University of Regina | Go far, together.

IFS DriverAlert

Equivalence Class Testing

Date: Feb. 14, 2025

Equivalence Class Testing is a testing technique used to minimize the number of test cases while ensuring effective coverage. It works by dividing input data into equivalence classes, where each class represents a group of values. Below, we define the equivalence classes for the tested functions and document the corresponding test cases, expected outcomes, and actual results.

Table 1: Equivalence classes:

Test Functions	Variables Being Tested	Equivalence Classes
calculate_eye_ratio()	Eye coordinates	Invalid format, Valid format
calculate_mouth_ratio()	Lips coordinates	Invalid format, Valid format
calculate_turn_ratio()	Face coordinates	Invalid format, Valid format
detect_signs()	average_ratio_lips average_ratio_eyes average_ratio_turn	Below threshold, At threshold, Above threshold (For each variable)
	yawn_start_time closed_start_time turn_start_time	None, Valid timestamp, Expired timestamp (For each variable)
	yawn_detected, yawn_alert sleepy_detected, closed_alert turn_detected, turn_alert	False, True (For each variable)
alert_func()	yawn_start_time closed_start_time turn_start_time	None, Valid timestamp, Expired timestamp (For each variable)
	yawn_detected, yawn_alert sleepy_detected, closed_alert turn_detected, turn_alert	False, True (For each variable)
	yawn_audio1, yawn_audio2 eyes_audio1, eyes_audio2 turn_audio1, turn_audio2	Valid file path, Invalid file path
prepare_eye_for_model()	eye_img	Invalid format, Valid format
get_box()	ids	Invalid format, Valid format
	width, height	Positive/Negative values

Table 2: Equivalence class tests:

Test Method	Test Purpose	Expected Outcome	Actual Outcome	Test Pass
calculate_eye_ratio()	Test with valid input	The result should be the eye's ratio whether it is open or closed	The result was the eye's ratio whether it is open or closed	✓
calculate_eye_ratio()	Test with missing points	The result should be a default '100' if missing points	The result was a default '100'	✓
calculate_mouth_ratio()	Test with valid input	The result should be the mouth ratio whether it is open or closed	The result was the mouth ratio whether it is open or closed	✓
calculate_mouth_ratio()	Test with 0 horizontal distance	It should gives value error	Value error was raised	✓
calculate_turn_ratio()	Test with valid input	The result should be the turn ratio whether it is looking straight or away	The result was the turn ratio whether it is looking straight or away	✓
calculate_turn_ratio()	Test with 0 face width distance	It should gives value error	Value error was raised	✓
detect_signs()	Test open/close mouth at threshold (ratio and duration)	The yawn detection should be based on the ratio and duration	The yawn detection was based on the ratio and duration	✓
detect_signs()	Test open/close mouth below threshold (ratio and duration)	The yawn detection should be based on the ratio and duration	The yawn detection was based on the ratio and duration	✓
detect_signs()	Test open/close mouth above threshold (ratio and duration)	The yawn detection should be based on the ratio and duration	The yawn detection was based on the ratio and duration	✓
detect_signs()	Test open/close eye at threshold (ratio and duration)	The closed eye detection should be based on the ratio and duration	The closed eye detection was based on the ratio and duration	✓

detect_signs()	Test open/close eye below threshold (ratio and duration)	The closed eye detection should be based on the ratio and duration	The closed eye detection was based on the ratio and duration	✓
detect_signs()	Test open/close eye above threshold (ratio and duration)	The closed eye detection should be based on the ratio and duration	The closed eye detection was based on the ratio and duration	✓
detect_signs()	Test face turn at threshold (ratio and duration)	The turn detection should be based on the ratio and duration	The turn detection was based on the ratio and duration	✓
detect_signs()	Test face turn below threshold (ratio and duration)	The turn detection should be based on the ratio and duration	The turn detection was based on the ratio and duration	✓
detect_signs()	Test face turn above threshold (ratio and duration)	The turn detection should be based on the ratio and duration	The turn detection was based on the ratio and duration	✓
alert_func()	Test with the alert variables are set	Alert should be played	Alert was played	✓
alert_func()	Test with the alert variables are not set	Alert should not be played	Alert was not played	✓
prepare_eye_for_model()	Test with valid image input	The image should be converted to the custom shape	The image was converted to the custom shape	✓
prepare_eye_for_model()	Test with invalid/empty image types	It should gives value error	Value error was raised	✓
get_box()	Test with valid image values	It should output the values of box around the eye	The output was the values of box around the eye	✓
get_box()	Test with invalid IDs	It should gives value error	Value error was raised	✓
get_box()	Test height/width with negative values	It should gives value error	Value error was raised	✓