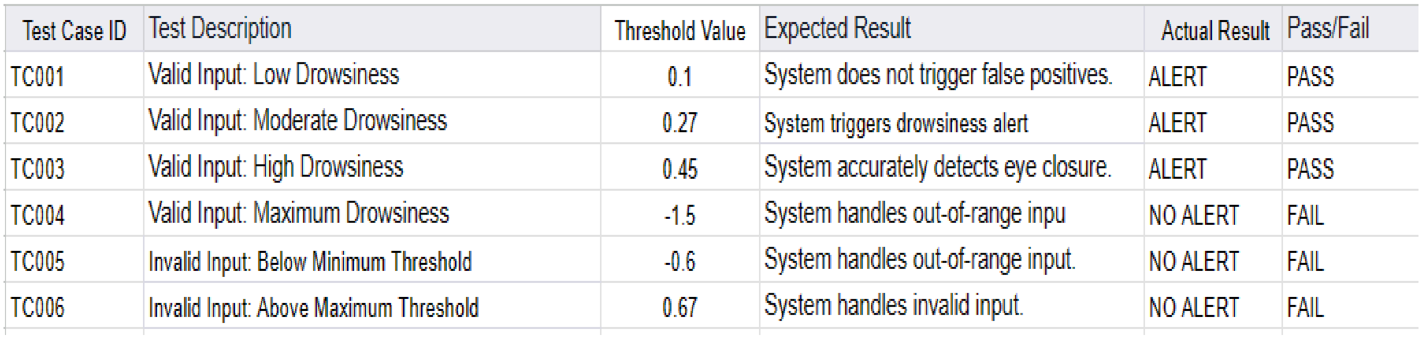
Test Case ID	Test Description	Threshold Value	Expected Result	Actual Result	Pass/Fail
TC001	Minimum Valid Threshold Value	0	System does not trigger false positives.	ALERT	PASS
TC002	Just Below Threshold	0.4	System does not trigger drowsiness alert.	NO ALERT	PASS
TC003	On Threshold	0.25	System triggers drowsiness alert	ALERT	PASS
TC004	Just Above Threshold	0.1	System triggers drowsiness alert.	NO ALERT	PASS
TC005	Maximum Valid Threshold Value	0.5	System accurately detects eye closure.	ALERT	PASS
TC006	Threshold Value Out of Range (Below Minimum)	-0.2	System handles out-of-range inpu	NO ALERT	FAIL
TC007	Threshold Value Out of Range (Above Maximum)	0.7	System handles out-of-range input.	NO ALERT	FAIL

5. EQUIVALENCE TABLE



Partition 0	Partition 1	Partition 2
<0	0-0.5	>0.5
INVALID	VALID	INVALID

TEST CASEs:



Test Case ID	Test Description	Threshold Value	Expected Result	Actual Result	Pass/Fail
TC001	Valid Input: Low Drowsiness	0.1	System does not trigger false positives.	ALERT	PASS
TC002	Valid Input: Moderate Drowsiness	0.27	System triggers drowsiness alert	ALERT	PASS
TC003	Valid Input: High Drowsiness	0.45	System accurately detects eye closure.	ALERT	PASS
TC004	Valid Input: Maximum Drowsiness	-1.5	System handles out-of-range inpu	NO ALERT	FAIL
TC005	Invalid Input: Below Minimum Threshold	-0.6	System handles out-of-range input.	NO ALERT	FAIL
TC006	Invalid Input: Above Maximum Threshold	0.67	System handles invalid input.	NO ALERT	FAIL

3 Resource & Environment Needs

3.1 Testing Tools

No Testing Tool is required. Manual Testing is done.

3.2 Test Environment

It mentions the minimum **hardware** requirements that will be used to test the Application.
The following **software's** are required in addition to client-specific software.

- Windows Operating System
- VS Code
- Camera and Microphone

4 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
cv2	Open CV
py	Python
imu	Imutils library