# Smart Eye: Real-Time Traffic Light Color Detection with Audio and Text Assistive Technology for Color Blind Drivers using YOLO Algorithm

A Thesis

Presented to

The Faculty of the College of Arts and Sciences

Iloilo Science and Technology University

La Paz, Iloilo City

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Computer Science

Eugene S. Alonzo

Kyle Lorace A. Espinosa

John Iris N. Quiambao

Jennifer C. Soldevilla

June 2024

APPROVAL SHEET

This study/project entitled SMART EYE: REAL-TIME TRAFFIC LIGHT COLOR DETECTION WITH AUDIO AND TEXT ASSISTIVE TECHNOLOGY FOR COLOR BLIND DRIVERS USING YOLO ALGORITHM, prepared and submitted by Eugene S. Alonzo, Kyle Lorace A. Espinosa, John Iris N. Quiambao, and Jennifer C. Soldevilla in partial fulfillment of the requirements of the degree BACHELOR OF SCIENCE IN COMPUTER SCIENCE is hereby recommended for approval.

LENNY M. AMAR, Ph. D.
*Adviser*

It has passed the final defense on December 19, 2023, and was approved by the Defense Committee on ______________with a grade of__________.

ERNEST ANDREIGH C. CENTINA, MSCS
*Chairman, Panel*

| | |
|---|---|
| CHRISTIAN REY J. INFANTE, MSCS<br>Panel Member | CHRISTIAN LESTER GEMINO, MSCS<br>Panel Member |

ERNEST ANDREIGH C. CENTINA, MSCS
BSCS Thesis Coordinator

Accepted and approved in partial fulfillment of the requirements for the degree BACHELOR OF SCIENCE IN COMPUTER SCIENCE.

| | |
|---|---|
| YVETTE G. GONZALES, D. Eng.<br>*Department Head, Computer Department* | TRACY N. TACUBAN, DIT<br>*Dean, College of Arts and Sciences* |

Date:_____________ Date:________________

*Acknowledgment*

The researchers wish to express their profound gratitude and sincere appreciation to all, who in one or many ways contributed their precious time and efforts and have helped in the success of this research paper. Special thanks to:

Almighty God, for He had showered upon his blessings for his unceasing guidance that lighted up the way of the researchers amidst the difficulties.

Dr. Lenny M. Amar, Thesis adviser, for her unlimited help, effort and time, valuable comments, and suggestions, without which the researchers would not have been able to finish this paper;

Mr. Ernest Andreigh C. Centina, Thesis Instructor, for his constructive criticism and suggestions during the defense which help consequently helped the researchers come up with a practical and beneficial inquiry;

Members of the panel, Mr. Christian Lester D. Gemino, and Mr. Christian Rey J. Infante, for their recommendations, helpful ideas, and comments, and for rendering their precious time and effort in reading and checking the documents;

Their classmates and friends who have been there throughout the challenges and success, for cooperation and wholehearted support;

To their family, who gave them undying support, guidance, love, and financial needs for the whole duration of the study; and

All those who are not mentioned here but who have somehow helped in the completion of the study. Thank you so much.

May the Almighty God richly bless all of you.

***Eugene S. Alonzo***
***Kyle Lorace A. Espinosa***
***John Iris N. Quiambao***
***Jennifer C. Soldevilla***

SMART EYE: REAL-TIME TRAFFIC LIGHT COLOR DETECTION WITH

AUDIO AND TEXT ASSISTIVE TECHNOLOGY FOR COLOR BLIND DRIVERS

USING YOLO ALGORITHM

BY

EUGENE S. ALONZO
KYLE LORACE A. ESPINOSA
JOHN IRIS N. QUIAMBAO
JENNIFER C. SOLDEVILLA
*Researchers*

DR. LENNY M. AMAR
*Adviser*

*ABSTRACT*

*Table of Contents*

*List of Figures*

*List of Tables*

*List of Appendices*

## CHAPTER 1

## Introduction

Chapter One is composed of seven parts: (1) Background of the Study, (2) General Objectives, (3) Specific Objectives, (4) Conceptual Framework, (5) Definition of Terms (6) Significance of the Study, and (7) Delimitation of the Study.

### Background of the Study

A genetic disease called color blindness makes it harder for someone to tell some colors from others. Color blindness affects 0.5% of women and 8% of males globally [1]. Color blindness can be simply defined as trouble in seeing or identifying colors like blue, green, and red. There are some rare cases where a person cannot see and identify any colors at all. A person with this syndrome also finds difficulties in differentiating the colors with shades. This syndrome is also called a color vision problem or color vision deficiency [2, 3].

Meanwhile, road traffic signals use a red, green, and yellow color code, and a person with abnormal color vision is confused about these colors. Motor vehicle tail and brake lights are red, the color that connotes danger and those with a protan color vision deficiency have a significantly lower than normal sensitivity to red light. They will have a reduced visual range for red traffic signals, taillights, and brake lights and may also fail to see lower-intensity red retroreflectors that mark out an otherwise unlit parked vehicle [2].

Laboratory studies of the ability of people with abnormal color vision to name the colors of simulated road traffic signals show that they often make errors naming the colors, with error rates typically between five to 10 percent, but for some colors under some conditions, as high as 20 to 30 percent. Their reaction times for naming the colors of the signals are significantly longer than those with normal color vision, reflecting their uncertainty in identifying the colors. This uncertainty is also evident in a reduced distance, from which drivers with abnormal color vision recognize signal colors. The reduction in recognition distance is 60 to 70 percent that of drivers with normal color vision, even when there is a positional cue provided by only one signal light being illuminated in a vertical array of three signals. Color recognition distances drop to 30 percent of normal when there is no positional cue [3].

Therefore, this paper will work on developing an application that can detect color, specifically traffic light colors. This is to help drivers who are having difficulty in identifying traffic light colors or are partially blind. Road and traffic sign recognition is one of the biggest fields in the Intelligent Transport System (ITS) due to the importance of road signs and traffic signals in daily life. They define a visual language that can be interpreted by the drivers. They illustrate the current traffic situation on the road, indicate dangers and difficulties drivers may encounter, give warnings to them, and help them with their navigation by providing useful information that makes driving safe and convenient. This paper will utilize the YOLO algorithm for real-time object detection, specifically traffic lights, and in detecting the colors of the traffic lights.

You Only Look Once (YOLO) suggests utilizing an end-to-end neural network that simultaneously predicts bounding boxes and class probabilities. Using a completely distinct approach to object detection, YOLO outperformed existing real-time object detection algorithms and produced cutting-edge findings. Thus, YOLO will be used in detecting traffic light colors with their corresponding status.

## Objectives of the Study

Generally, this study aims to develop Smart Eye: Real-Time Traffic Light Color Detection with Audio and Text Assistive Technology for Color Blind Drivers using the YOLO Algorithm.

Specifically, this study has the following specific objectives.

1. To capture a real-time video feed.
2. To detect traffic lights and to identify traffic light colors using the YOLOv8 algorithm.
3. To provide assistance instruction in audio and text.
4. To evaluate the software using ISO 25010.

**Conceptual Framework**

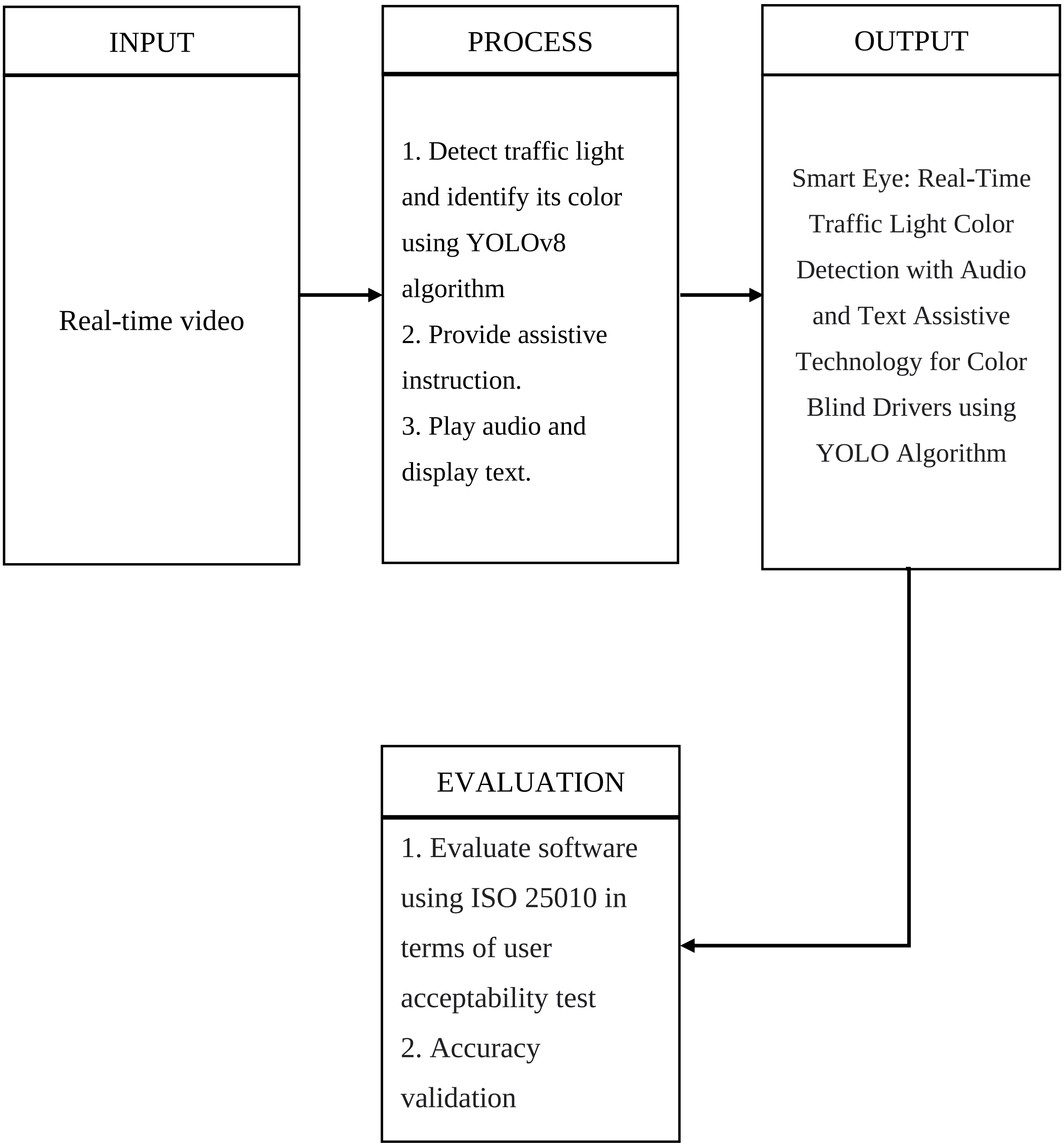



*Figure 1. Conceptual Framework of the Study*

The conceptual framework of the study is shown in Figure 1. The framework illustrated the flow of data entered in the software and how the data are processed. The figure is divided into three components: input, process, and output.

The web application will detect the traffic lights in real-time video footage using the web camera attached and at the same time it will determine the color of traffic lights to improve visibility for drivers with color blind. Then it will provide an audio and text alert to notify the user of the traffic light's color.

The output is a web-based application that will assist color-blind drivers using image processing which will be evaluated using ISO 25010 quality standard.

## Definition of Terms

**Assistive Technology (AT).** AT is any device, software, or equipment that helps people with disabilities learn, communicate, or function better. AT can be as high-tech as a computer [4].

In this study, assistive technology refers to the mobile application that will be developed by researchers to help people with color blindness.

**Data preprocessing**. It refers to the cleaning, transforming, and integrating of data to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.

In this study, data preprocessing refers to the training of the datasets in Google Colab and Robloflow, specifically the traffic lights datasets that will be used for the object detection of the system. [25]

**Image Processing.** A set of computational techniques for analyzing, enhancing, compressing, and reconstructing images [5].

In this study, this term refers to the use of algorithms and techniques to analyze, manipulate, and enhance digital images. It involves the processing of images captured by cameras, scanners, or other devices to extract information, improve quality, or transform the images to meet specific requirements.

**ISO 25010.** The study adopted ISO 25010, a quality model that is the cornerstone of a product quality evaluation system [6].

In this study, ISO 25010 refers to providing consistent terminology for specifying, measuring, and evaluating system and software product quality.

**Real-time**. It is the actual time during which something takes place.

In this study, it refers to a specific requirement and constraint that videos will be captured in real-time [26].

**YOLO Algorithm.** You Only Look Once (YOLO) is one of the most popular model architectures and object detection algorithms. It uses one of the best neural network architectures to produce high accuracy and overall processing speed, which is the main reason for its popularity.

In this study, YOLOv8 will used as the algorithm for detecting the traffic light object and traffic light color in a real-time web-based application setup created by the researchers.

## Significance of the Study

This study aims to benefit the following:

**Color Blind People.** This will help them to see the color correctly, assist them while driving, and will help them to recognize the correct color of the traffic light signals. This technology will serve as an aid for them to be able to do things like people with normal vision.

**Partially Blind People.** It will assist them while driving, and help them to recognize the correct color of the traffic light signals, especially during nighttime when they cannot completely see. This technology will help them in driving safely with their speech output.

**Students.** This will help interested students to have a better understanding of how color-blind works and how it is important to have ways to minimize its effects on a person.

**Developers.** This idea will help developers build and develop ways for the improvement of the existing methods and systems.

**Future researchers.** It will serve as their basis and give them background about the possible research they might conduct in the future.

### Delimitation of the Study

The focus of this study is to develop a web-based application that can detect traffic lights and their color and produce an audio and text output. To achieve the expected output, this paper will make use of a YOLOv8 algorithm method to detect the traffic light and identify the color status of the traffic lights every second.

This paper will only cater to persons with partial and total color blindness. For its color and object detection feature, traffic lights will be its main focus with consideration to the car windows' tint level with a 20% tint level as the lowest.

In case there is more than one traffic light with different traffic light colors and directions within the range of detection, the first traffic light detected will be its main focus and it will disregard the other ones. Our study will only be limited to a normal traffic light intended for two-lane roads only.

Moreover, it will be deployed as a web-based that can sustain a real-time-based video.

# CHAPTER 2

## Review of Related Literature

This chapter provides information that supports the development of Real-Time Traffic Light Color Detection with Audio and Text Assistive Technology Mobile Application for Color Blind Drivers using YOLO and Color Thresholding Algorithms. Although there is a lot of literature available, this study will only focus on the following fields/topics that are relevant to the development of the study. It has the following parts: (1) Yolo Algorithm , (2) Optimizing YOLO Performance for Traffic Light Detection and End-to-End Steering Control for Autonomous Vehicles in Gazebo-ROS2, (3) Road and Traffic Sign Color Detection and Segmentation - A Fuzzy Approach, (4) Detection and Classification of Vehicles by Using Traffic Video Based on Yolov8, (5) Color Detection System for Diamond Sorting Using Machine Learning, (6) Computer Vision for Color Detection, (7) Detection and Localization of Traffic Lights using YOLOV3 and Stereo Vision, (8) Traffic Light Detection and Recognition for Autonomous Vehicles, (9) Traffic Light Detection and Recognition in Autonomous Vehicles (AVs), (10) Real-time Traffic Monitoring System Based on Deep Learning and YOLOv8 (11) Specific Color Detection in Images using RGB Modelling in MATLAB, (12) Real-Time Traffic Sign Detection Using Color and Shape-Based Features, (13) Real-time Industrial Application of Single Board Computer-based Color Detection System, (14) Real-Time Traffic Light Detection with Adaptive Background Suppression Filter, (15) ISO 25010, and (16) Synthesis.

### YOLO Algorithm

According to the study, [7] You Only Look Once (YOLO) suggests utilizing an end-to-end neural network that simultaneously predicts bounding boxes and class probabilities. It varies from the strategy used by earlier object detection algorithms, which used classifiers as detectors. Using a completely distinct approach to object detection, YOLO outperformed existing real-time object detection algorithms and produced cutting-edge findings. Whereas Faster RCNN and similar algorithms use the Region Proposal Network to identify potential regions of interest before performing recognition on each of those regions independently, YOLO conducts all of its predictions using a single fully connected layer. While YOLO only requires one iteration, methods that use region proposal networks require numerous iterations for the same image.

### Optimizing YOLO Performance for Traffic Light Detection and End-to-End Steering Control for Autonomous Vehicles in Gazebo-ROS2

This research [8] introduces an enhanced version of the You Only Look Once (YOLO) object detection algorithm tailored for detecting traffic lights and incorporates end-to-end steering control for lane-keeping within a simulation environment. The study evaluates the efficacy of YOLOv5, YOLOv6, YOLOv7, and YOLOv8 models in detecting traffic light signals. Among these models, YOLOv8 demonstrates superior performance, achieving the highest mean Average Precision (mAP) at 98.5%. The algorithm underwent validation using a simulated Gazebo environment within the Robot Operating System 2 (ROS2) for an autonomous vehicle model.

### Road and Traffic Sign Color Detection and Segmentation - A Fuzzy Approach

According to Fleyeh [9], in 2015, a paper showed a new method for color detection and segmentation used for traffic signs. The method is based on invoking the HSV color space and uses hue and saturation to generate a binary image containing the road sign of a certain color. The images were taken by a digital camera mounted in a car. The system described in the previous section uses a fuzzy inference system to achieve color detection and segmentation. It is tested on more than one hundred images taken in different light conditions. The system is very robust, and it can detect all the colors described in the sample images. Furthermore, it was tested on color images from other European countries, and it could detect all the test signs.

### Detection and Classification of Vehicles by Using Traffic Video Based on Yolov8

This study [10], proposes an adaptive traffic signalization method utilizing YOLOv8, a CNN architecture, instead of conventional fixed-duration systems employed in traffic signalization which achieved an average score of 91% for object detection and classification performance. The training of the deep neural network involved utilizing real traffic images from 10 different intersections. Image labeling was carried out with the creation of 7 classes: car, bike, SUV, van, bus, truck, and person. Determining the number of vehicles waiting at red lights was accomplished by extracting coordinate information from bounding boxes associated with vehicles in Python-programmed software. The duration of the green light was computed based on vehicle numbers derived from classification and the scores assigned to each vehicle class. Consequently, the green light duration is dynamic, varying according to the number of vehicles present at intersection branches. It is predicted that with the proposed method, the waiting time at the red light, fuel consumption at the waiting moment, air pollution, stress level of drivers, accidents and many traffic problems will decrease.

**Color Detection System for Diamond Sorting Using Machine Learning**

According to the research,[11] the concept of color detection for diamond sorting uses machine learning to teach the system how to recognize color in RGB space and predict its name and values. The program gathers images from the dataset and analyses the diamond's color attributes. Diamond classification is based on models created with the aid of machine learning techniques and image analysis algorithms. OpenCV is used for computer vision and to process the picture that has to be identified, while Pandas, a Python library, collects and analyses the data from the dataset to provide a suitable forecast of the color name and values. The Color Detection technique is implemented in image processing using the Color Detection and Segmentation methodology, and the result may be seen manually by double-clicking on any section of the picture to determine the color of the diamond or automatically once the image is processed for quicker and easy sorting. The final result looks to be a success, as it can predict or recognize a color with 95% accuracy.

The study shows color detection for diamond sorting by using machine learning. It teaches the system to be able to recognize color. It achieved a 95% accuracy for the color detection. The common features that have been used in this study are that researchers have come up with a Color Detection. They also used OpenCV with a library of Pandas for their color detection.

### Computer Vision for Color Detection

According to the study,[12] conducted in 2021, the idea of teaching a computer to detect and define a color well enough to have useful applications was published. The detection algorithm proposed uses the advantage of the camera and fed in data to detect even the color based on RGB values. The algorithm involved calls on a function that runs loops on readjusting the distance based on the nearest match. The color detection done using computer vision has been achieved. The GUI has been successful in giving the users an option to choose from thus returning the correct RGB values when the picture is double-clicked over a certain color. Using the algorithm of mapping RGB values by finding the shortest distance and managing to mean in the three, the accuracy calculated by the error in RGB distance has hit 97%.

### Detection and Localization of Traffic Lights using YOLOV3 and Stereo Vision

According to Omar, I. Lee, G. Lee, and Park [13], a study focuses on traffic light distance measurement using stereo cameras which is a very important and challenging task in the image processing domain, where it is used in several systems such as Driving Safety Support Systems (DSSS), autonomous driving and traffic mobility. The study proposes an integrated traffic light distance measurement system for self-driving based on stereo image processing. Therefore, an algorithm to spatially locate the detected traffic light is required in order to make these detections useful. In this study, an algorithm to detect, and classify the traffic light colors and spatially locate traffic lights are integrated. Detection and color classification are made simultaneously via YOLOv3, using RGB images. 3D traffic light localization is achieved by estimating the distance from the vehicle to the traffic light, by looking at detector 2D bounding boxes and the disparity map generated by stereo camera. Moreover, Gaussian YOLOv3 weights based on KITTI and Berkeley datasets have been replaced with the COCO dataset. Therefore, a detection algorithm that can cope with mislocalizations is required in autonomous driving applications. The study proposes an integrated method for improving detection accuracy and traffic light color classification while supporting a real-time operation by modelling the bounding box (bbox) of YOLOv3. The obtained results show fair results within 20 meters away from the sensor, while misdetection and classification appeared at a further distance.

**Traffic Light Detection and Recognition for Autonomous Vehicles**

According to Mu, Xinyu, Deyi, Tianlei, and Lifeng, [14] traffic light detection and recognition is essential for autonomous driving in urban environments. A camera-based algorithm for real-time robust traffic light detection and recognition was proposed, and especially designed for autonomous vehicles. Although the current reliable traffic light recognition algorithms operate well, most of them are mainly designed for detection at a fixed position and the effect on autonomous vehicles under real-world conditions is still limited. Some methods achieve high accuracy on autonomous vehicles, but they can't work normally without the aid of high-precision maps. The authors presented a camera-based algorithm for the problem. The image processing flow can be divided into three steps, including pre-processing, detection, and recognition. Firstly, red-green-blue (RGB) color space is converted to hue-saturation-value (HSV) as the main content of pre-processing. In the detection step, the transcendental color threshold method is used for initial filtering, meanwhile, the prior knowledge is performed to scan the scene in order to quickly establish candidate regions. For recognition, the article uses a histogram of oriented gradients (HOG) features and a support vector machine (SVM) to recognize the state of the traffic light.

### Traffic Light Detection and Recognition in Autonomous Vehicles (AVs)

According to the study of Edward Swarlat Dawam and X. Feng [15], first, they developed a light controller and a dataset builder script. The light controller and dataset builder script were then used to build a dataset of traffic lights with different lights activated. Bounding boxes were annotated on the traffic light dataset using dlib's imglab software. The dataset uses the HOG with Linear SVM object detector. An RGB histogram approach is adopted to train a logistic regression model on the feature vector data to recognize which light is "on" among the training images. Finally, a robot control script is developed and tested. The script uses both the object detector and color recognizer for its detection and recognition. The results show 89% accuracy in identifying a red-yellow-green traffic light under extreme environmental conditions.

### Real-time Traffic Monitoring System Based on Deep Learning and YOLOv8

This study [16] introduces a real-time traffic monitoring system utilizing advanced deep learning techniques, employing the state-of-the-art You Only Look Once v8 algorithm. The system leverages the algorithm's capabilities for vehicle detection, classification, and segmentation, offering essential traffic information such as vehicle counting, classification, speed estimation, and size estimation. The proposed system comprises five stages: preprocessing, encompassing tasks like camera calibration, ROI calculation, and source video preparation; vehicle detection, utilizing a convolutional neural network model to locate vehicles in video frames; tracking, employing the ByteTrack algorithm to track identified vehicles; speed estimation, which estimates the speed of tracked vehicles; and size estimation, providing an estimate of vehicle dimensions. Researchers used a system running on the Nvidia GTX 1070 GPU and the results indicate that the detection and tracking stages achieve an average accuracy of 96.58% with an average error of 3.42%. The vehicle counting stage attains an average accuracy of 97.54% with a 2.46% average error, the speed estimation stage achieves an average accuracy of 96.75% with a 3.25% average error, and the size estimation stage shows an average accuracy of 87.28% with a 12.72% average error.

**Specific Color Detection in Images using RGB Modelling in MATLAB**

According to the research [17], the study gives an approach to recognizing colors in a two-dimensional image using the color thresh-holding technique in MATLAB with the help of an RGB color model to detect a selected color by a user in an image. The methods involved for the detection of color in images are the conversion of three dimensional RGB image into a grayscale image and then subtracting the two images to get two-dimensional black and white images, using the median filter to filter out noisy pixels, using connected components labeling to detect connected regions in binary digital images and use of bounding box and its properties for calculating the metrics of each labeled region. Further, the color of the pixels is recognized by analyzing the RGB values for each pixel present in the image. The algorithm is implemented using the image processing toolbox in MATLAB. The results of this implementation can be used in security applications like spy robots, object tracking, segregation of objects based on their colors, and intrusion detection.

In this study, the result of the detection depends on the threshold value that has been set for the images. The major problem with thresholding is that it considers only the intensity values of the pixels and does not take into consideration any relationship between them. Sometimes extra pixels are detected that are not part of the desired region, and with an increase in noise, these errors increase. In the method of labeling connected components for image segmentation, there is one problem, if overlapping objects are present in an image, then it will consider it as only one object.

### Real-Time Traffic Sign Detection Using Color and Shape-Based Features

According to this study [18], the paper uses a Support Vector Machine (SVM) to retrieve candidate regions of traffic signs in real-time video processing. Instead of processing each pixel, this approach utilizes a block of pixels as an input vector of SVM for color classification, where the dimension of each vector can be extended by a group of neighboring pixels. This helps to handle the diversification of data on both training and testing samples. After that, Hough transforms, and contour detection are applied to verify the candidate regions by detecting the shape of a circle and triangle. The experimental results are highly accurate and robust on the testing database, where samples are recorded in various states of the environment. Figure 3 presents the steps of the algorithm to detect traffic signs in real-time video processing. The method is based on color detection and segmentation using SVM on blocks of pixels.

The paper presented a real-time processing method of traffic-sign detection to apply in autonomous driving systems. The proposed method utilized linear SVM to classify color by a low complexity (average 23 ms per frame). After that shape matching has been applied to eliminate positive errors. 92.91 percent detection accuracy was achieved, and it has been applied to a real-time autonomous driving system, where the maximum speed of a car is limited to 30 km per hour. In the near future, it is foreseen that detection and recognition processes will be combined to generate a vision-based system of guidance and warning for autonomous driving systems.

### Real-time Industrial Application of Single Board Computer-based Color Detection System

According to this study [19], real-time industrial application of single-board computer-based color detection systems is realized. In this system, BeagleBoard-xM as a single board computer, a USB camera, a conveyor belt, and an LCD7 touch screen are used. OpenCV is used as an image-processing library in this color detection system. The main goal of this study is to define the number of different colored packages passing on the conveyor belt according to their color. Then, real-time results of the number of packages and the total package number are displayed on the LCD7 touchscreen. At the same time, the USB camera image of the related package on the conveyor belt is monitored on the same touch screen. If no image of any packages is taken by the USB camera for 60 seconds, the system is turned off.

**Real-Time Traffic Light Detection with Adaptive Background Suppression Filter**

This study [20], proposes a novel vision-based traffic light detection method for driving vehicles, which is fast and robust under different illumination conditions. The proposed method contains two stages: the candidate extraction stage and the recognition stage. In the candidate extraction stage, it was proposed that an adaptive background suppression algorithm highlight the traffic light candidate regions while suppressing the undesired backgrounds. In the recognition stage, each candidate region is verified and is further classified into different traffic light semantic classes. Then the method was evaluated on video sequences (more than 5000 frames and labels) captured from urban streets and suburban roads in varying illumination and compared them with other vision-based traffic detection approaches. The experiment shows that the proposed method can achieve the desired detection result with high quality and robustness; simultaneously, the whole detection system can meet the real-time processing requirement of about 15 fps on video sequences.

Common features that have been used in this study that researchers have come up with are the detection of traffic lights in a real-time environment. The related study used 15 fps for its camera.

**ISO 25010**

The study adopted ISO 25010 [21], a quality model that is the cornerstone of a product quality evaluation system. The quality model determines which quality characteristics will be considered when evaluating the properties of a software product. The quality of a system is the degree to which the system satisfies the stated and implied needs of its various stakeholders and thus provides value. The product quality model defined in ISO/IEC 25010 comprises the eight quality characteristics shown in the following.

- Functional Suitability: Refers to how well a product or system is able to provide functions that meet the stated and implied needs.
- Functional Completeness: Refers to the set of functions that covers all of the specified tasks and user objectives.
- Functional Correctness: Refers to how well a product or system provides the correct results with the needed degree of precision.
- Functional Appropriateness: Refers to how well functions are able to accomplish specific tasks and objectives.

**Reliability**

Reliability refers to how well a system, product, or component performs specific functions under specified conditions.

- Maturity: Refers to how well a system, product, or component is able to meet your needs for reliability.
- Availability: Refers to whether a system, product, or component is operational and accessible.
- Fault Tolerance: Refers to how well a system, product, or component operates despite hardware and/or software faults.

- Recoverability: Refers to how well a product or system can recover data in the event of an interruption or failure.

**Performance Efficiency**

Performance Efficiency refers to the performance related to the number of resources used.

- Time Behavior: Refers to the response and processing times, and throughput rates of a product or system while it's performing its functions.
- Resource Utilization: Refers to the amounts and types of resources used by a product or system while performing its functions.
- Capacity: Refers to the maximum limits of a product or system parameter.

**Usability**

Usability refers to how well a product or system can be used to achieve specific goals effectively, efficiently, and satisfactorily.

**Appropriateness Recognizability**

Refers to how well you can recognize whether a product or system is appropriate for your needs.

- Learnability: Refers to how easy it is to learn how to use a product or system.
- Operability: Refers to whether a product or system has attributes that make it easy to operate and control.
- User Error Protection: Refers to how well a system protects users against making errors.
- User Interface Aesthetics: Refers to whether a user interface is pleasing.
- Accessibility: Refers to how well a product or system can be used with the widest range of characteristics and capabilities.

**Security**

Security refers to how well a product or system protects information and data from security vulnerabilities.

- Confidentiality: Refers to how well a product or system is able to ensure that data is only accessible to those who have authorized access.
- Integrity: Refers to how well a system, product, or component is able to prevent unauthorized access and modification to computer programs and/or data.
- Non-repudiation: Refers to how well actions or events can be proven to have taken place.
- Accountability: Refers to the actions of an unauthorized user can be traced back to them.
- Authenticity: Refers to how well the identity of a subject or resource can be proved.

**Compatibility**

Compatibility refers to how well a product, system, or component can exchange information as well as perform its required functions while sharing the same hardware or software environment.

- Co-existence: Refers to how well a product can perform its required functions efficiently while sharing a common environment and resources with products, without negatively impacting any other product.
- Interoperability: Refers to how well two or more systems, products, or components are able to exchange information and use that information.

**Maintainability**

Maintainability refers to how well a product or system can be modified to improve, correct, or adapt to changes in the environment as well as requirements.

Modularity: Refers to whether the components of a system or program can be changed with minimal impact on the other components.

- Reusability: Refers to how well an asset can be used in more than one system.
- Analyzability: Refers to the effectiveness of an impact assessment on intended changes. In addition, it also refers to the diagnosis of deficiencies or causes of failures, or to identify parts to be modified.
- Modifiability: Refers to how well a product or system can be modified without introducing defects or degrading existing product quality.
- Testability: Refers to how effective the test criteria is for a system, product, or component. In addition, it also refers to the tests that can be performed to determine whether the test criteria have been met.

**Portability**

Portability refers to how well a system, product, or component can be transferred from one environment to another.

- Adaptability: Refers to how well a product or system can be adapted for different or evolving hardware, software, or other usage environments.
- Installability: Refers to how successfully a product or system can be installed and/or uninstalled.
- Replaceability: Refers to how well a product can replace another comparable product.

## Synthesis

The researchers believed that the literature and studies presented in this paper are related and somehow similar to the current study. Previous studies have shown the value of color and traffic light detection as an assistive tool for drivers with difficulty in identifying traffic lights and traffic light colors.

According to Fleyeh[10], the system uses a fuzzy inference system to achieve color detection and segmentation. It is tested on more than one hundred images taken in different light conditions. The system is very robust, and it can detect all the colors described in the sample images. Furthermore, it was tested on color images from other European countries, and it could detect all the test signs.

The other key point for further study is the effect of reflections on the stability of the hue in outdoor images. Combining these results with shape recognition of the road signs, and pictogram recognition, which are parts of the future work, will provide a good means to build a complete system that supplies drivers with information about the signs in real-time as part of the intelligent vehicle.

The study [8], introduces an enhanced version of the You Only Look Once (YOLO) object detection algorithm tailored for detecting traffic lights and incorporates end-to-end steering control for lane-keeping within a simulation environment. The study evaluates the efficacy of YOLOv5, YOLOv6, YOLOv7, and YOLOv8 models in detecting traffic light signals. Among these models, YOLOv8 demonstrates superior performance, achieving the highest mean Average Precision (mAP) at 98.5%. The algorithm underwent validation using a simulated Gazebo environment within the Robot Operating System 2 (ROS2) for an autonomous vehicle model.

It is found in the study of Mu, Xinyu, Deyi, Tianlei, and Lifeng, [14] that a camera-based algorithm for real-time robust traffic light detection and recognition was proposed, and specially designed for autonomous vehicles. The authors presented a camera-based algorithm for the problem. It is found that image processing flow can be divided into three steps, including pre-processing, detection, and recognition. Firstly, red-green-blue (RGB) color space is converted to hue-saturation-value (HSV) as the main content of pre-processing. In the detection step, the transcendental color threshold method is used for initial filtering, meanwhile, the prior knowledge is performed to scan the scene in order to quickly establish candidate regions. For recognition, this article uses a histogram of oriented gradients (HOG) features and a support vector machine (SVM) to recognize the state of the traffic light.

Real-Time Traffic Light Color Detection with Audio and Text Assistive Technology for Color Blind Drivers using YOLO Algorithm is a cutting-edge solution designed to help people with color blindness navigate traffic lights. The program increases the display of the traffic signal to make it more visible and identifiable for those with protanopia and utilizes computer vision technology to determine the color of traffic lights. This synthesis entails fusing a variety of methods and technologies. First, a web camera mounted on computer device can be used to determine the color of traffic signals using computer vision. This is accomplished using methods for image processing that determine first the traffic light in the roads and then the color of the traffic light by analyzing the pixel values of the image. Finally, it will produce a text and audio output that will refresh/update every seconds to give a notification to the driver.

# CHAPTER 3

## Methodology

This chapter discusses the Methodology of the study, which includes (1) Project Description, (2) Project Development, (3) Requirement Analysis, (4) Design Specification, (5) Testing and Evaluation, (6) Data Gathering Instrument, (7) Validity of the Instrument, (8) Respondent of the Study and (9) Data Processing and Statistical Tools. This will explain the application features and show how the algorithm within the application was developed.

### Project Description

The study Real-Time Traffic Light Color Detection with Audio and Text Assistive Technology for Color Blind Drivers using YOLO Algorithm is a mobile application that can assist colorblind people as well as partially blind. This will help them navigate their surroundings, especially when they are driving, as it has a feature that can detect traffic lights and identify their colors. The application developed will help people who are partially blind or those who are having difficulty in detecting traffic light colors. The main objectives of this study are to: develop a mobile application intended for a real-time live video setup; detect traffic light signals; produce an audio every second; and provide a text caption as an output. When used in a car, the car window tint levels will be observed, with 20% as the lowest. The application will be deployed on Android phones only with Android version 9 and up.

Existing solutions include color enhancement for uploaded images, traffic light color detection using devices, and color identification for uploaded videos and images. This study aims to innovate the existing solution by creating a mobile application that can take video input in a real-time setting, detect traffic lights, identify traffic light colors, and produce an audio and text output. With this, it will offer assistance to the target market more efficiently and conveniently.

**Development Process - Agile Methodology**

Software development is flexible and iterative thanks to the agile methodology. It places a strong emphasis on cooperation and communication between the product owner, the development team, and the end users. Agile divides the project into manageable, brief iterations, or sprints, which usually run 1-4 weeks. The team provides a working product increment that enhances the application after each sprint.

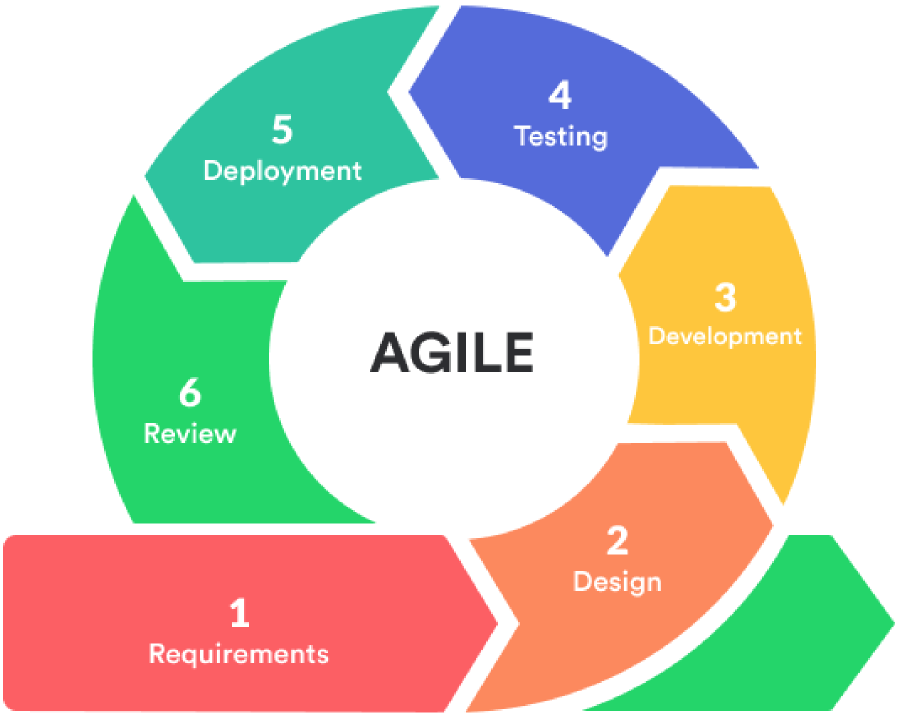



*Figure 2. Agile Methodology Diagram*

Figure 2 above shows the Agile Software Development Life Cycle method diagram used in conducting this study. Agile Methodology is composed of six phases namely, Requirements, Design, Development, Testing, Deployment, and Review/Maintenance. Each phase has a different function intended and is very crucial for the conduct of this study.

Agile methodology can be utilized in the following ways for a traffic light detector and recoloring in a real-time video application:

**Requirements:** The researchers work together to define the project's objectives, parameters, and specifications. Understanding the video input format, selecting the precise traffic light patterns to detect, and specifying the application's output format are all included in this.

**Design:** The group created an overall architecture, set of algorithms, and set of technologies for the application. The design must be adaptable enough to support alterations and enhancements based on user feedback.

**Development:** To construct the application incrementally, the development team will work in brief sprints. For instance, the team might concentrate on identifying the traffic lights in the first sprint before switching to recoloring them in the second. A workable product increment that improves the application should be delivered at the end of each sprint.

**Testing:** To make sure the application satisfies the requirements; it will be tested at each iteration. Functional and non-functional testing, such as usability and performance testing, should both be done. Any flaws or problems discovered during testing must be fixed in the following sprint.

**Deployment**: After each sprint is finished, the application will be released into production. If necessary, deployment can also be carried out progressively, for instance by deploying software at the end of each sprint to a test environment for user acceptance testing.

**Review/Maintenance:** Based on user input and evolving specifications, the team will continue to maintain and develop the application. This might entail enhancing functionality, resolving bugs, or adding new features.

In conclusion, Agile is best for making this application because it provides flexibility and adaptability. The Agile technique is generally well-suited for the traffic light detector and recoloring in a real-time video application. The team can respond fast to changes in requirements or user needs because of the brief sprints and constant feedback loops. The iterative strategy also allows the team to provide a working product increment at the end of each sprint, which can aid in gaining stakeholders' trust and confidence.

**Requirement Analysis**

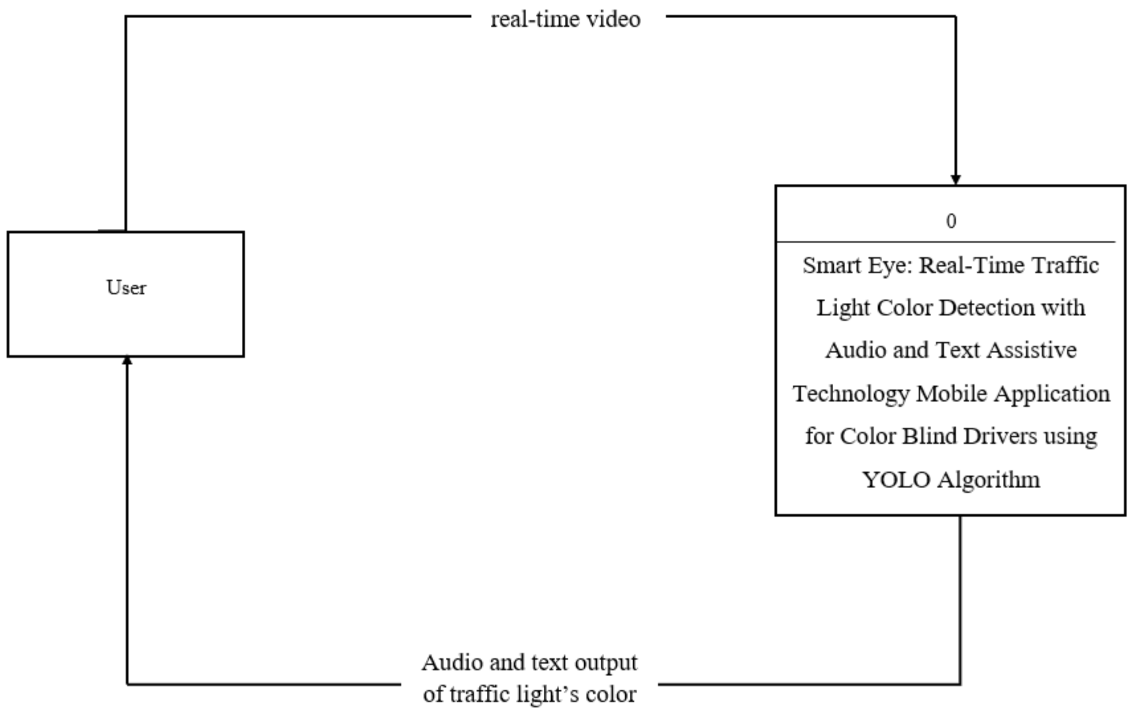



*Figure 3. Context Diagram of the Study*

The context diagram of Smart Eye: Real-Time Traffic Light Color Detection with Audio and Text Assistive Technology for Color Blind Drivers using YOLO Algorithm, as shown in Figure 3, illustrates the actual flow of the data coming from the user. In this case, the mobile software will be represented by Smart Eye: Real-Time Traffic Light Color Detection with Audio and Text Assistive Technology for Color

Blind Drivers using YOLO Algorithm, while the user is the representation of the external entity with whom they will interact with the software. In between the external entity and the process is the data flow that indicates the communication between the entity and the software. In the diagram, the software accepts the real-time video data from the user's mobile camera and will process it for traffic light color identification and produce audio and text output if a traffic light is detected.

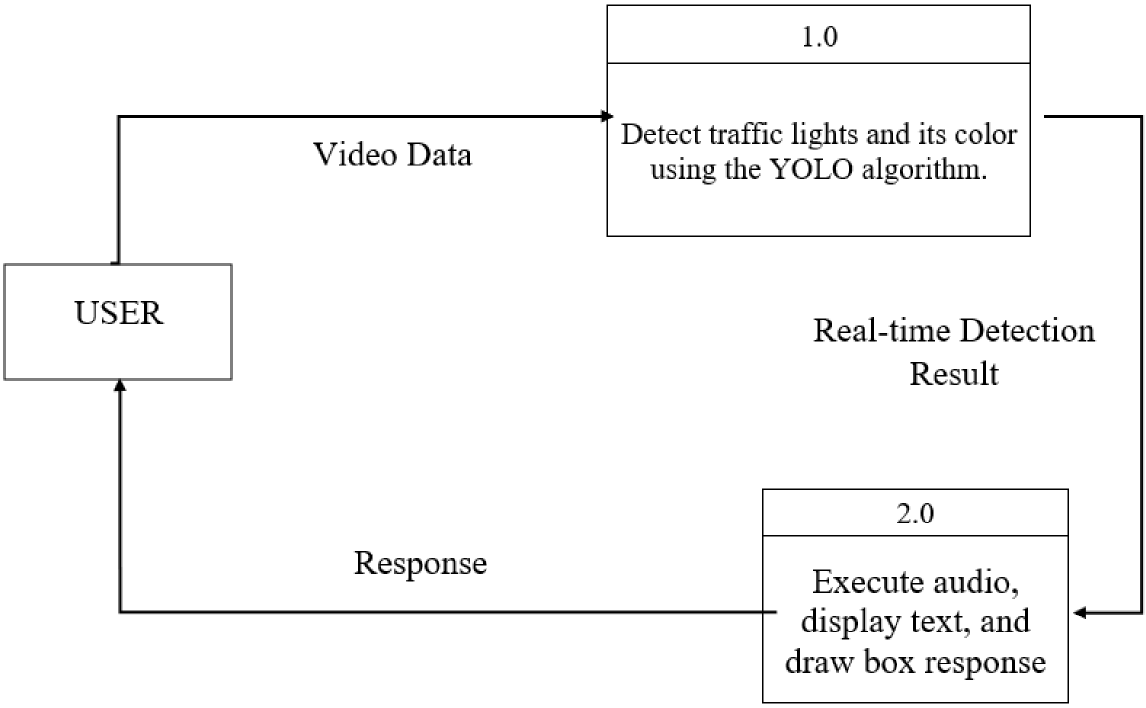



*Figure 4. Data Flow Diagram Level 0 of the Study*

Level 0 data flow diagram of the software is shown in Figure 4, which shows the breakdown of the Smart Eye: Real-Time Traffic Light Color Detection with Audio and Text Assistive Technology for Color Blind Drivers using YOLO Algorithm. This diagram shows how the real-time video data will be processed for object detection, color detection, and producing an audio and text output for the user’s assistance.

The Smart Eye: Real-Time Traffic Light Color Detection with Audio and Text Assistive Technology for Color Blind Drivers using YOLO Algorithm Data Flow Diagram Level 0 contains one external entity and two processes. As shown in the diagram, the software accepts real-time video data from the user's mobile camera and will start to perform processing on the data starting at level 1.0, which is the traffic lights and traffic lights color detection and identification. Then it will undergo the last stage of processing, which is to play an audio, display text, and show box on the detected traffic light output for the user as shown in level 2.0.

With this, the following pseudocode I formulated and followed for the proposed study.

```
# Import necessary libraries and modules:
import cv2
import pyttsx3  # Assuming you're using pyttsx3 for text-to-speech
import threading
cv2: OpenCV library for computer vision tasks.
pyttsx3: A text-to-speech conversion library in Python.
threading: Python module for multithreading, used to run the main

# Function to convert text to speech
def text_to_speech(text):
    engine = pyttsx3.init()
    engine.say(text)
    engine.runAndWait()
function in a separate thread.

text_to_speech: A function that takes a text input and converts it into
speech using the pyttsx3 library.
```

*Script 1. Importing and Text to Speech Function*

```
# Function to detect and identify traffic light color:

def detect_traffic_light(frame):affic light color:

    # YOLO implementation to detect traffic lights
    # Extract the color information from the detected traffic lights
    color = get_traffic_light_color()  # Assume you have a function to get the
color
    if color is not None:
# Display the color on the web interface
        update_web_interface(color)
```

```
# Convert the color to a corresponding text message
        if color == "red":
            message = "Stop!"
        elif color == "green":
            message = "Go!"
        elif color == "yellow":
            message = "Wait!"

        # Use text-to-speech to announce the message
        text_to_speech(message)
```

*Script 2. Traffic Lights Detection Function*

detect_traffic_light: This function is intended to perform the detection and identification of traffic light colors using YOLO (You Only Look Once) or any other method.

get_traffic_light_color(): It is assumed that there is a function to extract the color information from the detected traffic lights. The function checks if a color is detected (if color is not None), then updates the web interface and converts the color to a corresponding text message for text-to-speech.

```
def update_web_interface(color):
    # Update the web interface with the detected color
```

*Script 3. Function to update the web interface with the detected color*.

update_web_interface: This function is a placeholder for updating the web interface with the detected traffic light color. The actual implementation would depend on the web framework or technology being used.

```
def main():
    # Start the video capture using YOLO_Video.py
    video_capture = cv2.VideoCapture("YOLO_Video.py")

    while True:
        # Capture each frame from the video feed
        ret, frame = video_capture.read()

        # Call the function to detect and identify traffic light color
        detect_traffic_light(frame)

        # Check for user input or other events to exit the loop if needed
        # ...

    # Release the video capture object
    video_capture.release()
```

*Script 4. Main function.*

main: The main function sets up video capture using YOLO_Video.py, continuously captures frames, and calls the detect_traffic_light function for each frame. It is intended to run indefinitely until an exit condition is met (e.g., user input).The video capture object is released at the end.

```
if __name__ == "__main__":
    thread = threading.Thread(target=main)
    thread.start()
```

*Script 5. Starting the Main function.*

The code checks if the script is being run as the main module. If yes, it starts the main function in a separate thread using the threading module. This allows the main function to run concurrently with other parts of the program.

## Design Specification

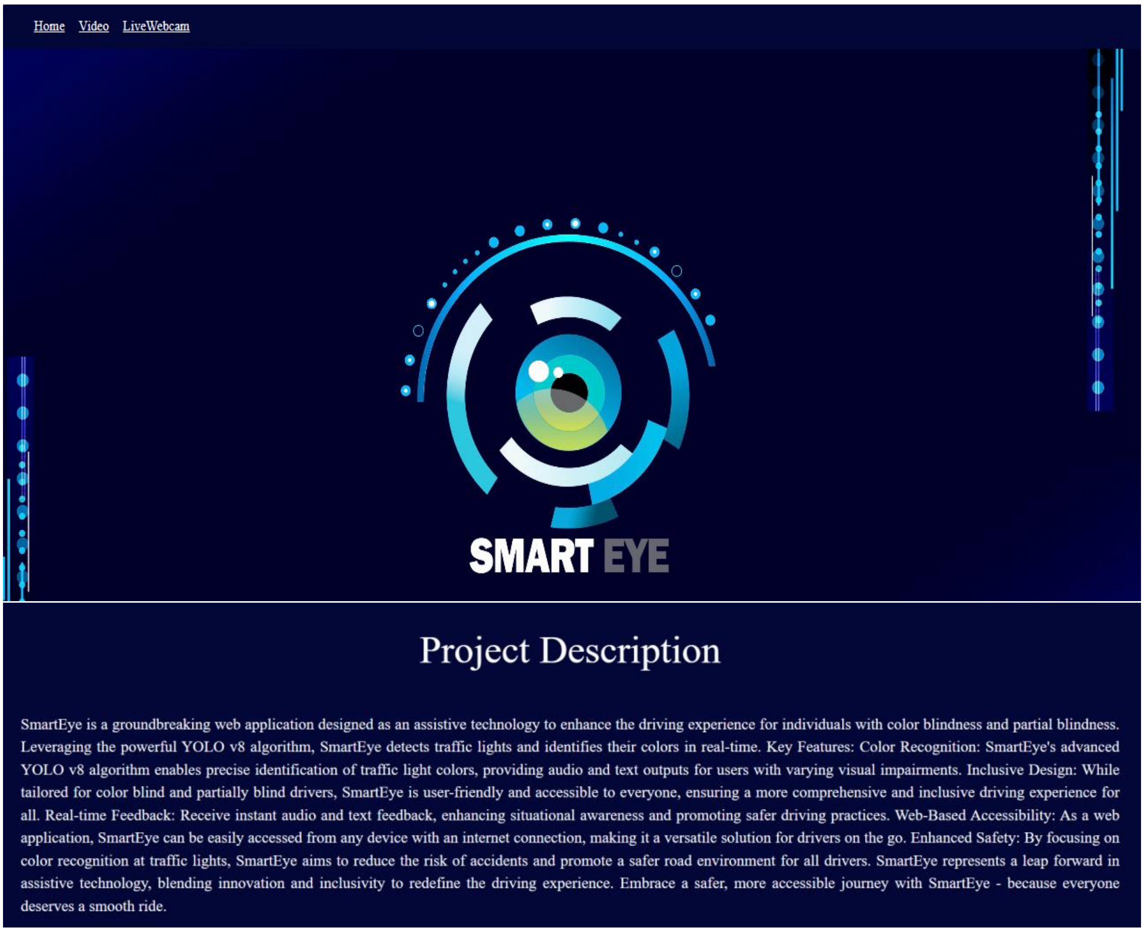


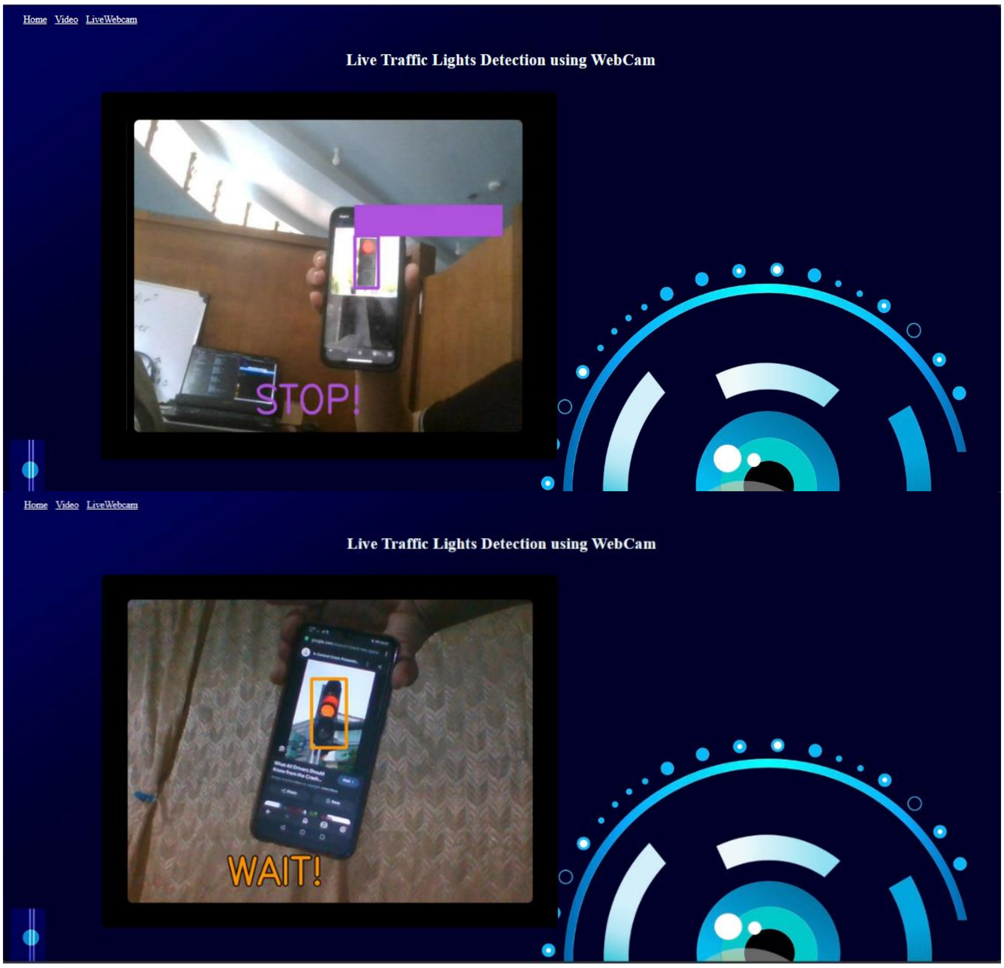

*Figure 5. Systems Homepage User Interface*

The figure above is the user interface of the system for the home page named Smart Eye. It consists of three buttons in the upper left namely: “Home”, “Video”, and “Live Webcam” which have a specific function.

Smart Eye is a groundbreaking web application designed as an assistive technology to enhance the driving experience for individuals with color blindness and partial blindness. Leveraging the powerful YOLO v8 algorithm, Smart Eye detects traffic lights and identifies their colors in real-time.


*Figure 6. Uploaded Test Video User Interface*

The figure above shows the user interface when you click the "Video" button where you can upload your video and test if the detection is working. The system is working the same as what you can see in the figure below.



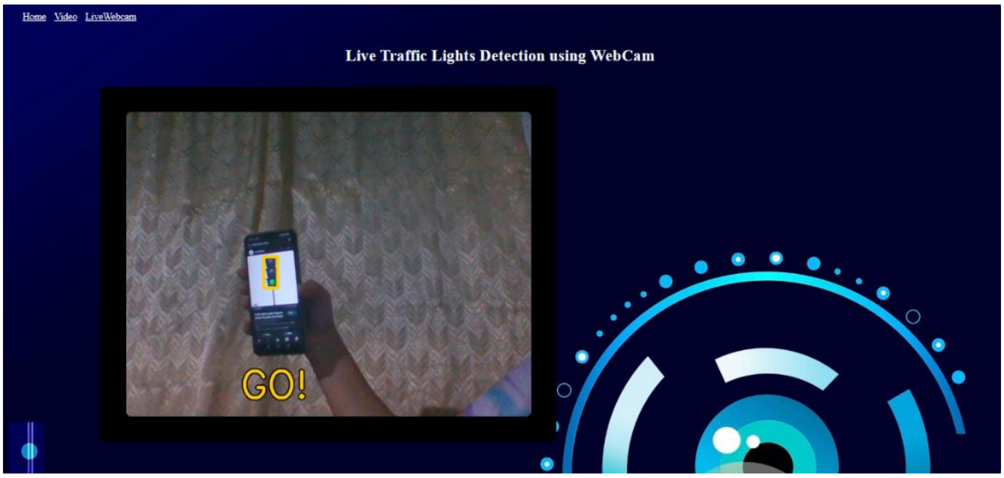



*Figure 7. Real-time Feed via Webcam User Interface*

The figure above shows the user interface when you click the “Live Webcam” button which you can detect in a real-time feed. After the detection, it displays a text based on the color of the traffic lights and it plays an audio informing the status of the traffic lights.

## Testing and Evaluation

The researchers trained the model using the annotated dataset and used the validation set to monitor the model's performance. After training, the researchers tested the model on a separate set of images to evaluate its performance.

The researchers developed a mobile application using the Flutter framework to implement the trained model. The mobile application was designed to work offline and be easily accessible to drivers who may not have access to the internet.

Functionality was evaluated based on whether the mobile application met the user requirements and provided accurate results. Reliability was evaluated based on the mobile application's ability to perform consistently and without errors. Usability was evaluated based on the mobile application's ease of use and user interface design.

Efficiency was evaluated based on the mobile application's performance and speed. Other characteristics evaluated included security, maintainability, and portability.

The application was tested by doing a live demonstration on a car while driving on a highway and a person walking on the street.

**System Evaluation**

The evaluation of the software is based on the evaluation criteria of the standards of ISO 25010. The ISO 25010 software quality model identifies eight main quality characteristics namely: Functionality Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Security, Maintainability, and Portability.

In evaluating the software's performance with the ISO 25010 standards, five (5) IT experts will be selected. The respondents will be selected considering that they have the appropriate qualifications, convenience, and accessibility to the researchers.

Furthermore, the software will also be evaluated by its target end users. Five (5) potential users but persons with difficulty in identifying traffic lights or traffic light color will be preferred to be selected in this study to evaluate the software's functionality, usability, and user experience based on set objectives.

The researchers used a five-point Likert Scale design survey material. The results of the survey will be interpreted using the scale shown in Table 1.

| Scale | Percentage |
|---|---|
| 1.00 - 1.80 | Very Poor |
| 1.81 - 2.61 | Poor |
| 2.62 - 3.42 | Fair |
| 3.43 - 4.23 | Good |
| 4.24 - 5.00 | Excellent |

*Table 1. Scale and Interpretation*

The researchers will get the means to determine the level of performance of the software. It is computed by adding up all the values in the series and dividing them by their count.

**White Box Testing**

White box testing is a technique in which the program logic is examined, and test data is derived. This entails practicing programmers viewing and testing the program to determine whether there is a software vulnerability.

The researchers will test the software on a variety of programmers to guarantee that it adheres to traditional programming methods. Testing and identifying features and components allow developers to consider implementation carefully. This procedure, which is based on the requirements and specification document, will be used to document, and calculate all tests.

### Black Box Testing

Black box testing refers to the process of evaluating software without knowing the internal structure of the code or program. All testing is done from the client's point of view, and the tester is only aware of what the software is supposed to do, but not of how the software processes these requests. While the tester is aware of the software's inputs and expected outcomes, they are unaware of how the software or application processes the input requests and generates the outputs. The tester accepts only valid and invalid input and determines the expected results.

The researchers will conduct a test of the software design which involves people who are having a hard time seeing traffic lights, especially during nighttime. This method of testing should determine whether the software can generate accurate data based on the end user's input. It will also assess how responsive the software's user interface, features, and capability is. In addition, based on the requirements and specification documents, all test cases that must be tested using this method will be calculated.

### User Acceptability Testing

This study will utilize the User Experience Questionnaire (UEQ) [22] to test the “Smart Eye: Real-Time Traffic Light Color Detection with Audio and Text Assistive Technology for Color Blind Drivers using YOLO Algorithm” user acceptability.

The researchers will evaluate the system by surveying questionnaires using black box testing for acceptance and white box testing with the ISO 25010 quality standard. ISO 25010 quality standard is a quality model that is the cornerstone of a product quality evaluation system. The quality model determines which quality characteristics will be taken into account when evaluating the properties of a software product [23]. The User Acceptance Test is a phase of software development in which the software is tested in the real world by its intended audience. User Acceptance Test is often the last phase of the software testing process and is performed before the tested software is released to its intended market. The goal of the User Acceptance Test is to ensure the software can handle real-world tasks and perform up to development specifications [24].

**Validity of the Instrument**

ISO standards are internationally agreed upon by experts [54]. The quality model is the cornerstone of a product quality evaluation system. The quality model determines which quality characteristics will be taken into account when evaluating the properties of a software product.

The quality of a system is the degree to which the system satisfies the stated and implied needs of its various stakeholders and thus provides value. Those stakeholders' needs for functionality, performance, security, maintainability, etc. are precisely what is represented in the quality model, which categorizes the product quality into characteristics and sub-characteristics.

The product quality model defined in ISO/IEC 25010 comprises the eight quality characteristics shown in Figure 5[55].

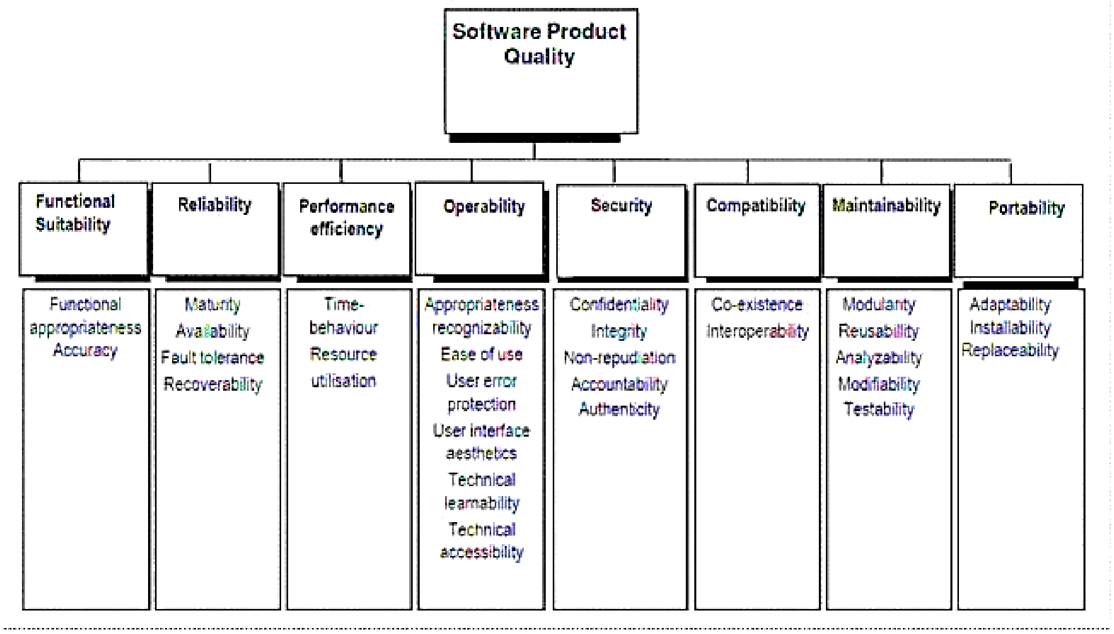



*Figure 8. ISO/IEC 25010 Software Product Quality*

## Respondent of the Study

The researcher gathered data from ten (1s0) selected respondents. The researchers will use purposive random sampling techniques to determine the respondents. The inclusion criteria for respondents for the white box testing are (1) At least 3 years of experience in the field of ICT (2) The respondents must have an Android phone to use the mobile application to test the system.

The inclusion criteria for probable users are (1) Color- or partially-blind drivers; (2) People who have a basic knowledge of computers, or any cross-platform mobile app widely available on Android devices; and (3) The respondents must have an Android phone to use the mobile application to test the system.

## Data Processing and Statistical Tools

The researchers will use the mean and standard deviation to measure the survey and the qualities of the respondents' outputs and to process the data collected from the respondents.

| M | Interpretation |
|---|---|
| 4.50-5.00 | Strongly Agree |
| 3.50-4.49 | Agree |
| 2.50-3.49 | Undecided or Neutral |
| 1.50-2.49 | Disagree |
| 1.00-1.49 | Strongly Disagree |

*Table 2. Mean Interpretation*

The provided ranges typically correspond to the mean (average) scores of the responses. Table 3 shows the interpretation of each range:

*4.50-5.00*

In this range, the respondents have given extremely positive feedback, indicating a high level of agreement or satisfaction with the surveyed system.

*3.50-4.49*

In this range, the respondents have expressed positive sentiments, showing agreement with the surveyed system, although not as strongly as those in the first range.

*2.50-3.49*

In this range, the respondents have provided responses that suggest a level of neutrality or uncertainty. They neither strongly agree nor disagree with the surveyed system.

*1.50-2.49*

In this range, the respondents have given feedback indicating disagreement or dissatisfaction with the surveyed system. However, it's not the strongest disagreement possible.

*1.00-1.49*

In this range, the respondents have expressed extremely negative feedback, indicating a strong disagreement or dissatisfaction with the surveyed system.

## CHAPTER 4

## Results and Discussion

This chapter discusses the implementation of the proposed system that involves the detection of traffic lights with audio and text output, a user-interactive design interface, accuracy validation of the model, and the result of system evaluation by ICT experts and its users. The systems output and overall performance in terms of Functional Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Security, Maintainability, and Portability results are also discussed.

### Development of the System

#### *Capturing a real-time video feed.*

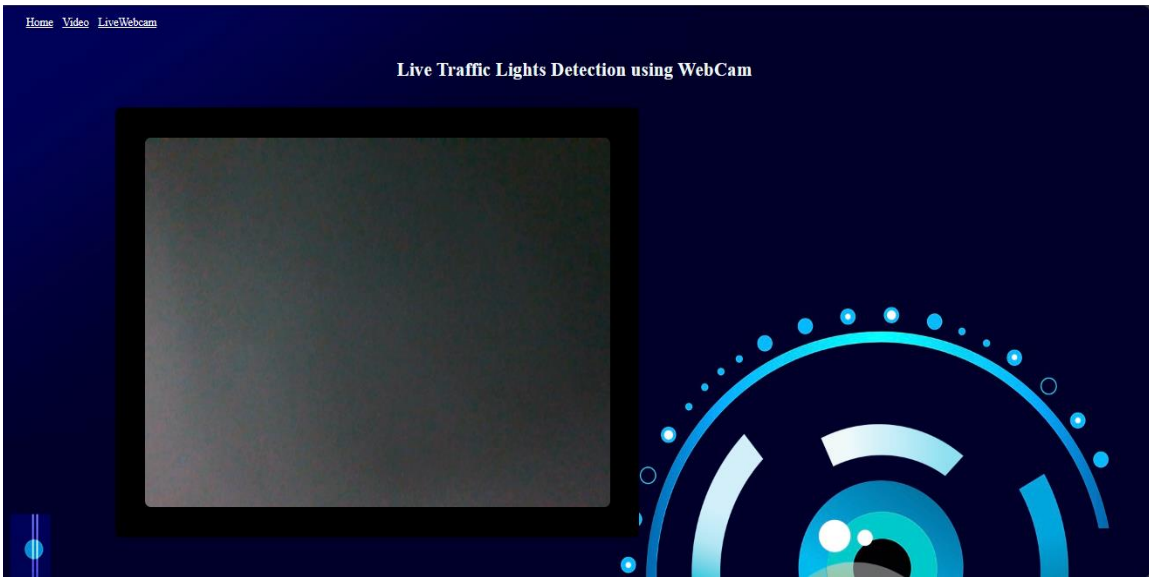



*Figure 9. Web-based Application capturing real-time video using a webcam*

Figure 9 shows a visual representation of the Smart Eye capturing real-time video using a webcam. Upon opening the web application, the user is prompted to click the "Live Webcam" button to capture a real-time video. Upon clicking the button, it will automatically open the webcam attached and then capture a real-time live video from the field of view.

```
cap = cv2.VideoCapture(video_capture)
    frame_width = int(cap.get(3))
    frame_height = int(cap.get(4))
```

*Script 6. Real-Time Video Capture using Computer Webcam*

The program utilizes the built-in webcam of the computer to capture real-time feed. The captured feed was then directed into the trained model for object detection.

***Detecting traffic lights and identifying traffic light colors using the YOLOv8 algorithm.***

The researchers used an existing dataset by Wawan Pradana on traffic lights detection in Roboflow to train the model.

```
!python train.py model=yolov8s.pt data={dataset.location}/data.yaml epochs=80
imgsz=640
```

*Script 7. Training of the YOLOv8 Model*

A pre-trained YOLOv8 model by Ultralytics was used and fine-tuned using the mentioned dataset in the official Ultralytics Google colab notebook for custom training.

```
   model = YOLO("YOLO-Weights/jen.pt")
    classNames = ['green', 'red', 'yellow']
```

*Script 8. YOLOv8 Model for the Traffic Lights and Traffic Light Color Detection*

In this case, the model is designed to detect and classify objects belonging to three classes: 'green', 'red', and 'yellow'. These class names typically correspond to the types of objects you want the model to identify, and they are used for labeling the detected objects during the inference process.

***Providing assistance instruction in audio and text.***

```
for r in results:
        boxes = r.boxes
        for box in boxes:
            x1, y1, x2, y2 = box.xyxy[0]
            x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
            print(x1, y1, x2, y2)
            conf = math.ceil((box.conf[0] * 100)) / 100
            cls = int(box.cls[0])
            class_name = classNames[cls]
            label = f'{class_name}{conf}'
            t_size = cv2.getTextSize(label, 0, fontScale=2, thickness=3)[0]
            print(t_size)
            c2 = x1 + t_size[0], y1 - t_size[1] - 3
            if class_name == 'green':
                color = (0, 204, 255)
                display_text = "GO!"
            elif class_name == "red":
                color = (222, 82, 175)
                display_text = "STOP!"
            elif class_name == "yellow":
                color = (0, 149, 255)
                display_text = "WAIT!"
            else:
                color = (85, 45, 255)
                display_text = ""
```

*Script 9. Displaying Text of the Traffic Light Color*

The detected class determines which display text to use. If the class is "green," the display text is set to "GO!"; when it is "red," it is set to "STOP!"; and when it is "yellow," it is set to "WAIT!" These display text assignments are used later in the code to represent the detected traffic lights on the image. For other classes, the display text is an empty string. The specific visualization method may entail placing the provided text next to bounding boxes that are drawn around the items that are recognized.

```
if current_color == 'green':
                        text_to_speech("GO!")
                    elif current_color == 'red':
                        text_to_speech("STOP!")
                    elif current_color == 'yellow':
                        text_to_speech("WAIT!")
```

*Script 10. Audio Playback of the Traffic Light Color*

The audio playback is played whenever is detects a traffic color. Depending on the value of current_color, a specific message is passed to the text_to_speech function. If current_color is 'green', the message "GO!" is passed to the text_to_speech function. If current_color is 'red', the message "STOP!" is passed to the text_to_speech function. If current_color is 'yellow', the message "WAIT!" is passed to the text_to_speech function.

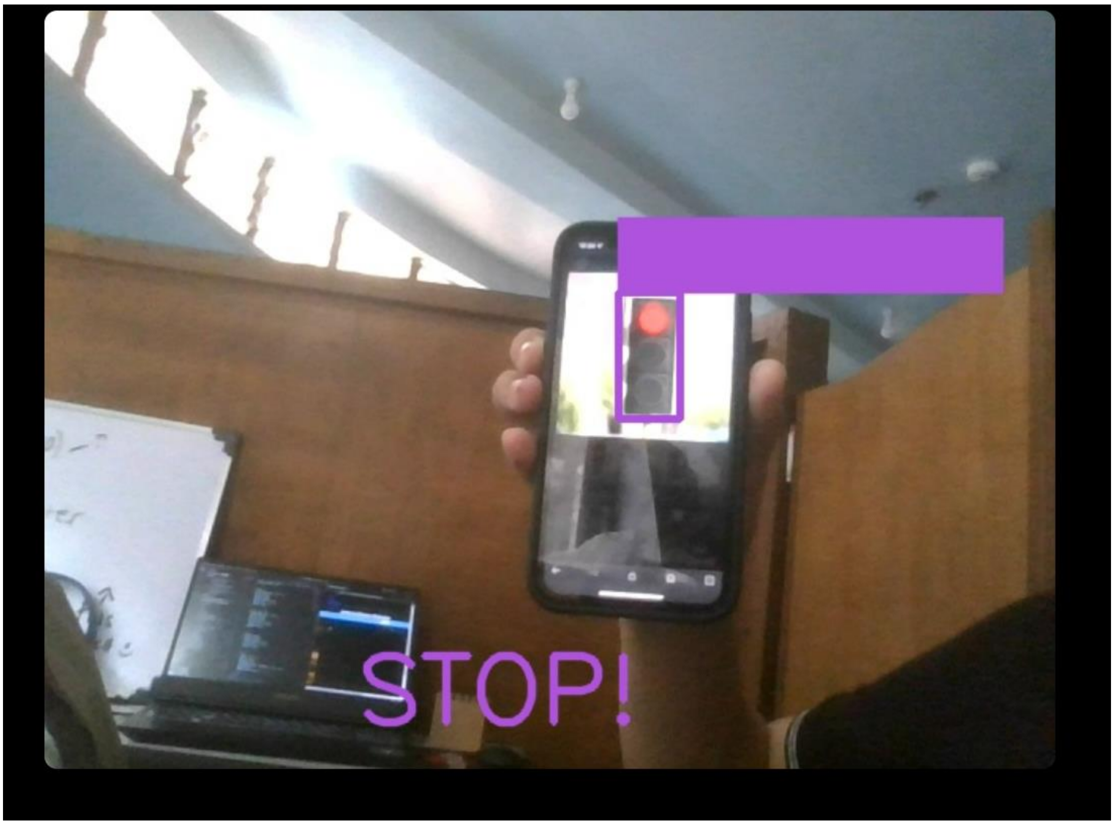



*Figure 10. Example of Traffic Light Detection with Audio and Text Assistance*

Figure 10 above shows the GUI example of the system when detecting the traffic lights and traffic light color. After the detection, it displays a text based on the color of the traffic lights and it plays an audio informing the status of the traffic lights.

**Accuracy Validation**

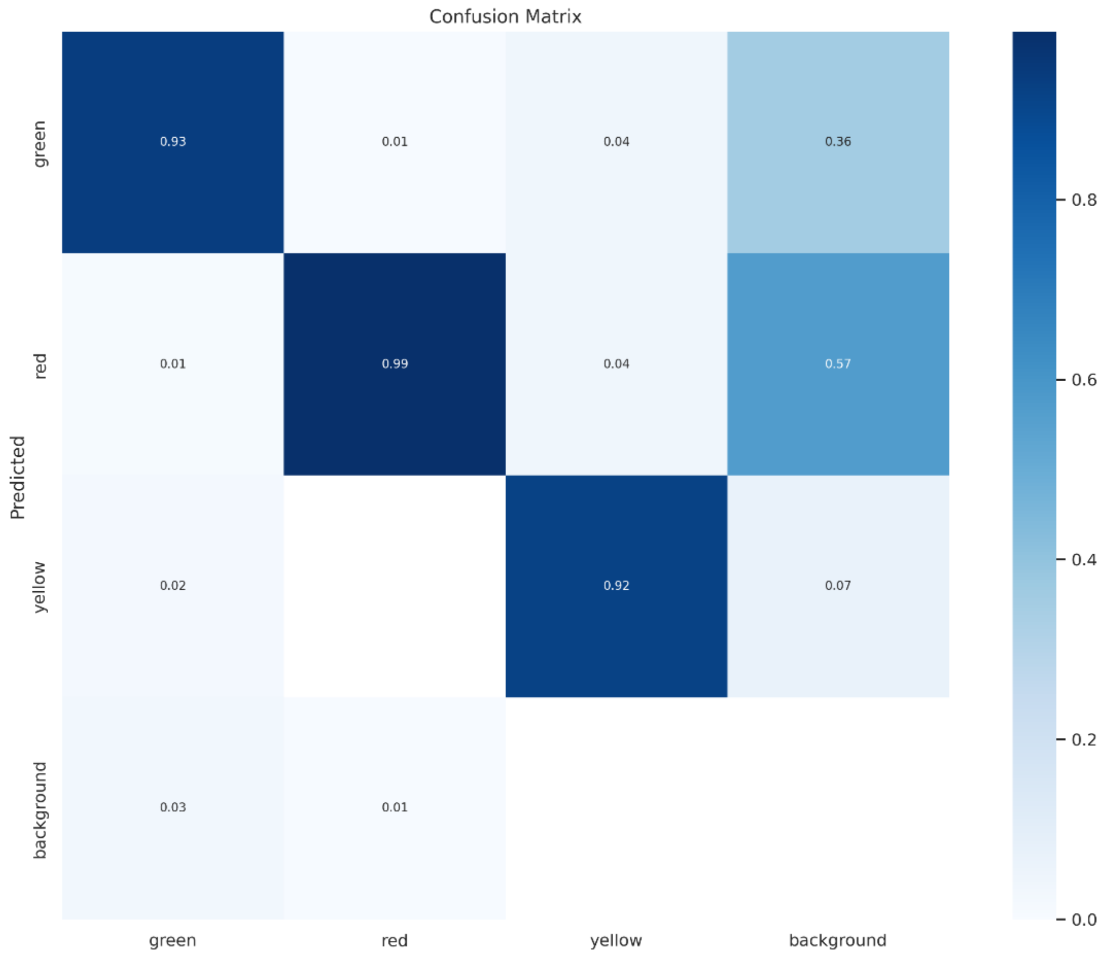



*Figure 11. Confusion Matrix of the YOLOv8 model*

Figure 11 shows the accuracy validation of the model using the confusion matrix of the YOLOv8. The result showed that for every second, the model can detect accurately 93% green, while 0.01% red, 0.04% yellow, and 0.36% background. On the other hand for the red, the model can detect precisely 99% while 0.01% is green, 0.04%

yellow, and 0.57% background. Lastly for yellow, it detects 92% correctly, while the green is 0.02%, red is 0% and the background is 0.07%.

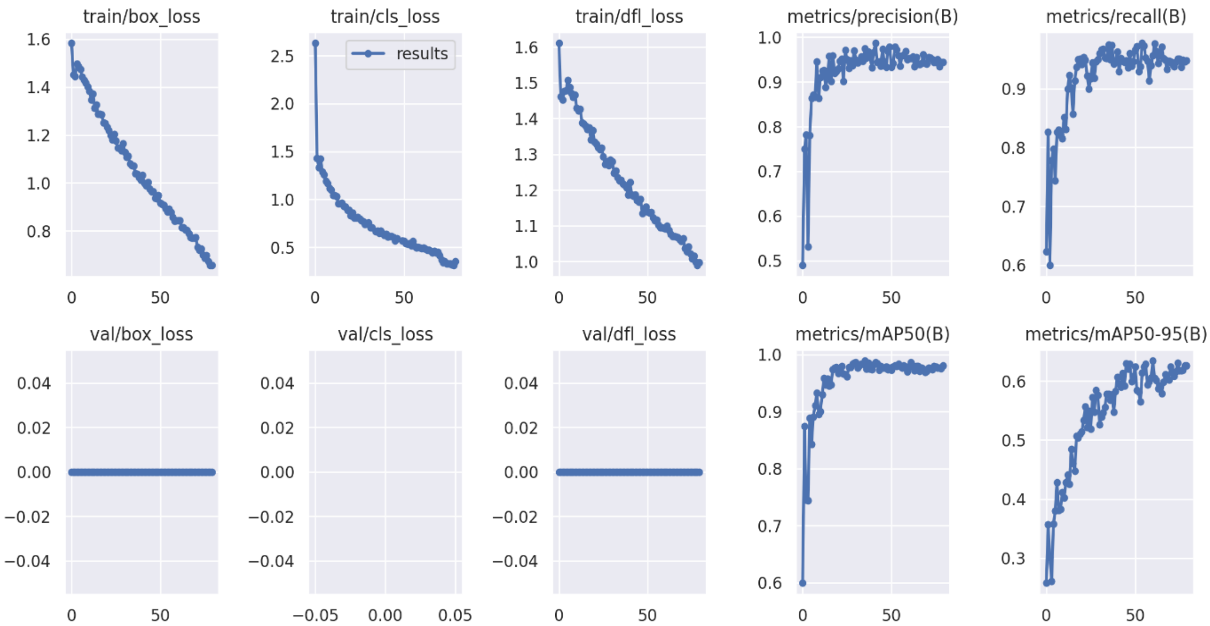



*Figure 12. Training, Testing, and Validation Matrix*

The visual representation of the training, testing and validation matrix of the YOLOv8 model. The results show that there is no overfitting in the model since the graph above is not fluctuating.

**Evaluation of the System Based on the User Acceptability Test Result**

The system was evaluated by five (5) color-blind drivers, the system functionality is based on the research objectives, usability, and user experience.

| Functionality | Mean | Interpretation |
|---|---|---|
| 1. Detect traffic lights and their color using the YOLO algorithm. | 5 | Excellent |
| 2. Display text and play audio of traffic light color status. | 4.6 | Excellent |
| 3. Evaluate the system based on ISO 25010 | 4.6 | Excellent |

| | | |
|---|---|---|
| Mean | 4.73 | Excellent |

*Table 3. Functionality of the System*

*Functionality*. The result shown in Table 3, shows that the overall system functionality was rated “Excellent” by probable users (M=4.73) The first objective was rated as “Excellent” (M=4.8), the second objective was rated as “Excellent” (M=4.8), and the third objective was rated as “Excellent” (M=4.6). The survey shows that the system provides functions that meet the stated and implied needs of its users. Moreover, the result shows that the system complies with all its requirements based on the research stated objectives of the research.

| Usability and User Experience | Mean | Interpretation |
|---|---|---|
| 1. Appropriateness recognizability. (The users recognize the appropriate need for the system) | 4.8 | Excellent |
| 2. Learnability. (The users can use the system with effectiveness, efficiency, freedom from risk, and satisfaction in a specified context of use to achieve specified goals of learning) | 4.2 | Good |
| 3. Operability. (The system is easy to operate and control) | 4.6 | Excellent |
| 4. User error protection. (The system protects the user against making errors) | 4.4 | Excellent |
| 5. User interface aesthetics. (The user interface enables pleasing and satisfying interaction for the user) | 4.8 | Excellent |
| 6. Accessibility. (The system is designed to be used by different types of users) | 4.6 | Excellent |
| Mean | 4.57 | Excellent |

*Table 4. Usability and User Experience of the System*

*Usability and User Experience.* The result shown in Table 4 revealed that the system has an overall rating of "Excellent" in terms of usability and user experience (M=4.57). Moreover, the result shows that the system is easy to learn, use, and operate by the user in a given environment and the system provides easily recognizable logical concepts.

| Criteria | Mean | Interpretation |
| --- | --- | --- |
| Functionality | 4.73 | Excellent |
| Usability and User Experience | 4.57 | Excellent |
| Mean | 4.65 | Excellent |

*Table V. Summary of the User Acceptability Test*

Table 5 shows the summary of the User Acceptability Test which the probable users of the system considered the system "Excellent" (M=4.65). This implies that the system is capable of functioning as expected and can provide a good user experience based on its features.

**Evaluation of the Software Quality Standards Based on ISO 25010 by ICT Experts**

The system was evaluated by ICT Experts using the ISO 25010 software quality standards to test its conformance to international standards in terms of its functional suitability, reliability, performance efficiency, usability, security, compatibility, maintainability, and portability.

| Functional Suitability | Mean | Interpretation |
|---|---|---|
| 1. Functional completeness. (The system's set of functions covers all the specified tasks and user objectives) | 4 | Good |
| 2. Functional correctness. (the system provides the correct results with the needed degree of precision) | 4.4 | Excellent |
| 3. Functional appropriateness. (The system's functions facilitate the accomplishment of specified tasks and objectives) | 4.8 | Excellent |
| Mean | 4.4 | Excellent |

*Table 6. Functional Suitability Evaluation Result by ICT Experts*

*Functional Suitability*. The result shown in Table 6, revealed that the system has "Excellent" functional suitability (M=4.4). The system has functional stability as demonstrated by the "Good" functional completeness (M=4), Functional correctness (M=4.4) and functional appropriateness (M=4.8). The "Excellent" overall functional suitability (M=4.4) of the system means that it conforms to the functional suitability characteristic set by the ISO 25010. This implies that the system's functionality has complied with international quality standards and that the system functions based on the specified objectives.

| Performance efficiency | Mean | Interpretation |
|---|---|---|
| 1. Time behaviour. (The system's response and processing times and throughput meet requirements.) | 4.8 | Excellent |
| 2. Resource utilization. (The amounts and types of resources used by the system meet requirements.) | 4.2 | Good |
| 3. Capacity. (The maximum limits of a product or system parameter meet requirements. | 4.4 | Excellent |
| Mean | 4.46 | Excellent |

*Table 7. Performance Efficiency Evaluation Result by ICT Experts*

*Performance Efficiency.* Table 7 shows that the system has "Excellent" overall performance efficiency (M=4.46). It is expected that the actual performance of the system is dependent on the hardware of the machine on which it is run, thus the system's response time, resource utilization, and capacity exhibit accuracy and efficiency. Furthermore, the result shows that the system has fully conformed to the standard set by ISO in terms of performance efficiency for computer systems.

| Compatibility | Mean | Interpretation |
|---|---|---|
| 1. Co-existence. (The system can perform its required functions efficiently while sharing a common environment and resources with other products, without detrimental impact on any other product) | 5 | Excellent |
| 2. Interoperability. (The system can exchange information and use the information that has been exchanged.) | 4.4 | Excellent |
| Mean | 4.7 | Excellent |

*Table 8. Compatibility Evaluation Result by ICT Experts*

*Compatibility*. Shown in Table 8, indicates that the system has an overall "Excellent" compatibility (M=4.7). The result shows that the system has conformed to the compatibility standard set by ISO for computer systems. Furthermore, the result shows that the system has no known issues while sharing hardware resources and operating system environment with other applications.

| Usability | Mean | Interpretation |
|---|---|---|
| 1. Appropriateness recognizability. (The users recognize the appropriate need of the system) | 4.2 | Good |
| 2. Learnability. (The users can use the system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use to achieve specified goals of learning) | 4.6 | Excellent |
| 3. Operability. (The system is easy to operate and control) | 4.6 | Excellent |
| 4. User error protection. (The system protects users against making errors) | 4.4 | Excellent |
| 5. User interface aesthetics. (The user interface enables pleasing and satisfying interaction for the user) | 3.8 | Good |
| 6. Accessibility. (The system is designed to be used by different types of users) | 4.4 | Excellent |
| Mean | 4.33 | Excellent |

*Table 9. Usability Evaluation Result by ICT Experts*

*Usability*. The result shown in Table 9, revealed that the overall usability of the system is "Excellent" (M=4.33). Therefore, the system has complied with the usability characteristics set by the ISO for computer systems. This means that the system was found useful and easy to navigate by probable users. However, the system's usability depends on several factors such as the quantity of data and hardware specifications.

| Reliability | Mean | Interpretation |
|---|---|---|
| 1. Maturity. (the system is reliable under normal operation) | 4 | Good |
| 2. Availability. (the system is reliable in times it is required to be used) | 3.6 | Good |
| 3. Fault tolerance. (The system operates as intended despite the presence of hardware or software faults) | 3.8 | Good |
| 4. Recoverability. (In the event of an interruption or a failure, the system can recover the data directly affected and re-establish the desired state of the system) | 3.8 | Good |
| Mean | 3.8 | Good |

*Table 10. Reliability Evaluation Result by ICT Experts*

*Reliability*. The result shown in Table 10, revealed that the system has an overall "Good" reliability feature (M=3.8). This is due to the ratings of "Good" in maturity (M=4), availability (M=3.6), fault tolerance (M=3.8), and recoverability (M=3.8) features. Therefore, the system has complied with a good rating with the reliability characteristic requirement set by the ISO for computer systems. However, the system can be improved in its overall performance in terms of Reliability.

| Security | Mean | Interpretation |
|---|---|---|
| 1. Confidentiality. (The system ensures that data are accessible only to those authorized to have access) | 4.8 | Good |
| 2. Integrity. (The system prevents unauthorized access to, or modification of, computer programs or data.) | 4.6 | Excellent |

| | | |
|---|---|---|
| 3. Non-repudiation. (The system records transactions and can be proven to have taken place so that the transactions cannot be repudiated later) | 4 | Good |
| 4. Accountability. (The transactions can be traced uniquely to the entity). | 4 | Good |
| 5. Authenticity. (The identity/function of the resource is the same as it was discussed). | 4 | Good |
| Mean | 4.28 | Good |

*Table 11. Security Evaluation Result by ICT Experts*

*Security.* The result shown in Table 11, indicates that the system has an overall "Good" security feature (M=4.28). The result implies that the system protects information data and is secured as indicated by the authentication process required by its users. The results show and also ensure that user's data are accessible only to those authorized to have access. Furthermore, the system has complied with the security requirement standards set by the ISO for computer systems.

| Maintainability | Mean | Interpretation |
|---|---|---|
| 1. Modularity. (the system is composed of discrete components such that a change to one component has minimal impact on other components) | 4.4 | Excellent |
| 2. Reusability. (A part of a system can be used in more than one system, or in building other systems). | 4.2 | Good |

| | | |
|---|---|---|
| 3. Analysability. (The impact of the intended change to one or more parts of the system can be assessed, diagnosed for deficiencies or failures, or be identified on which parts to be modified.) | 3.8 | Good |
| 4. Modifiability. (The system can be effectively and efficiently modified without introducing defects or degrading existing quality) | 4.2 | Good |
| 5. Testability. (test criteria can be established for the system and tests can be performed to determine whether those criteria have been met) | 4.6 | Excellent |
| Mean | 4.24 | Excellent |

*Table 12. Maintainability Evaluation Result by ICT Experts*

*Maintainability*. Table 12 shows that the overall system maintainability is "Excellent" as indicated by the (M=4.24) overall mean. The results imply that the system can be easily maintained and updated with new systems should the need arise. This also implies that the system requires little to no changes to adapt to new operating system environments. The result also implies that the system components are discreet and such change to one component has a minimal to no impact on the other component. Furthermore, the result shows that the system complies with the maintainability feature standards set by the ISO for computer systems.

| Portability | Mean | Interpretation |
|---|---|---|
| 1. Adaptability. (The system can effectively and efficiently be adapted for different or evolving | 4 | Excellent |

| | | |
|---|---|---|
| hardware, software or other operational or usage environments.) | | |
| 2. Install ability. (The system can be successfully installed and/or uninstalled in a specified environment. | 4.2 | Excellent |
| 3. Replace ability. (The system can replace another specified software product for the same purpose in the same environment) | 4.4 | Excellent |
| Mean | 4.2 | Good |

*Table 13. Portability Evaluation Result by ICT Experts*

*Portability*. The result shown in Table XIII implies that overall, the system has "Good" portability characteristics (M=4.2). The result implies that the system can somewhat easily adapt to changes and can be installed or uninstalled easily. The effectiveness of the system in which its components can be transferred from one hardware to the other or another environment proves that the system has complied with the standard portability characteristics set by the ISO for computer systems.

| ISO 25010 Criteria | Mean | Interpretation |
|---|---|---|
| Functional Suitability | 4.4 | Excellent |
| Performance Efficiency | 4.46 | Excellent |
| Compatibility | 4.7 | Excellent |
| Usability | 4.33 | Excellent |
| Reliability | 3.8 | Good |
| Security | 4.28 | Excellent |

| | | |
|---|---|---|
| Maintainability | 4.24 | Excellent |
| Portability | 4.2 | Excellent |
| Over-all Mean | 4.3 | Excellent |

*Table 14. ISO 25010 Evaluation Summary of Results*

*Summary of Result.* The result, shown in Table 14, an overall mean of 4.3 establishes that the system has excellent quality based on the ISO 25010 International Standards. The result confirms that the system has excellent quality of value that can be in service to its users. Specifically, it has “Excellent” functional suitability (M=4.4), performance efficiency (M=4.46), Compatibility (M=4.7), usability (M=4.33), security (M=4.28), maintainability (M=4.24), “Good” reliability (M=3.8), and portability (M=2).

The result of the study conformed to the goal of ISO 25010 which states that only quality software products must be deployed to its stakeholders, and to ensure the value of these products, a comprehensive specification and evaluation of the quality of software and software-intensive systems must be done. Thus, the “Smart Eye: Real-Time Traffic Light Color Detection with Audio and Text Assistive Technology for Color Blind Drivers using YOLO Algorithms” has been developed to provide quality software for all its stakeholders.

# CHAPTER 5

## Summary, Conclusion, and Recommendations

This chapter presents the summary of findings, the conclusions that were drawn from the findings and the recommendations to further enhance the system for future researchers.

### Summary

This study aimed to develop a web-based application with audio and text output for color-blind drivers using the YOLO algorithm. It captures real-time video and detects traffic lights and traffic light colors to inform the user of the current status of the traffic light.

The system was created using the agile development methodology, which proves extremely advantageous for project advancement. This method employs a systematic process commencing with planning, then progressing through overall design conception, implementation, testing of the model, subsequent model deployment, and finally, the ongoing maintenance of the system.

In this project, an image processing technique was utilized to detect, classify, and recognize traffic lights and traffic light colors, specifically using the YOLOv8 framework.

### Conclusion

The following conclusions were drawn after the conduct of the study.

1. The system fulfills the defined requirements and objectives, encompassing the creation of an algorithm designed to capture and recognize real-time traffic lights and traffic light color.

2. The system meets the objectives by displaying text and executing an audio playback.
3. After conducting tests to evaluate the system's accuracy using User Acceptability Testing and ISO 25010, the image processing technique was identified as highly proficient.

**Recommendation**

Considering the findings of the study and conclusion, the following are the recommendations.

1. It is recommended to deploy this system to a mobile application platform so that it can be tested in a traffic lights system on the roads and to determine the possible changes in the system.
2. Conduct rigorous performance testing on the software across different scenarios and conditions to validate its accuracy, precision, and speed in verifying traffic light colors in real time. This step will help identify any potential weaknesses and allow for necessary adjustments.
3. Conduct comparative evaluations with other state-of-the-art object detection algorithms to assess the performance and advantages of the YOLO algorithm with audio and text output specifically for traffic light detection. This would provide a comprehensive understanding of the strengths and limitations of the proposed system.
4. For future studies, the researchers recommended including traffic lights with two or more colors in a row or traffic lights intended for a four-lane road.

## Bibliography

[1] Types of Colour Blindness. Colour Blind Awareness.‖ [Online]. Available: http://www.colourblindawareness.org/colour-blindness/typesof-colour-blindness/ . [Accessed: 25-Aug-2017].

[2] “Application software | Definition, Examples, & Facts | Britannica,” www.britannica.com. https://www.britannica.com/technology/application-software

. (Accessed: May 4, 2023).

[3] “UPS for DownS,” *www.upsfordowns.org*. https://www.upsfordowns.org/blog/assistive-technology-for-communication-academics-daily-living#:~:text=Assistive%20Technology%20%28AT%29%20is%20any%20device%2C%20software%20or. (Accessed: May 4, 2023).

[4] “Enhancement,” *Techopedia*. https://www.techopedia.com/definition/7039/enhancement#:~:text=An%20enhancement%20is%20any%20product%20change%20or%20upgrade . (Accessed: May 4, 2023).

[5] “Image processing | computer science,” *Encyclopedia Britannica*. https://www.britannica.com/technology/image-processing. (Accessed: May 4, 2023).

[6] [25] Britton, J. (no date) What is ISO 25010?, Perforce Software. Available at: https://www.perforce.com/blog/qac/what-is-iso-

25010#:~:text=ISO%2025010%2C%20titled%20%E2%80%9CSystems%20and,is%20a%20software%20quality%20standard. (Accessed: May 1, 2023).

[7] “Yolo algorithm for object detection explained [+examples],” YOLO Algorithm for Object Detection Explained [+Examples]. [Online]. Available: https://www.v7labs.com/blog/yolo-object-detection. [Accessed: 07-Apr-2023].

[8] N. Kulkarni, “Color Thresholding Method for Image Segmentation of Natural Images,” *Researchgate*, Jan. 04, 2012. https://www.researchgate.net/publication/268348536_Color_Thresholding_Method_for_Image_Segmentation_of_Natural_Images#:~:text=thresholding%20is%20by%20defining%20a,other%20pixels%20to%20the%20background. (accessed Jul. 02, 2023).

[9]I. S. LLC, “ImageMagick,” *ImageMagick*. https://imagemagick.org/script/color-thresholding.php#:~:text=Use%20color%20thresholding%20to%20specify (accessed Jul. 02, 2023).

[10]H. Fleyeh, Road and Traffic Sign Color Detection and Segmentation -A FuzzyApproach. 2015.

[11] M. Afor, T. Forjong Dewingong, B. Aliyu, P. Kumar Mishra, and G. Mishra, “Color Detection System for Diamond Sorting Using Machine Learning,” International Journal of Advances in Engineering and Management (IJAEM), vol. 4, no. 2395–5252, p. 2309, 2022, doi: https://doi.org/10.35629/5252-040523092315.

[12] D. T. Joy, G. Kaur, A. Chugh, and S. B. Bajaj, “Computer Vision for Color Detection,” papers.ssrn.com, 2021.

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3879981 (accessed Apr. 06, 2023).

[13]W. Omar, I. Lee, G. Lee, and K. M. Park, "DETECTION AND LOCALIZATION OF TRAFFIC LIGHTS USING YOLOV3 AND STEREO VISION," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B22020, pp. 1247–1252, Aug. 2020, doi: https://doi.org/10.5194/isprs-archives-xliii-b2-2020-1247-2020.

[14]G. Mu, Z. Xinyu, L. Deyi, Z. Tianlei, and A. Lifeng, "Traffic light detection and recognition for autonomous vehicles," The Journal of China Universities of Posts and Telecommunications, vol. 22, no. 1, pp. 50–56, Feb. 2015, doi: https://doi.org/10.1016/s1005-8885(15)60624-0.

[15]Edward Swarlat Dawam and X. Feng, "Traffic Light Detection and Recognition in Autonomous Vehicles (AVs)," *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech),* 2022. https://www.semanticscholar.org/paper/Traffic-Light-Detection-and-Recognition-in-Vehicles-Dawam-Feng/a5e31e6b83844183ec72e39e6e986e195e93c004 (accessed May 29, 2023).

[16]M. Bagheri, M. Madani, Ramin Sahba, and Amin Sahba, "Real time object detection using a novel adaptive color thresholding method," Dec. 2011, doi: https://doi.org/10.1145/2072652.2072656.

[17]V. Goel, T. Jain, S. Singhal, and S. Kole, "Specific Color Detection in Images using RGB Modelling in MATLAB," Research Gate, Mar. 2017.

https://www.researchgate.net/profile/Silica-Kole/publication/315302830_Specific_Color_Detection_in_Images_using_RGB_Modelling_in_MATLAB/links/5f6e652b299bf1b53ef43f1c/Specific-Color-Detection-in-Images-using-RGB-Modelling-in-MATLAB.pdf (accessed May 01, 2023).

[18] T. Le, S. Tran, S. Mita, and T. Nguyen, "Real Time Traffic Sign Detection Using Color and Shape-Based Features ," Research Gate, 2010. https://www.researchgate.net/profile/Thuc-Nguyen-16/publication/220963964_Real_Time_Traffic_Sign_Detection_Using_Color_and_Shape-Based_Features/links/5759398108aec91374a1639f/Real-Time-Traffic-Sign-Detection-Using-Color-and-Shape-Based-Features.pdf (accessed May 01, 2023).

[19]S. Solak and E. Doğru Bolat, "Real time industrial application of single board computer based color detection system," *IEEE Xplore*, Nov. 01, 2013. https://ieeexplore.ieee.org/abstract/document/6713860 (accessed May 01, 2023).

[20]Z. Shi, Z. Zou, and C. Zhang, "Real-Time Traffic Light Detection With Adaptive Background Suppression Filter," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 690–700, Mar. 2016, doi: https://doi.org/10.1109/tits.2015.2481459.

[21]R. A. A. Raof *et al.*, "Color thresholding method for image segmentation algorithm of Ziehl-Neelsen sputum slide images," *IEEE Xplore,* Nov. 01, 2008. https://ieeexplore.ieee.org/document/4723398

[22] A. S. Gillis, "User Acceptance Testing (UAT)", 2022. [Online]. Available:

https://www.techtarget.com/searchsoftwarequality/definition/user-acceptance-testing-UAT

[23] ISO/IEC 25010, “Software Product Quality”, [Online]. Available:

https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

[24] A. S. Gillis, “User Acceptance Testing (UAT)”, 2022. [Online]. Available:

https://www.techtarget.com/searchsoftwarequality/definition/user-acceptance-testing-UAT

[25] GeeksforGeeks. (2023, May 6). Data preprocessing in data mining. https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/