

## PLAGIARISM SCAN REPORT

Words	889	Date	September 18,2020
Characters	5741	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	48 Unique Sentences
------------------	----------------	-------------------------------	------------------------

## Content Checked For Plagiarism

5.2.2 Alert Generating system: - • Mount Pi camera on Vehicle-Model o Logic- To capture the road signs pi camera is attached to the vehicle which will help to capture some alert, warning and some precaution signals. • Collect dataset of signals and Train dataset by using Cascade Trainer-GUI tool o Logic - To achieve some good accuracy in result the real time data is being trained which will help to react quick and more accurate. • Import that trained model in program. o Logic - The trained data is imported in program to test it on some alert, warning signals. • Generate program for generating alert signals. o Logic - Finally it will generate some alert or give some instructions accordingly. 6. SOFTWARE TESTING 6.1 Unit test cases generation and its testing reports Test case No Test case Input Expected Output Actual Output Status 01 Ultrasonic wave transfer Ultrasonic wave Ultrasonic wave transfers successfully Ultrasonic wave transfers successfully Pass 02 Ultrasonic wave receives Ultrasonic wave Ultrasonic wave receives successfully Ultrasonic wave receives successfully Pass 03 Calculate time between transfer and receive wave Ultrasonic wave Time between wave transfer and wave receive calculated successfully Time between wave transfer and wave receive calculated successfully Pass 04 Capture image Camera module Capture image successfully Capture image successfully Pass 05 Call vehicle stop module Call to module Vehicle stop module working successfully Vehicle stop module working successfully Pass 6.2 Integration test cases generation and its testing reports Test case No Test case Input Expected Output Actual Output Status 01 Object detection (within threshold) Ultrasonic wave Object detected Successfully Object detected Successfully Pass 02 Object detection (not within threshold) Ultrasonic wave Object does not detect Successfully Object does not detect Successfully Pass 03 Distance calculation Ultrasonic wave Distance calculated successfully Distance calculated successfully Pass 04 Check captured image has traffic signal image or not. Captured image Image has traffic signal. Image has traffic signal. Pass 6.3 System test cases generation and its testing reports Test case No Test case Input Expected Output Actual Output Status 01 if object detected, Stop vehicle Ultrasonic wave, call to stop vehicle module Vehicle stopped successfully Vehicle stopped successfully Pass 02 if object not detected, go ahead with vehicle Ultrasonic wave Vehicle not stopped Vehicle not stopped Pass 03 Detect traffic signal from inputted image Captured image Traffic signal detected successfully Traffic signal detected successfully Pass 04 if traffic signal detected, Generate alert for traffic signal Captured image Generate alert for detected traffic signal Generate alert for detected traffic signal Pass 05 if traffic signal not detected, does not generate alert for traffic signal Captured image Dose not generate any alert for traffic signal Dose not generate any alert for traffic signal Pass 06 Detected traffic sign: - stop and generate alert for stop Captured image Generate alert for 'stop' signal successfully Generate alert for 'stop' signal successfully Pass 07 If two traffic signals detected: - Generate alerts for both signal Captured image Generate alerts for both traffic signals successfully Generate alerts for both traffic signals successfully Pass 7. Output Screen 7.1 Output-figure-1: - Starting page of module. We give one button to start system and another one to stop system. 7.2 Output-figure -2: - Here live distance are shown and according to that distance status of vehicle was shown. In current screenshot system shoes status: - 'safe' and color is green which indicate safe state. 7.3 Output-figure -3: - Following screenshot have distance in normal rage, so system shows vehicle status: - 'Normal' and color is yellow which indicate normal state. 7.4 Output-figure -4: - Following screenshot have distance in danger rage, so system shows vehicle status: - 'danger' and color is red which indicate danger state. 7.5 Output-figure -5: - Captured image inputted to the system and system analyze traffic signal, and display traffic sign name on screen and also gives voice alert according to sign. 8. PERFORMANCE ANALYSIS Traffic sign detection accuracy depends upon training of dataset. We provide large dataset for taring then it's given maximum accuracy. So, in point of accuracy the amount of dataset plays vital role. As first stap we provide 20 images and we got no much good accuracy. Then we go with increasing number of images, and analyse accuracy. After taking 200 positive and negative images we got good accuracy. In the end, we provide dataset with 500 images of each positive and negative, and we achieve better accuracy as compared previous. Following figure describes graph

images of each positive and negative, and we achieve better accuracy as compared previous. Following figure describes graph about dependency of dataset and accuracy: - Fig 8.1 dataset vs accuracy The above diagram shows the graph over accuracy for the number of images as a dataset. For below 100 images accuracy provides by the system is nearby 20%. As the number of images as datasets increases also accuracy increases. Testing is important step during whole lifecycle of system. We train our traffic sign detection on cascade trainer. Firstly, we test that model on cascade tester which is by default tester for cascade trainer. Then we deploy on actual model and analyze accuracy. Following graph elaborate about all observation of testing of traffic sign detection models. Fig.8.2 Testing Above line graph describes the relationship between testing on pc-tool and testing on the actual module. Testing on the actual module provides 5% to 10 % less accurate than testing on pc-tool. . .

Sources	Similarity
---------	------------