T7_G14_Leong Yee Chung.pdf

by Leong Yee Chung

Traffic Signs Classification

By

Leong Yee Chung

Lip Zhen Yi

Khew Sei Fong

Steffi Yim Kar Mun

A PROPOSAL

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADAS	Advanced Driver Assistance Systems
TSDR	Traffic Signs Detections and Recognition
HOG	Histogram of Oriented Gradients
LBP	Local Binary Pattern
SIFT	Scale-Invariant Feature Transform
CNN	Convolutional Neural Network
FFT	Fast Fourier Transform
RGB	Red, Green, Blue
HSV	Hue, Saturation, Value
DAS	Driver Assistance Systems
YOLO	You Only Look Once
CLAHE	Contrast Limited Adaptive Histogram Equalization
ROI	Return on Investment

ABSTRACT

The concept of autonomous vehicles was introduced decades ago and the reality of introducing vehicles with efficient self-driving abilities is being realized. While it is a futuristic idea to implement autonomous vehicle which can improve user comfort and aid those who cannot operate motor vehicles due to medical conditions, there are still improvements to be done to the traffic signs detection system, a feature that is very significant in an autonomous vehicle. The exploration of artificial intelligence has given opportunities to accurately detect traffic signs from real-time video feeds. This project aims to develop a system that is efficient in traffic sign detection through shape and color detection to ensure road safety and assist autonomous vehicles.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

CHAPTER 1

Introduction

1.1 Problem Statement and Motivation

Done By: Lip Zhen Yi

Nowadays, autonomous vehicle and the advanced driver assistance systems (ADAS) gradually become the norm. This makes the traffic signs detection system necessary to come out with these autonomous vehicles and ADAS. This is because people will rely too much on autonomous driving technology and ADAS and thus ignore the road situation and traffic signs [1]. In this way, people may have accidents due to their carelessness. The traffic signs are to give some important information more on the rules on the road to maintain the road order and take care of people's safety. However, a traditional way to detect the traffic signs may face some problems when functioning due to different kinds of traffic signs, environmental issue or some other issues. For example, traditional method may be difficult to handle large amount of traffic signs including the differences in shape, color, size and also the design among different places. Then, environment issues might further make the detecting process harder such as bad weather like rain, thunderstorm or fog and poor lightning condition. Accurate detection may also be affected by other issues which are the graffiti or signs that are worn out. Hence, an intelligent traffic signs detection system should be developed.

The motivation of this project is to improve road safety and thus reduce the occurrence of traffic accidents. Besides, this project aims to increase the reliability and usability of traffic sign detection systems. This is because autonomous driving technology and ADAS have become more common and familiar. This traffic signs detection system needs to make sure it detects the traffic signs accurately and effectively.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

1.2 Research Objectives

Done by: Khew Sei Fong

The objective of our research is to develop and implement an advanced traffic sign detection and recognition system to improve autonomous vehicle navigation and reduce the frequency of accidents. The primary objective is to create a traffic sign detection system with a detection accuracy rate of 95% for all standard traffic signs under various lighting conditions and a recognition accuracy of 95% for correctly recognizing the traffic signs. Besides, the system needs to be able to recognize the signs in input images which the signs are in different colors and shapes with high velocity as 30 milliseconds per image.

1.3 Project Scope and Direction

Done by: Khew Sei Fong

The traffic sign detection and recognition system will have two main components: segmentation and recognition. For traffic sign segmentation, the system will detect and extract traffic signs from images using color segmentation techniques for red, blue, and yellow signs, which are common in traffic signage. Shape segmentation will also be applied to accurately isolate the signs based on their geometric shapes, such as circles, triangles, and rectangles. This dual approach of color and shape segmentation will enable the system to effectively identify and isolate traffic signs from various backgrounds and environmental conditions.

For traffic sign recognition, features such as color histograms and shape descriptors will be extracted from the segmented signs and stored in a .csv file. These features will then be used to train machine learning models, which will classify the traffic signs into their respective categories. The performance of these models will be evaluated using accuracy metrics and a confusion matrix, providing a clear understanding of how well the system can recognize different types of traffic signs.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

1.4 Contributions

Done by: Steffi Yim Kar Mun

The project strives to improve current traffic signs detection systems to move autonomous vehicles closer to having a highly accurate and reliable self-driving system. By advancing the state of traffic sign detection in terms of color segmentation and shape detection, this project sets a new benchmark for future innovations that will further improve the efficiency of autonomous vehicles and ensuring road safety.

The project addresses the challenges encountered in previous research and ways to overcome them. It focuses on the improvement of past traffic signs detections and recognition (TSDR) systems and seeks to deliver a traffic sign detection system that surpasses its predecessors in terms of accuracy and reliability.

The implementation of this project brings significant benefits and advantages to consumers, especially for individuals who are unable to physically operate a vehicle. Enhanced accuracy and reliability in detecting traffic signs will contribute to the overall safety and trust towards autonomous vehicles. Thus, consumers can feel more confident in their decisions to purchase and use self-driving cars in the future.

The system also can be used to predict other traffics signs of red, blue and yellow colour of various shapes by using a classifier model that is trained on existing data.

1.5 Background Information

Done by: Leong Yee Chung

Traffic detection is a subfield of computer vision and artificial intelligence that focuses on identifying and interpreting traffic signs from images or videos that are captured by cameras. Advanced Driver Assistance Systems (ADAS) and driverless vehicles are essential to improve road safety by enabling vehicles to be aware of and respond to traffic regulations such as speed limits, turn warnings and intersection

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

experience. The process generally involves two main stages: detection and recognition but this paper mainly focuses on the area study of detection.

These images are then processed using various image processing and computer vision techniques to identify and classify traffic signs. Detected signs can warn the driver and even adjust the vehicle's speed and behavior to comply with traffic laws. Detection methods are categorized into color-based, shape-based, and learning-based methods, including deep learning approaches. On the other hand, to recognize and classify traffic sign involves features extraction and classification, Features extraction consist of Histogram of Oriented Gradients (HOG), Local Binary Pattern (LBP), Scale-Invariant Feature Transform (SIFT), the extracted features are then fed into classifiers such as Random Forests or deep learning models like Convolutional Neural Network (CNNs) and Capsule Networks to identify the specific type of traffic sign.

Traffic signs can be classified into several categories based on their shapes and colors. These categories include warning signs represented by a red triangle, prohibition signs with a red circular shape, mandatory signs displayed as a circular blue structure, reservation signs typically rectangular and blue, and temporary signs often in yellow triangular forms [9]. Traffic sign detection systems face several challenges like lighting variations, occlusion, weather conditions, aging and damage of traffic signs which all these factors can make the system harder to detect and recognize [9]. Achieving processing, high accuracy, and low latency is crucial for immediate use in ADAS and autonomous vehicles. Challenges affecting performance. Although existing methods show high accuracy in identifying characters in images, the transition to video data presents new challenges. These include maintaining accuracy across diverse scenarios, minimizing false alarms, and ensuring real-time processing capabilities [6].

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

CHAPTER 2

Literature Review

2.1 Shape-Based Traffic Sign Detection Methods

Done By: Lip Zhen Yi

Shape-based detection methods are crucial in the field of traffic sign detection within machine learning systems. These methods applied geometric properties of traffic signs such as edges, curves, and some unique shapes to detect and categorise the traffic signs in different environments. In [2], it provided a thorough analysis of different shape-based methods, their performances and their application challenges. It categorized shape-based methods into four primary types which are shape detection technique, shape analysis and matching, fourier transformation and lastly key point detection.

Shape detection technique used to detect the standard size of the traffic signs. It proposed a more effective method called fast radial symmetry which derived from Hough method [3]. A radial symmetry voting mechanism will be generated to detect symmetrical shapes. It is suitable for speed sign detection and more effective than Hough method as it is robust. Figure 2.1.1 shows the radial symmetry voting results which the input image is taken from a vehicle's perspective and showing a traffic sign in the background.

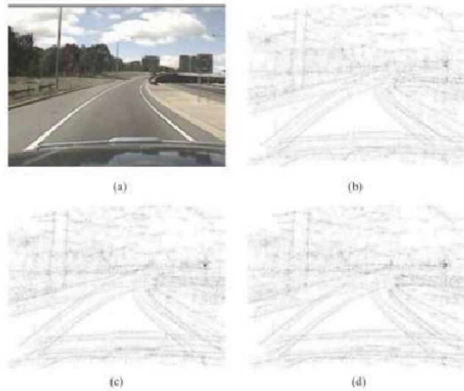

Figure 2.1.1 Radial symmetry voting method [2]

(a) input image with a traffic sign

(b), (c), (d) sample radial symmetry images for the three largest radii

Next, shape analysis and matching stated that complex shape models involved designing models for different traffic sign shapes but these models were noisy and sensitive to shape's movements. Also, a decomposition method used to simplify complicated shapes into simpler components. This technique improved detection by dividing related signs and reducing internal components.

Besides, fourier transformation used to detect the traffic sign shapes based on their shape contours. An enhanced method called Fast Fourier Transform (FFT) was introduced to express various traffic sign shapes and implementing algorithms for localization. Thus, this method was enhancing the efficiency and effectiveness of the traffic sign shapes' representation.

Last but not least, key points detection used to detect the singularities of the traffic signs. In this review, it introduced two types of key points detection which are Harris corner detector and Gabor filter. Harris corner detector aims to identify traffic sign corners and angle points as well as offer a reliable method for locating signs using key point analysis. For Gabor filter, it used to extract local characteristics out of attractive points. Thus, this technique worked well for identifying and grouping characteristics from traffic signs.

For advantages in this method, it is not affected by the color changes by the weather, lightning or the aging of the traffic signs. This proved that shape-based methods are more dependable than color-based methods, especially in an environment with inconsistent lightning [2]. Besides that, shape-based methods are capable of accurately identifying and categorising signs according to their unique geometric features such as triangles and circles. This might help for detecting signs in greyscale photos or in cases when the color information is not accurate.

For disadvantages, shape-based methods need to depend on well-defined and clear edges. The detection accuracy might reduce when the sign edges are blurry because of motion or lower quality photos [2]. Moreover, this method might be difficult to detect the shape of a traffic sign in surroundings with crowded and complicated backgrounds. This can result in wrong detection or missed signs.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

2.2 Colour-Based Traffic Sign Detection Method

Done by: Khew Sei Fong

In the proposed system, colour-based detection method is one of the important elements for traffic sign detection and recognition. This method is used to enable traffic signs can be detected clearly under various environment conditions and lighting conditions. There are different colour spaces used in the colour-based detection method, such as RGB (Red, Green, Blue) colour space, HSV (Hue, Saturation, Value) colour space and HSI (Hue, Saturation, Intensity) colour space. The RGB space is used to binarize the input images and also remove the unwanted pixels and make the image become clearer. While the HSV space is used to resistant to the lighting changes and the HSI space works for detecting the symbol of the sign image.

In this project, the colour segmentation technique is selected and the HSV threshold is used in colour segmentation to resistant to lighting changes and improve the accuracy of detection and recognition. The reason for using HSV colour space is that the colour of the input image can be detected without taking the saturation of colour in account. The HSV threshold is set to identify the desired colour present in a frame. The HSV values must be always adjusted in track bar to determine the optimum values. For this project, the desired colour will be the common colours of traffic signs, such as blue, red, and yellow.[4]

Figure 2.2.1 is the procedures of how HSV space is used in Colour Image Segmentation Technique. Firstly, we need to input an image and select the Region of Interest (ROI) from the input image. Secondly, the RGB to HSV conversion of the input image is required because HSV space is more suitable for using in object detection. The optimum values for Hue, Saturation, and Value are then determined by separating and adjusting the values of the three components. Finally, the background of the image is masked and only the objects of interest are found and saved for further processing. [5]

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

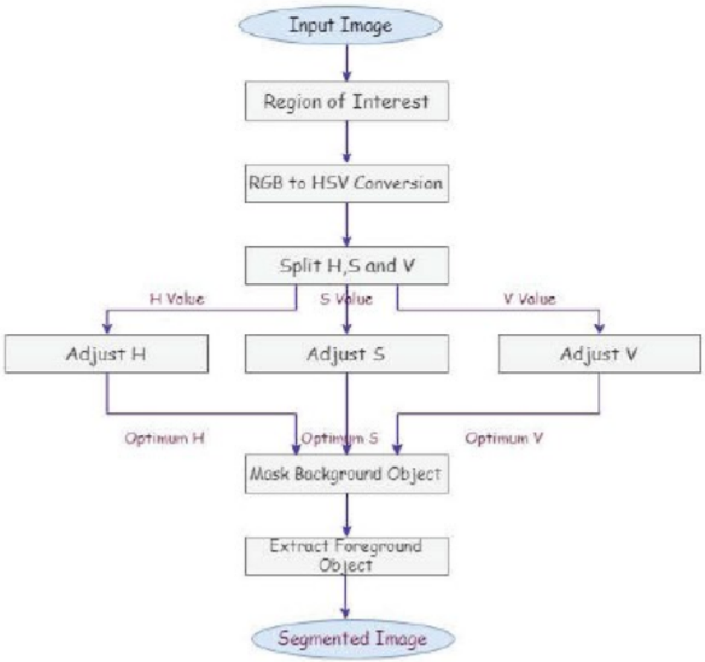

Figure 2.2.1 Colour Segmentation Technique Using HSV [5]

Besides that, Gaussian blur is optional to be used to smoothen and improve the colour feature of the specific frame before the image is processed by HSV threshold for reducing the noise so that accuracy of colour detection can be increased. If the input image has a high resolution, then Gaussian blur can be ignored. The blurring process is performed by kernel window and the blur level is depends on the kernel size.[4] Figure 2.2.2 is the example of 5x5 kernel window of Gaussian Blur.

$$\frac{1}{273}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 7 & 4 & 1 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 7 & 26 & 41 & 26 & 7 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 1 & 4 & 7 & 4 & 1 \\ \hline \end{array}$$

Figure 2.2.2 5x5 Gaussian Blur Kernel Window [4]

The advantages of using color-based detection methods, particularly with the HSV color space, provide significant benefits for traffic sign detection under various environmental and lighting conditions. The HSV color space separates color information from intensity, making it more robust against changes in lighting compared to the RGB color space. This allows for more accurate detection of traffic signs regardless of shadows, brightness, or glare. Additionally, using a color segmentation technique with HSV thresholds helps isolate specific traffic sign colors like red, blue, and yellow, which are common in traffic signs, thereby improving overall detection accuracy. The application of Gaussian blur further enhances the quality of the color features by reducing noise and smoothing the image, which can be particularly beneficial in low-resolution or noisy environments.

Despite its advantages, the HSV color space has some limitations. It requires adjustment of the HSV thresholds for optimal performance, which can be challenging and time-consuming, especially when dealing with diverse environmental conditions. Additionally, Gaussian blur can also lead to the loss of important details in the image, particularly when applied to high-resolution images or when the kernel size is not appropriately chosen. This could potentially result in reduced detection accuracy for small or complex traffic signs.

2.3 Traffic Sign Detection Using CNN

Done by: Steffi Yim Kar Mun

Traffic signs detection and identification is a very crucial feature that is to be implemented in autonomous vehicles because the system must be reliable and accurate enough to identify traffic signs in these self-driving vehicles to ensure driver and road safety. Surveys show there are a few methods used by previous research, which are color-based methods, shape-based methods and hybrid methods [6]. In addition, in recent years, convolutional neural networks (CNN), have been introduced to driver assistance systems (DAS)[7]. Convolutional neural networks of multiple layers are suggested to improve traffic sign recognition through both supervised and unsupervised learning [7], where multi phases of invariant of images can be learnt by the model, with a filter bank layer, a non-linear transform layer and a spatial feature pooling layer exist in each layer. [7] uses CNN, colour information and learnable filters to detect blue circular 'Mandatory' signs with arrows, and white triangular 'Danger' signs with red borders.

Figure 2.3.1 Signs to be Detected by the Algorithm

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

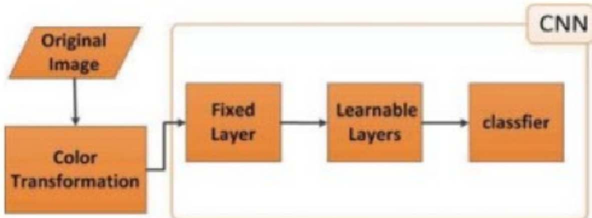

Figure 2.3.2 Flowchart of how the Algorithm Works

The first layer of the CNN in the algorithm convolves the gray scale image from color transformation phase with fixed filters and makes a comparison of the correlation coefficient value with threshold to recognise traffic signs. The learnable filters extract features to be examined by the classifier on which category of traffic signs said sign belongs to [7].

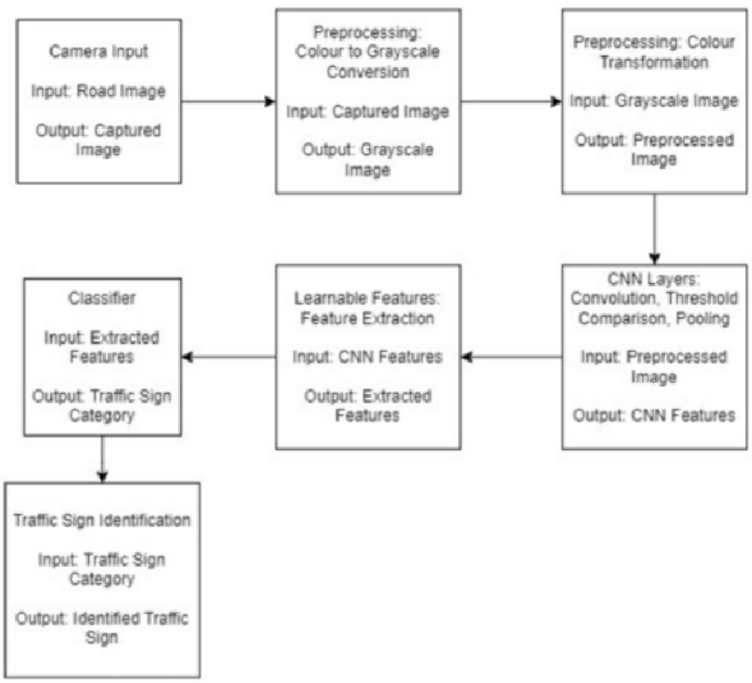

Figure 2.3.3 Block Diagram of Driver Assistance System

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

2.4 Summary of researched papers (Leong Yee Chung)

Various studies have explored different algorithms and methods to improve traffic sign detection and recognition accuracy and efficiency. [11] proposes an improved traffic sign detection and recognition algorithm for intelligent vehicles. This method aims to address common issues with traditional traffic sign detection approaches, such as environmental sensitivity and real-time performance. The proposed approach utilizes HSV segmentation and Gabor kernels to effectively handle lighting changes, achieving high accuracy of 99.75% with an average processing time of 5.4ms per frame. However, the performance heavily relies on the quality and diversity of the training dataset, and the high computational requirements may limit real-time application in low-resource environments. To improve the algorithm's performance, I would suggest integrating shadow detection and data augmentation to better handle extreme conditions, as well as including diverse traffic sign images from various regions and using transfer learning for better adaptability.

In [12], it proposed several key methods to detect and recognize traffic signs effectively such as color-based detection with space conversion to HSV/HSI to distinguishes the colors under varying lighting conditions, shape-based detection with contour detection and polygon approximation to identify geometric shapes of traffic signs, feature extraction with HOG, LBP, SIFT and classification with machine learning model classifiers such as Random Forests or deep learning models like CNNs to recognize and classify the traffic signs accurately. By converting the images to HSV or HSI color spaces, the methods can effectively handle different lighting conditions. Employing classifiers like Random Forests and CNNs leading the method having high recognition accuracy, as these models can effectively distinguish between different traffic sign classes. It has the ability to handle various traffic sign shapes and colors makes it adaptable to different countries' traffic sign standards. However, the combined use of multiple feature extraction techniques (HOG, LBP, SIFT) and complex classifiers like CNNs need a lot of effort for computing, affecting real-time performance. To reduce computing effort, we can use feature selection to find the most important features and reduce the number of features processed. Additionaly, using lightweight neural network models like You Only Look Once (YOLO) can improve real-time performance.

The methods above provided improvements in accuracy and robustness, addressing the computational challenges and dependency on diverse datasets remains crucial. Future research should focus on optimizing computational efficiency and further enhancing the adaptability of traffic sign detection systems to varying environmental conditions.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

2.5 Comparison (Whole Group)

Compared to color-based approaches, shape-based solutions are more resistant to color variations [2]. They frequently pay attention to the signs' size and clarity. However, while key point detection methods need visible points or corners, which may cause unstable in ambiguous settings, edge detection-heavy systems might break down when edges are not apparent.

In [5], HSV values have been split and adjusted in Color Segmentation. For our project we are using HSV space without splitting them but using a range of the color desired. Besides, [4] using Gaussian Blur 3X3 as example to explain function of Gaussian Blur. However in our project, we are using Gaussian Blur 9X9 to blur the input image to get a more accurate result.

In [7], it performs colour segmentation and feature extraction only on two types of traffic signs, which are blue circular 'Mandatory' signs with arrows, and white triangular 'Danger' signs with red borders, whereas this project aims for red, blue and yellow sign colour segmentation and shape detection, which allows for a wider variety of traffic sign types to be detected and more accuracy.

In [11], it uses complex machine learning models which require high computational resources, our method could achieve similar accuracy by using traditional image processing techniques, making it more computationally efficient. By combining color and shape information, ensuring robust detection and segmentation in various conditions, while maintaining computational efficiency.

CHAPTER 3

3.1 System Design Diagram/Equation

3.1.1 Traffic Sign Segmentation

Traffic sign segmentation is a crucial step in traffic sign recognition systems, where the goal is to isolate the traffic signs from the background for further feature extraction and classification. In our system, segmentation is performed using two methods: Color-Based Segmentation and Shape-Based Segmentation. The segmentation process in our traffic sign classification system consists of multiple steps, as depicted in Figure 3.1. The system first performs preprocessing on the image before applying color-based segmentation. Afterward, shape-based segmentation is used to further isolate the traffic signs. If necessary, a second shape segmentation is applied to refine the shape. Finally, a check is made to determine if the segmented image is nearly empty, in which case the system falls back to the second shape-segmented image.

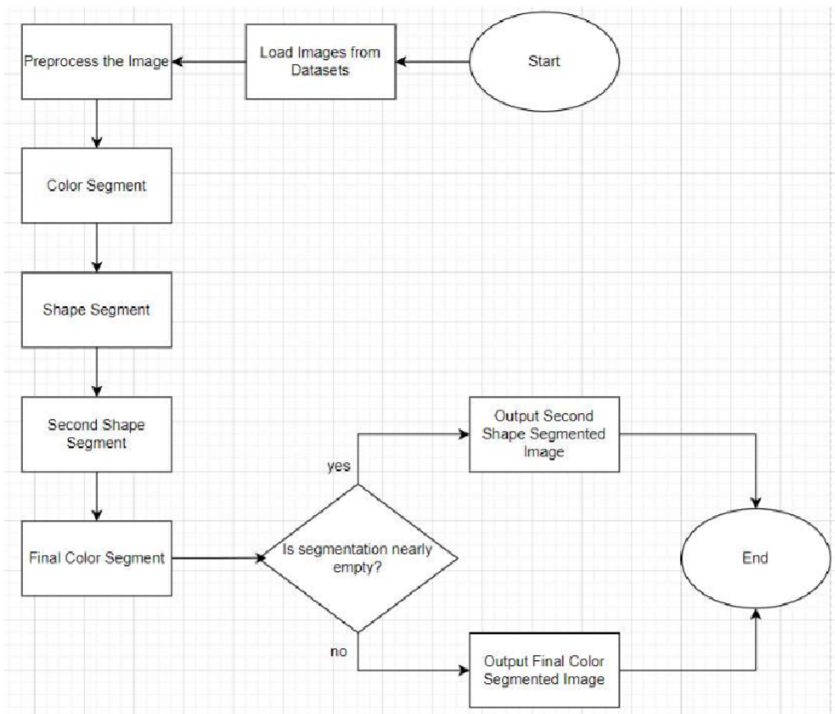

Figure 3.1 Traffic Sign Segmentation Process Flowchart

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

3.1.2 Color-based Segmentation

Done by: Lip Zhen Yi

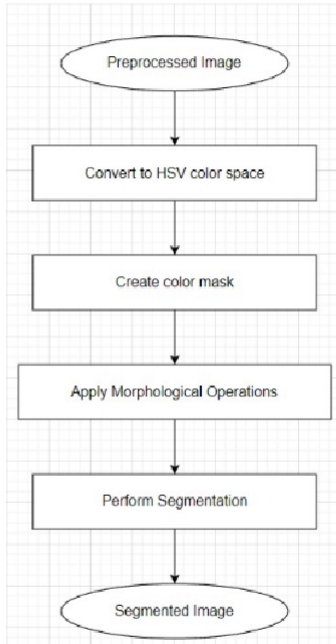

Figure 3.1.2.1 Traffic sign segmentation process flow

Based on Figure 3.1.2.1, the system will first preprocess the traffic sign image to ensure that the quality of image is optimised for further segmentation which are segmentRedSign(), segmentYellowSign() and segmentBlueSign(). Then, the image is converted from the RGB color space which are red, green and blue to the HSV color space (Hue, Saturation, Value). The HSV color space differentiates between chromatic content (hue) and intensity (value) to facilitates the isolation of certain colors which are red, blue and yellow in this system. After that, a color mask is created based on the hue and saturation values from the HSV image. This created mask will selectively suppress some section of the image and highlights the area that correspond to the target color. Then, morphological operations such as opening and closing are applied to the color mask to remove noise and fill gaps. The areas of traffic sign are segmented based on the color mask and separated them from the

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

background. This segmented image can be further processed for shape-based segmentation.

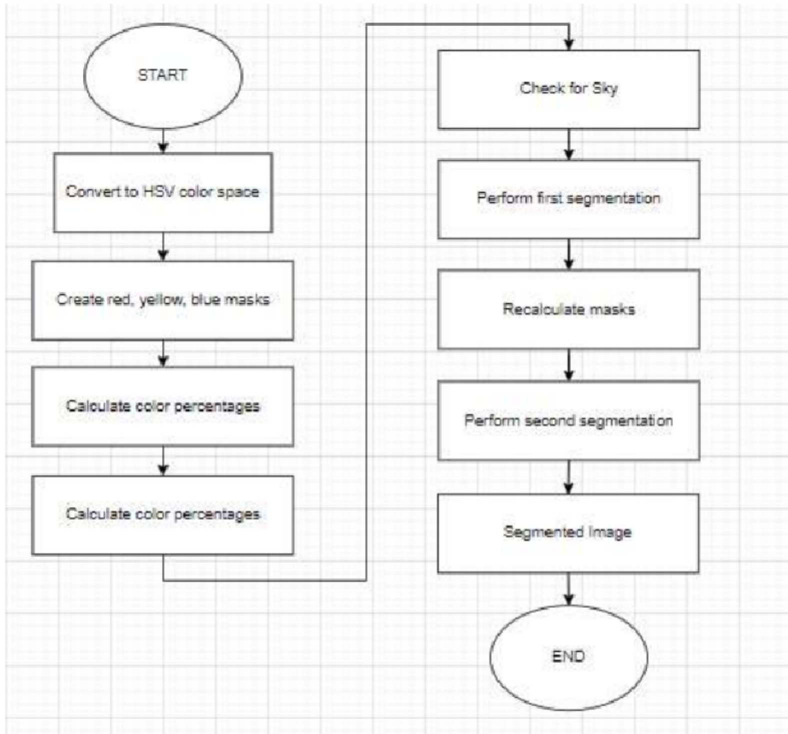

Figure 3.1.2.2 Color-based segmentation process flow

Based on Figure 3.1.2.2, traffic signs' specific colors are used in the color-based segmentation method to separate them out from their surroundings and background. Standard traffic sign colors, such red, yellow and blue can be identified and segmented using the HSV color space. In this system, colorSegment() will be used to determine which color the image should use to segment. First, the images will be converted from RGB color space to HSV color space as HSV color space is able to detect specific colors more efficiently. The equation for converting from RGB values to HSV values is as follow:

$$H = \cos^{-1}\left(\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right)$$

where H is hue representing red, yellow, blue and R,G,B are red, green and blue respectively.

Then, color masks are created for red, yellow and blue colors with certain color range. For red signs, the system applies lower red and upper red ranges to capture different shades of red. For yellow signs, the system will establish a suitable HSV range for yellow and orange tones. While for blue signs, the system will separate them from the background of the image. These color masks will apply morphological operations such as closing and opening to remove the noise. The masks for red, yellow and blue color are defined as:

$$Mask = \begin{cases} 1 & if\ Lower_{color} \leq HSV \leq Upper_{color} \\ 0 & otherwise \end{cases}$$

After that, the system will calculate the percentage of pixels that match to each color masks which are red, yellow and blue to identify the dominant color of the image. The percentage of the image covered by the color can be calculated by:

$$Color\ Percentage = \frac{number\ of\ pixels\ in\ mask}{total\ number\ of\ pixels\ in\ the\ image} \times 100$$

The color which detected as the highest percentage will be selected as the dominant color. If blue color is the dominant color, the system will apply a special sky detection logic to make sure the blue color in the image representing a traffic sign and not the sky. This is because the presence of big blue regions in the upper part of the image is frequently used to identify the sky. The sky detection can be defined as:

$$Sky\ Present = \begin{cases} True & if\ Blue\ color\ Percentage > P_{threshold} \\ False & otherwise \end{cases}$$

where $P_{threshold}$ is a predefined threshold used to access whether the image has enough blue coverage to qualify as the sky.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

Then, the system will detect the contours in the color mask and segment the traffic sign based on the detected contour from the background. The final segmented image will be used as the input of the shape-based segmentation.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

3.1.3 Shape-based Segmentation

While color is a helpful starting point for traffic sign identification, traffic signs often share similar colors but vary in shape (e.g. circles, triangles, squares, and octagons). Thus, to achieve better accuracy, the system incorporates shape-based segmentation, which focuses on extracting the shape information from the traffic signs after the initial color segmentation.

The shape-based segmentation process involves several steps as outlined below:

1. Grayscale Conversion: The first step is converting the color-segmented image into a grayscale image to simplify the shape detection process. This step helps in removing any color information while retaining the structure of the image for further analysis.

2. Adaptive Thresholding: The grayscale image undergoes adaptive thresholding, which dynamically adjusts the threshold values based on local neighborhoods, resulting in a binary image where the traffic sign is distinguished from the background. This binary image is then inverted to highlight the object of interest (the traffic sign) for further contour detection.

3. Contour Detection: The next step is detecting contours within the thresholded image. Contours represent the boundaries of objects, which in this case correspond to the shapes of the traffic signs. These contours are analyzed based on their geometric properties such as perimeter and area.

4. Shape Classification Using Area Similarity: For shape classification, the system compares the area of the detected contours with known geometric shapes (e.g., circle, triangle, square, octagon). Each detected contour is evaluated against the expected area of these shapes based on its perimeter. If the area similarity is within a predefined threshold, the system labels the shape as either a triangle, square, octagon, or circle.

5. Fallback to Approximation and Convex Hull: In cases where no exact match is found using area similarity, the system applies shape approximation. Contours are simplified into polygons, and the convex hull technique is used to smooth out or refine the shape's boundaries. This approach helps in handling noisy or

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

broken contours, which may occur due to imperfections in the image or noise during the segmentation process.

- Circularity Check Formula (for Circle Detection):
  This formula computes the circurlarity to check if the detected shape is close to a circle.

$$Circlelarity = \frac{4\pi \times A}{P^2}$$

  A: area

  P: perimeter

- Triangle Area Formula:
  Classify different triangle based on area similarity.

$$A_{triangle} = \frac{\sqrt{3}}{4} s^2$$

  Where s $= \frac{P}{3}$, P as the perimeter

- Square Formula:

$$A_{square} = s^2$$

  Where s $= \frac{P}{4}$, P as the perimeter

- Octagon:

$$A_{octagon} = 2(1 + \sqrt{2})s^2$$

  Where s $= \frac{P}{8}$, P as the perimeter

6. Final Shape Segmentation: Based on the shape classification or approximation process, the system generates a shape mask, which is then applied to the original image. The final segmented shape is extracted from the original image, ready for further processing such as feature extraction or classification.

The figure 3.1.3 shows the steps involved in the shape-based segmentation process. By following this approach, the system achieves a robust shape segmentation process that effectively handles different types of traffic signs, even in challenging scenarios where contours may be noisy or broken.

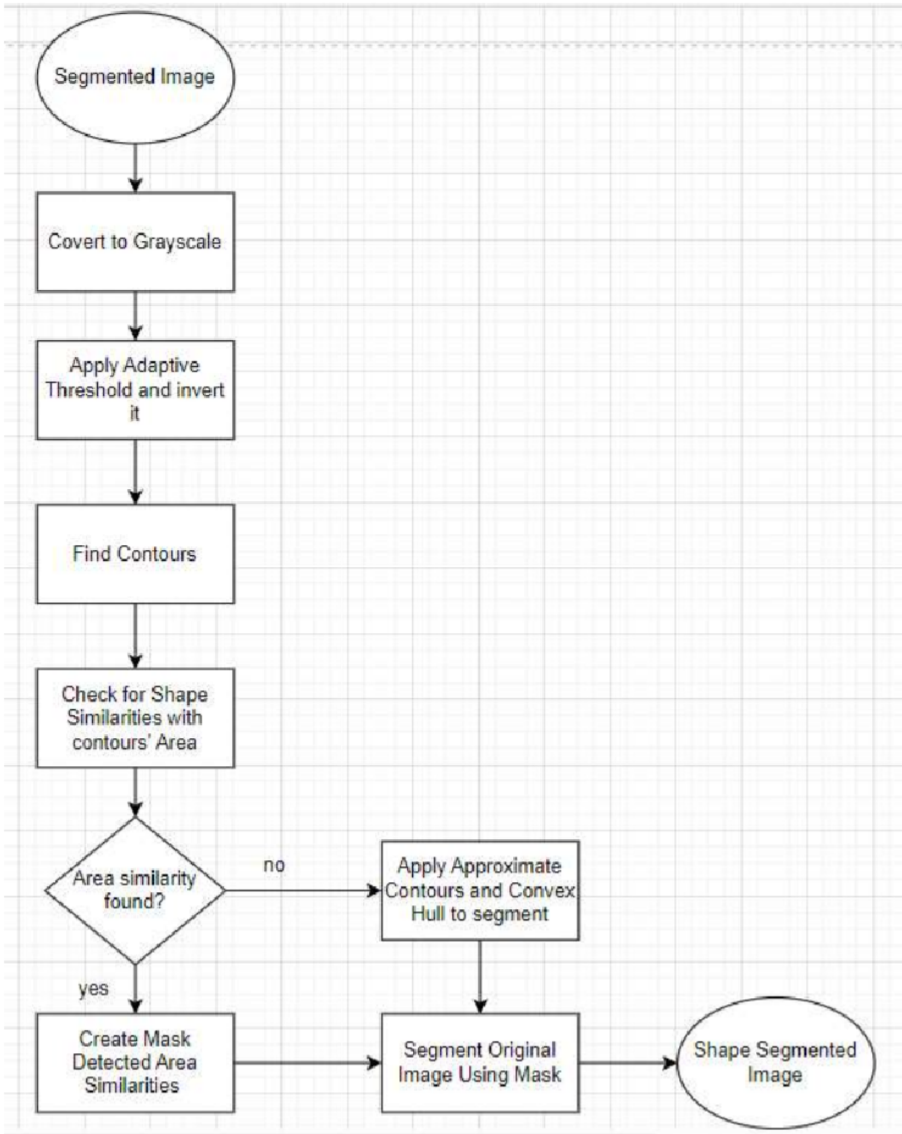

Figure 3.1.3 Shape-Based segmentation process flow

3.2 Feature Extraction

Done by: Steffi Yim Kar Mun

Two methods of feature extraction have been used to extract the features of traffic sign images, namely Colour Histogram for the extraction of colour data, and Hu Moments for the extraction of shape data.

The colour histograms were generated using 256 bins for each channel, resulting in three separate histograms representing the blue, green, and red channels. This approach captures the distribution of pixel intensities across these channels, providing detailed colour information. Each histogram reflects the amount of colour present in the image, allowing for more effective colour feature extraction. The distance between two colour histograms, I and Q, can be computed as below, where a_{ij} indicates the likeliness between the colours corresponding to bins i and j:

$$D_{his}^2(I,Q) = \sum_i^N \sum_j^N a_{ij}(I_i - Q_i)(I_j - Q_j)$$

Hu moments extraction is to capture shape feature data. It consists of a set of seven invariant descriptors, designed to remain constant under translation, scale, and rotation, making them ideal for recognizing and distinguishing shapes. This method ensures that even when images appear in different orientations or sizes, their geometric properties can still be effectively identified.

$$\boldsymbol{normalized\ value} = \frac{\boldsymbol{value - minVal}}{\boldsymbol{maxVal - minVal}}$$

3.3 Classifications

Done by: Khew Sei Fong

In the classification part, we have implemented 3 machine learning models which are Random Forest, Support Vector Machine (SVM) and k-nearest neighbour (KNN).

Random Forest is a powerful ensemble learning method that enhances the accuracy of prediction by combining the multiple decision trees. It operates by constructing numerous decision trees, each trained on a different subset of training data and then aggregating their results. This can reduce the variance of model which in turn minimizes the risk of overfitting which is a common issue with single decision tree. Overfitting always occurs when a model captures noise in training data and lead to poor generalization on unseen data. By averaging the predictions of numerous decision trees, Random Forest creates a more stable and accurate model that generalizes well to new data.

SVM is a supervised learning algorithm used for classification and regression tasks. It works by finding the best boundary that separates data points of different classes. SVM is effective with high-dimensional data and can handle cases where data is not linearly separable by using technique called kernel trick. One of SVM's strength is its focus on the most challenging data points which helps reduce overfitting and improve generalization to new data.

KNN is a simple and effective algorithm used for classification and regression. It works by finding the closest data points to the one being predicted and then using their values or classes to make decision. The number of neighbours, 'K' is an important parameter. Small value of 'K' can make the model sensitive to noise while large 'K' value may cause the missing of important patterns. KNN is well-known with its simplicity and effectiveness in situations where the relationship between features and labels is complex.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

CHAPTER 4

System Design

4.1 System Block Diagram

Done by: Khew Sei Fong

In Figure 4.1, the system is designed to detect the traffic signs in input image. First, the image will be input into the traffic sign detection and recognition system and the input image will be pre-processed using resizing, Gaussian blur and CLAHE to prepare the image for further processing. After that, the color segmentation will be carried out on the pre-processed image to segment the traffic sign based on the specific color ranges. After color segmentation, the segmented image will be further processed by shape segmentation to identify specific shapes of traffic sign. Then, the second shape segmentation will be carried out to improve the segmentation accuracy. The final color segmentation will be performed on the output from the second shape segmentation to ensure that the traffic sign is identified and segmented correctly. If the output of the final color segmentation is nearly empty, the output of second shape segmentation will be selected for further processing. Otherwise, the output of final color segmentation will be selected.

For feature extraction, the relevant features such as color histograms and shape descriptors will be extracted from the selected segmented image. The extracted features will be stored into .csv file for further processing.

The extracted features will be loaded as dataset for classification. The dataset will be split into training and testing datasets. 80% of the dataset is used for training machine learning models and 20% is reserved for testing. The models such as Random Forest, SVM and KNN is trained using the training data. After training, the models are saved for future use. The saved models are then tested on the testing dataset to evaluate their performances. The results of classification will be output.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

CHAPTER 4

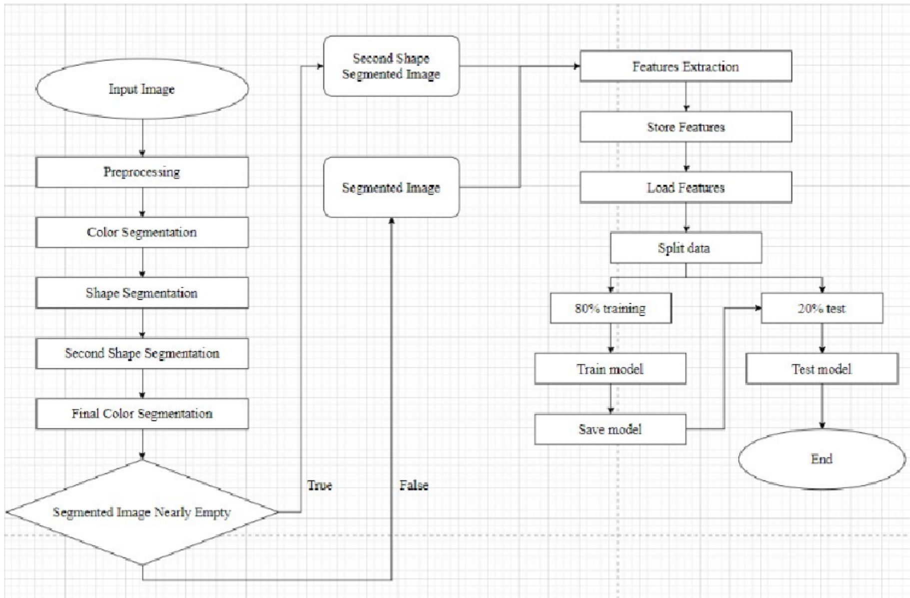

Figure 4.1 Block diagram of overall system

4.2 System Components Interaction Operations

Done By: Khew Sei Fong

There are some system components in this project and there is operation for each component. They will also interact with other component for further process.

First, the Input Image component accepts raw traffic signs images as input to system and passes the images to Preprocessing component. In Preprocessing component, image processing techniques such as resizing, Gaussian blur and CLAHE are applied to ensure the images can be detected easier. After that, the processed images are sent to Color Segmentation component for detecting and segmentation on dominant colors like red, blue and yellow and it passes the segmented image to Shape Segmentation component. In Shape Segmentation component, the shapes in the image such as circle, triangles or square are identified and segmented. After that, the segmented shapes are sent to Second Shape Segmentation component for refining the initial shape segmentation. The refined shape segmentation result is sent to Final Color Segmentation for ensuring the images are segmented correctly. If the final segmented image is nearly empty, the result from that Second Shape Segmentation component is selected. Otherwise, the result from the Final Color Segmentation component is chosen and sent to Feature Extraction Component.

In Feature Extraction component, the relevant features such as color and shape will be extracted from the segmented image. The extracted features will be sent to Store Features component. In Store Features component, the extracted features will be saved and stored for later use in classification.

In classification, there are also a few components. First, the extracted features will be loaded as dataset and dataset is shuffled. The dataset will be sent to Split Data component and the dataset will be split into 2 parts. 80% of the dataset will be used for training and 20% will be used for testing. The training dataset will be sent and used in Train Model component for training machine learning model. After training, the model is saved and it will be tested in Test Model component to evaluate the performance of the trained model. The evaluation will be shown to users and end the system flow.

4.2.1 Segmentation

(Done by: Leong Yee Chung)

The segmentation process in our system isolates traffic signs from the surrounding environment to facilitate accurate traffic sign detection. The process includes two main methods: color-based segmentation and shape-based segmentation. Each method works together to ensure that the traffic sign is accurately segmented from the image background.

In the initial step, color segmentation is applied to detect and isolate traffic signs based on their distinct colors (red, yellow, and blue). The system uses predefined HSV color ranges to create binary masks for each color. These masks isolate the regions in the image that contain the traffic signs.

The system has implemented three types of color segmentation: red, yellow, and blue color segmentation. An algorithm detects which type of color segmentation should be applied based on the dominant colors present in the image. To ensure the system segments out the image correctly, multiple segmentations are performed to narrow down the region of interest.

To improve segmentation accuracy, our system performs multiple rounds of segmentation. The first round involves color segmentation, which helps narrow down the area of interest but may not fully separate the traffic sign from the background. For instance, if parts of the background have colors similar to the traffic sign, color segmentation alone might fail to achieve a clean separation.

To address this, the system follows color segmentation with shape-based segmentation, which focuses on identifying the geometric properties of the traffic sign. This approach helps refine the segmented area by using contour detection and shape approximation. The process involves:

1. Initial Shape Segmentation: The first shape segmentation analyzes the contours of the segmented area. If a contour's area matches a predefined shape (e.g., circle, triangle, square, or octagon), it is used to isolate the traffic sign.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

2. Fallback for Inaccurate Area Similarity: If no clear area similarity is detected between the contours and expected shapes, the system falls back to using approximate contours and applies the convex hull technique. This ensures that the traffic sign is segmented smoothly, eliminating for irregularities or noise.

3. Second Shape Segmentation: A second round of shape segmentation is performed to capture any remaining parts of the traffic sign. This stage further refines the segmented area, ensuring that the contours are as close to the actual traffic sign's shape as possible.

Each round of shape segmentation progressively enlarges the segmented shape, as the system uses adaptive thresholding and bitwise_not as a mask to detect and refine the traffic sign's boundaries.

After the second round of shape segmentation, the system applies a final color-based segmentation to tighten the segmentation around the traffic sign. However, if this final segmentation results in a nearly empty image due to the wrong color segmentation being used, the system uses a fallback mechanism. It reverts to using the second shape-segmented image, which provides a more accurate representation of the traffic sign's shape.

By using multiple rounds of segmentation with adaptive thresholding, bitwise inversion, and the convex hull method, the system progressively refines the traffic sign's boundaries. This ensures that the segmented shape is accurate, even in challenging conditions where the traffic sign may be partially obscured or affected by noise.

4.2.1.1 Color Segmentation

Done By: Khew Sei Fong

In color segmentation, the input image is converted from BGR format to HSV color space which is better suited for color-based segmentation. HSV color ranges are defined for red, blue and yellow, with red and yellow each having two ranges while blue has one. These ranges are used to create masks for each color with the red and yellow masks being combined respectively. The Morphological operations are applied on the blue mask to clean it up.

As shown as Figure 4.2.1.1, there is a function to calculate the percentage of image occupied by each color using the masks. To address the issue of blue signs being mistaken for the sky, the function includes additional checks for blue saturation and distribution in the image.

```
Red Percentage: 0%
Yellow Percentage: 6.7326%
Blue Percentage: 0%
Dominant color: Yellow
After first segmentation - Red Percentage: 0%
After first segmentation - Yellow Percentage: 6.68344%
After first segmentation - Blue Percentage: 0%
```

Figure 4.2.1.1 Percentage of image occupied by each color

Based on the dominant color, the function performs an initial segmentation of the image using different functions (segmentRedSign(), segmentYellowSign(), segmentBlueSign()) for each color. After initial segmentation, the color percentages are recalculated on the segmented image. If a new dominant color emerges, a second segmentation is performed.

Finally, the function decides whether to use the result of the initial or secondary segmentation based on the refined color analysis, ensuring an accurate representation of the segmented traffic signs.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

4.2.1.2 Shape Segmentation

(Done by: Leong Yee Chung)

In shape segmentation, the input image is converted to grayscale image. Then, the system applies adaptive thresholding to the grayscale image to create a binary image where the foreground shapes are highlighted. Adaptive thresholding dynamically adjusts based on local image properties, making it robust to varying lighting conditions. This binary image is then inverted using bitwise_not, so the traffic sign appears in white and the background is black, and this is why the shape segmentation progressively enlarges the segmented shape as shown in figure 4.2.1.2.1. Contours are detected from the binary image. These contours represent the boundaries of objects in the image, corresponding to the traffic signs. The contours are found using the findContours() function with RETR_LIST to list out all contours detected.

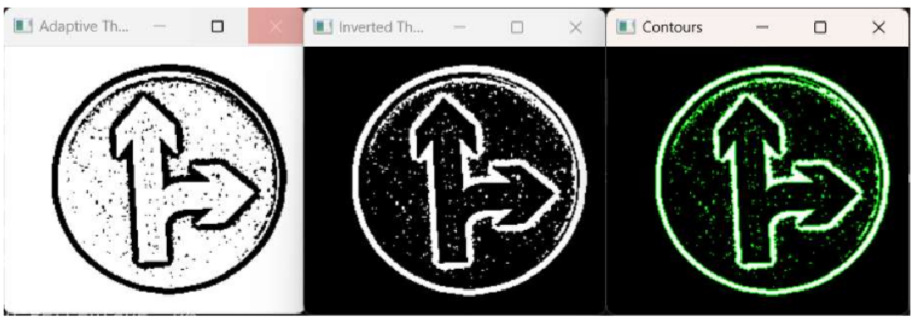

Figure 4.2.1.2.1 Adaptive Threshold Inverted Binary Image with Contours

After that, there is a function to create a mask, shapeMask to store the segmented shapes. It iterates through each detected contour and calculates its area. If the area exceeds a threshold (e.g., 1000 pixels), it is considered a valid shape, further checks are performed to classify the shape based on its geometric properties. We use geometric properties like circularity, side lengths, and aspect ratio to identify shapes such as circles, triangles, squares and octagons.

For circularity, the system checks if a shape is circular using the circularity formula, which compares the contour's area and perimeter. A circularity value greater than 0.85 is used to classify a shape as a circle. A perfect circle has a circularity value of 1. Real-world images are often noisy, so by setting a threshold of 0.85, we allow for minor distortions while still accurately classifying a shape as a circle. This tolerance gives the algorithm flexibility to handle slight variations in the shape.

For triangles, if the contour's perimeter divided by three gives a side length, the system calculates the area of the triangle based on this side length and compares it with the detected contour's area. A tolerance of 20% is applied to check for area similarity. Triangles in real-world images can be distorted due to various factors like camera angles and shadows. A 20% tolerance allows for these distortions, accommodating slight variations in side lengths and angles while still recognizing the shape as a triangle. This tolerance prevents the system from misclassifying the shape due to small deviations from the perfect equilateral shape.

For squares or rectangles, the aspect ratio of the bounding box is used to differentiate squares from rectangles. A square is identified if the aspect ratio is close to 1 (between 0.95 and 1.05), and the area is checked with a 5% tolerance. The 5% tolerance for aspect ratio allows for slight scaling or perspective distortion when detecting real-world traffic signs, ensuring that nearly square shapes are still correctly classified. Similarly, the 5% tolerance for area allows for slight deviations from perfect geometric conditions.

For octagons, the side length is derived from the perimeter, and the area is calculated using the formula mentioned in Chapter 3.1.3.

If the contour matches a known shape based on area similarity, it is drawn on the shapeMask and labelled on the output image. If no shape is found based on area similarity, there is a function falls back to using the approximate contours by applying the convex hull to smooth or fix the broken contours. These contours are then drawn on the shapeMask. Finally, the function uses the shapeMask to segment the corresponding shapes from the original image and returns the segmented result where only the detected shapes.

The workflow in Figure 4.2.1.2.2 visually demonstrates the system's decision-making process.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

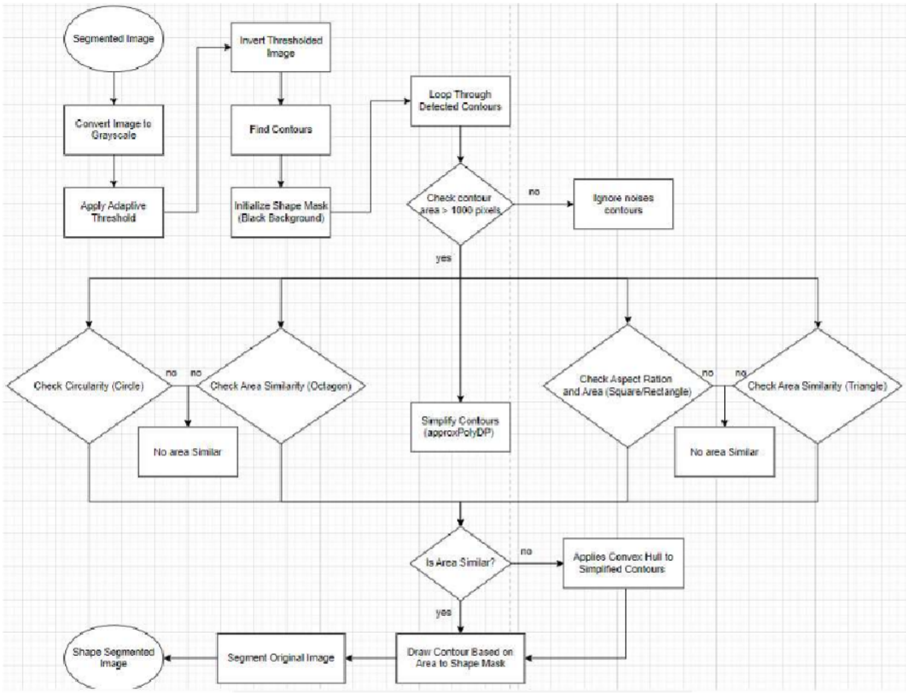

Figure 4.2.1.2.2 Shape Segmentation Process Workflow

4.2.2 Feature Extraction

4.2.2.1 Hu Moments

(Done by: Leong Yee Chung)

Hu Moments are a set of seven invariant descriptors used for shape recognition, based on the moments of an image. These descriptors are particularly useful because they remain invariant to transformations such as translation, scaling, and rotation, making them robust for shape-based object detection in traffic sign recognition systems. In our system, we utilized Hu Moments to describe the shape characteristics of segmented traffic signs. The process of extracting Hu Moments involves multiple steps, as outlined in Figure 4.2.2.1

The process begins by converting the segmented traffic sign image to a grayscale format. Since the background in the image is often black, a mask is applied to exclude this background, ensuring that the computation of moments focuses solely on the sign. The mask is created by thresholding the image to detect non-black pixels and then inverting it to highlight the traffic sign.

Once the mask is applied to the grayscale image, the system computes the spatial moments, which are then used to calculate the seven Hu Moments. These moments encapsulate the geometric properties of the shape, providing a robust way to describe traffic signs. To enhance the utility of the Hu Moments across different images, the values are normalized using Min-Max normalization. To ensure consistency and comparability of the Hu Moments across different images, we implemented Min-Max normalization. This method scales the values of the Hu Moments to fall within a range of 0 to 1, making the features more robust when dealing with different traffic signs of varying shapes and sizes. The normalization process adjusts the raw Hu Moment values based on the minimum and maximum values in the data, ensuring that even the smallest and largest feature values are represented within the same scale.

After the Hu Moments are computed and normalized, the system assigns a label to each traffic sign based on its color: **0** for blue signs, **1** for red signs, and **2** for yellow signs. This labelling allows the data to be easily processed and classified in subsequent machine learning tasks. Both the Hu Moments and the assigned labels are saved to a CSV file, creating a dataset of features and corresponding labels that can be used to train and evaluate machine learning models.

Figure 4.2.2.1.1 illustrates the complete workflow for extracting and storing Hu Moments from segmented traffic signs, starting with image preprocessing and concluding with the generation of a labelled dataset.

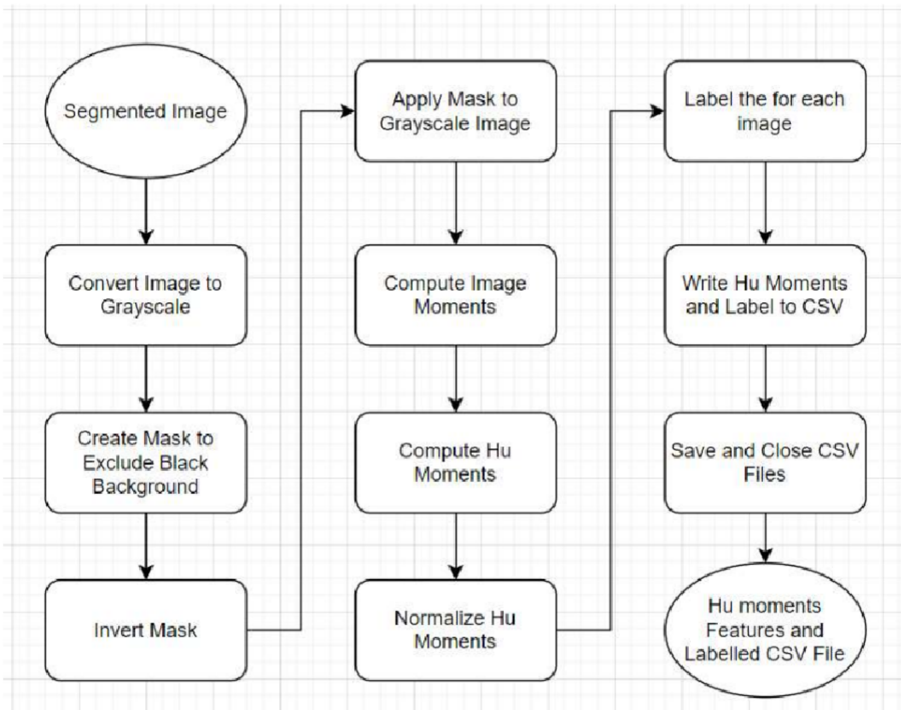

Figure 4.2.2.1 Hu Moments Feature Extraction Process Flowchart

4.2.2.2 Colour Histogram

(Done by: Steffi Yim Kar Mun)

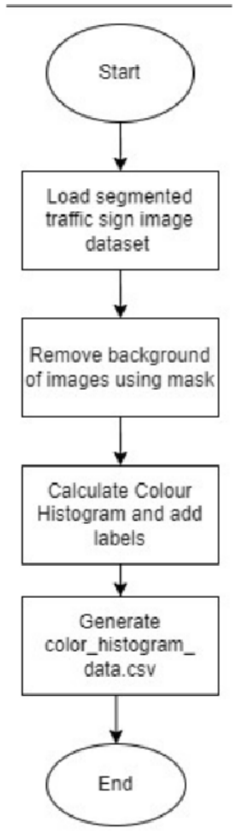

Figure 4.2.2.2 Colour Histogram Feature Extraction Process Flowchart

Based on Figure 4.2.2.2, feature extraction is started off by loading the dataset of traffic signs in the system. The traffic sign images in this dataset are already processed using shape segmentation and colour segmentation.

Next, the background of the images will be removed, because even though the segmented images now only consist of the traffic signs and black background, the black background can affect the colour data result extracted from these images using Colour Histogram.

Then the system will conduct colour feature extraction using the Colour Histogram technique which extracts the colour data of each image and displays them using blue, green and red channels which consist of 256 bins respectively. Labels will

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

be added to each image, 0 for informative signs, 1 for prohibitive signs and 2 for warning signs. The system then saves all the features extracted of each image in a csv file called ‘color_histogram_data.csv’.

4.2.3 Classification

Done By: Khew Sei Fong

Based on Figure 4.2.3, there are 3 models implemented in this project which are Random Forest, SVM and KNN. First, user can select one of them for further process. Due to there are 2 features extraction methods have been used which are color histogram and hu moments, user can also select one of them or select the combined features for classification later.

In classification, the features extracted from feature extraction will be loaded as dataset. For selecting combined features, the two extracted features will be loaded and combined as one dataset. The dataset will be shuffled using a fixed random seed which 123 to avoid inaccurate prediction caused by the data being in sequence. After that, the dataset is split into two parts which are training data and testing data. 80% of the dataset will be used for training and rest 20% will be used for testing later.

Hyperparameter tuning will be carried out for finding the best parameter of selected model. The best parameter will be used with the training data in training stage to ensure the model achieving the highest prediction accuracy.

The trained model will be tested by predict the testing data. The performance of model will be evaluated in accuracy, precision, recall and also F1-score. The result of evaluation is displayed to the users.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

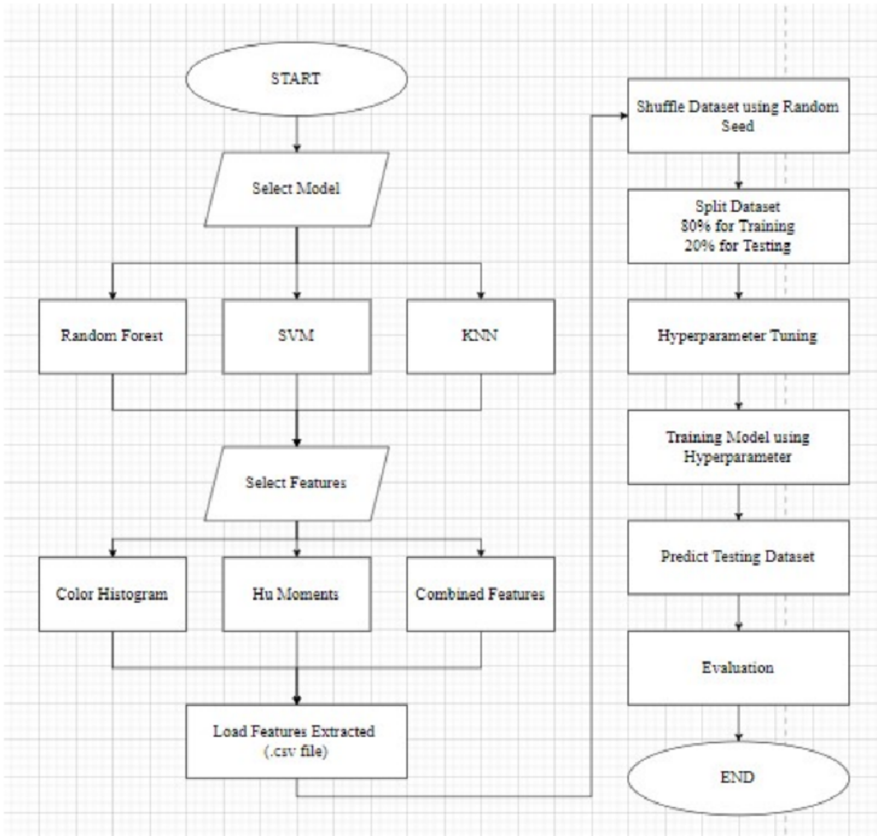

Figure 4.2.3 Process Flow of Classification

CHAPTER 5

System Implementation

5.1 Hardware Setup

Done by: Done by All Members

The hardware involved in this project is laptop devices. The laptops used to write the code for traffic sign shape segmentation, colour segmentation, feature extraction, and

Description	Specifications
Model	Victus by HP Laptop 16-e1044AX
Processor	AMD Ryzen 5 6600H with Radeon Graphics
Operating System	Windows 11
Graphic	NVIDIA® GeForce RTX™ 3050 Laptop GPU (4 GB GDDR6 dedicated)
Memory	16.0 GB
Storage	512 GB PCIe® NVMe™ TLC M.2 SSD

classification.

Table 5.1.1 Specifications of laptop

Description	Specifications
Model	Acer Nitro 5
Processor	AMD Ryzen 7 5800H
Operating System	Windows 11
Graphic	NVIDIA GeForce GTX 1650
Memory	8GB DDR4 RAM
Storage	512GB M.2 NVMe PCIe 3.0 SSD

Table 5.1.2 Specifications of laptop

Description	Specifications
Model	Dell Latitude 7420
Processor	11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz 1.80 GHz
Operating System	Windows 10
Graphic	Intel(R) Iris(R) Xe Graphics
Memory	16.0 GB
Storage	512 GB

Table 5.1.3 Specifications of laptop

Description	Specifications
Model	Lenovo IdeaPad 310-14ISK 80SL
Processor	Intel Core i5-6200U
Operating System	Windows 10
Graphic	NVIDIA GeForce 920MX
Memory	12GB DDR4 RAM
Storage	512 GB SSD

Table 5.1.4 Specifications of laptop

5.2 Software

Done by: Leong Yee Chung

The software used to perform traffic sign detect and segment is:

- Microsoft Visual Studio Enterprise 2022 (64-bit) Version
    - Version 17.5.5
- C++ Programming language
- OpenCV library
    - Version 4.9.

5.3 System Setting and Configuration

Done By: Khew Sei Fong

The project utilizes the OpenCV library, version 4.9.0 which provides a wide range of functionalities essential for this project, including machine learning, feature extraction, segmentation and preprocessing capabilities. The programming language used is C++, especially version C++20.

The dataset used in this system consists of 84 traffic sign images, with each color category (red, blue, yellow) containing 28 signs. The dataset is split such that 67 images are used for training and the remaining 17 images are reserved for testing. This splitting process is performed during the classification phase.

For classification, three machine learning models are implemented in this system which are KNN, SVM and Random Forest. Additionally, three types of features are used for training the models which are Color Histogram, Hu Moments and Combined Features. For parameter used for each model, the KNN model is configured with K=3. The SVM model is configured with a regularization parameter C of 1.0 and a gamma value of 0.1. The Random Forest model is configured with a maximum depth (MaxDepth) of 10, a minimum sample count (MinSampleCount) of 10 and a maximum number of categories (MaxCategories) of 2.

5.4 System Operation

5.4.1 Segmentation

Done by: all members

In the segmentation part of this system, users can choose to have the system display the original traffic sign image and the segmented traffic sign image side by side for comparison, and this can be done by selecting for the segmentation of specific color traffic signs or all the signs found in the dataset, as shown in Figure 5.4.1.1.

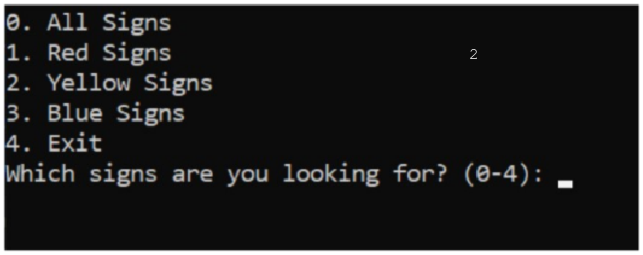

Figure 5.4.1.1 Selection Menu for Segmentation

After making their selection, for example, red signs, the system will display the output as shown in Figure 5.4.1.2, Figure 5.4.1.3 and Figure 5.4.1.4.

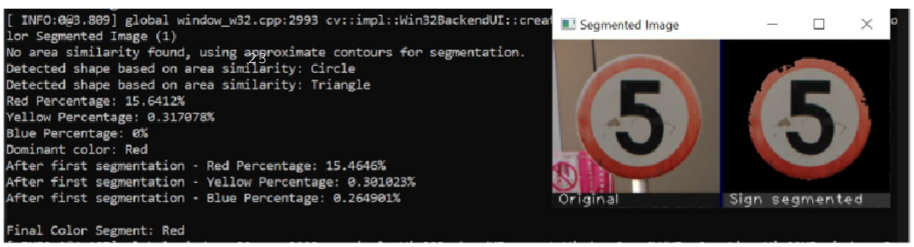

Figure 5.4.1.2 Output for Red Sign

Figure 5.4.1.3 Output for Yellow Sign

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

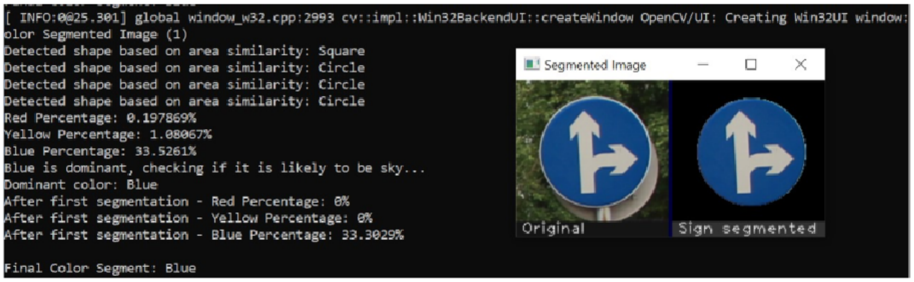

Figure 5.4.1.4 Output for Blue Sign

5.4.2 Extraction

Done by: Leong Yee Chung & Steffi Yim Kar Mun

Users can choose to either conduct feature extraction which includes both Hu Moments and Color Histogram techniques or exit the program.

```
Menu Options:
1. Feature Extraction (Hu Moments & Color Histograms)
2. Exit
Enter your choice: _
```

Figure 5.4.2.1 Conducting Feature Extraction

Once the extraction processes have finished, an output as shown in Figure 5.4.2.2 will be shown.

```
Feature extraction complete. Data saved to hu_moments_data.csv and color_histogram_data.csv with labels.
Menu Options:
1. Feature Extraction (Hu Moments & Color Histograms)
2. Exit
Enter your choice: 2
Exiting program.
```

Figure 5.4.2.2 Successful Feature Extraction and Exiting Program

The feature extraction data will all be saved in their respective csv files, namely 'color_histogram_data.csv' and 'hu_moments_data.csv'. Snippets of both csv files are shown in Figure 5.4.2.3 and Figure 5.4.2.4.

1	FileName	Label	ColorHist_	ColorHist_	ColorHist_	ColorHist_	ColorHist_	ColorHist_	ColorHist_	ColorHist_	ColorHist_	ColorHist_	ColorHist_	ColorHist_
59	Segmentec	2	0	0	0	0	0	0	0.007519	0.015038	0.022556	0.045113	0.075188	0.052632
60	Segmentec	2	0.532632	1	0.567368	0.276842	0.182105	0.155789	0.153684	0.137895	0.157895	0.128421	0.132632	0.110526
61	Segmentec	2	0	0	0.018829	0.066946	0.098326	0.123431	0.108787	0.09205	0.150628	0.127615	0.125523	0.117155
62	Segmentec	2	0	0	0	0.012579	0.006289	0.142558	0.54717	0.767296	0.899371	1	0.922432	0.731656
63	Segmentec	2	0.010152	0.030457	0.167513	0.456853	0.583756	0.959391	1	0.951777	0.540609	0.449239	0.274112	0.312183
64	Segmentec	2	0	0	0.002525	0.002525	0	0.007576	0	0.015152	0.015152	0.015152	0.020202	0.027778
65	Segmentec	2	0	0	0	0	0	0	0	0	0	0.001387	0.002774	0.011096
66	Segmentec	2	0.233645	0.224299	0.196262	0.233645	0.271028	0.280374	0.336449	0.214953	0.457944	0.336449	0.35514	0.429907
67	Segmentec	2	1	0.065492	0.038321	0.019957	0.015366	0.015085	0.011243	0.013492	0.010681	0.010588	0.008432	0.008995
68	Segmentec	2	1	0.586957	0.425466	0.385093	0.27795	0.242236	0.27795	0.234472	0.209627	0.194099	0.220497	0.170807
69	Segmentec	2	0.741758	0.401099	0.434066	0.296703	0.313187	0.236264	0.412088	0.318681	0.313187	0.296703	0.28022	0.346154
70	Segmentec	2	0	0.01	0.04	0.02	0.02	0.1	0.06	0.07	0.04	0.07	0.11	0.11
71	Segmentec	2	1	0.411651	0.394175	0.343689	0.281553	0.306796	0.269903	0.215534	0.279612	0.250485	0.207767	0.213592
72	Segmentec	2	1	0.674286	0.582857	0.617143	0.565714	0.502857	0.468571	0.502857	0.48	0.394286	0.422857	0.525714
73	Segmentec	2	0	0.001942	0.011651	0.007767	0.017476	0.034952	0.042718	0.060194	0.252427	0.669903	0.673786	0.421359
74	Segmentec	2	1	0.674286	0.582857	0.617143	0.565714	0.502857	0.468571	0.502857	0.48	0.394286	0.422857	0.525714
75	Segmentec	2	0	0	0	0	0	0	0	0	0	0	0	0.000873
76	Segmentec	2	0	0	0.022472	0.016854	0.039326	0	0	0.005618	0	0.011236	0.016854	0
77	Segmentec	2	0	0	0.017752	0.068047	0.238166	0.5	0.868343	1	0.964497	0.810651	0.788462	0.661243
78	Segmentec	2	1	0.4241	0.333333	0.317684	0.269171	0.255086	0.215962	0.258216	0.234742	0.219092	0.173709	0.200313
79	Segmentec	2	1	0.549451	0.335165	0.35989	0.398352	0.478022	0.510989	0.549451	0.445055	0.431319	0.299451	0.288462
80	Segmentec	2	0.986486	0.445946	0.472973	0.486486	0.459459	0.418919	0.540541	0.513514	0.567568	0.621622	0.486486	0.662162

Figure 5.4.2.3 color_histogram_data.csv snippet

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

1	FileName	Label	HuMomen	HuMomen	HuMomen	HuMomen	HuMomen	HuMomen	HuMoment7
26	Segmente	0	1	0.047094	0.002669	0.000667	1.27E-07	6.38E-05	0
27	Segmente	0	1	0.000368	8.10E-06	1.20E-07	2.33E-10	0	2.33E-10
28	Segmente	0	1	0.000163	5.14E-06	1.43E-08	8.76E-12	0	8.76E-12
29	Segmente	0	1	0.000995	7.10E-05	9.86E-06	0	8.62E-08	5.24E-11
30	Segmente	1	1	0.000209	4.08E-05	5.32E-06	1.99E-08	0	1.99E-08
31	Segmente	1	1	0.000521	3.08E-05	7.15E-06	6.56E-08	0	6.56E-08
32	Segmente	1	1	0.001067	1.40E-07	3.82E-10	4.15E-19	4.79E-12	0
33	Segmente	1	1	0.000423	9.90E-07	6.40E-10	3.33E-18	4.72E-12	0
34	Segmente	1	1	0.007925	8.39E-06	1.20E-07	2.77E-14	3.19E-09	0
35	Segmente	1	1	0.000688	3.41E-09	7.05E-09	7.79E-18	1.11E-11	0
36	Segmente	1	1	7.47E-06	3.39E-07	1.47E-06	0	1.41E-09	1.54E-13
37	Segmente	1	1	0.001422	3.70E-06	6.79E-07	0	4.63E-09	2.45E-13
38	Segmente	1	1	0.000265	6.34E-07	3.41E-10	2.09E-12	0	2.09E-12
39	Segmente	1	1	0.00013	3.30E-06	9.42E-08	2.69E-11	0	2.69E-11
40	Segmente	1	1	1.98E-05	2.66E-07	1.22E-10	0	8.49E-14	1.30E-19
41	Segmente	1	1	4.46E-07	8.39E-07	4.42E-10	3.45E-19	1.17E-13	0
42	Segmente	1	1	0.001989	2.13E-06	7.59E-09	2.19E-16	8.30E-11	0
43	Segmente	1	1	0.006322	2.24E-05	1.75E-07	5.34E-09	0	5.34E-09
44	Segmente	1	1	6.60E-06	1.35E-05	2.94E-10	2.84E-18	1.59E-13	0
45	Segmente	1	1	6.86E-05	6.31E-06	3.30E-08	2.77E-15	1.06E-10	0

Figure 5.4.2.4 hu_moments_data.csv snippet

5.4.3 Classification

Done By: Khew Sei Fong & Lip Zhen Yi

User can select the classifier from the 3 models which are KNN, SVM and Random Forest.

```
Select a classifier:
1. KNN
2. SVM
3. Random Forest
4. Exit
5. Generate Reports for All Classifiers
Enter choice:
```

Figure 5.4.3.1 Selecting classifier (model)

After selecting classifier, user needs to select the features used for classification which are Color Histogram, Hu Moments and Combine Features.

```
Classifier with feature:
0. Back
1. Color
2. Hu Moments
3. Combine both Color and Hu Moments
Enter choice:
```

Figure 5.4.3.2 Selecting features

After selecting classifier and features used, the evaluation output will be generated and displayed to the users. In the evaluation output, there are prediction accuracy of model, precision, recall and F1-score for each sign.

```
K: 1, Accuracy: 94.1176%
K: 3, Accuracy: 82.3529%
K: 5, Accuracy: 76.4706%
K: 7, Accuracy: 70.5882%
K: 9, Accuracy: 58.8235%

Best K: 1, Best Accuracy: 94.1176%
1. Accuracy: 94.1176%
2. Precision, Recall, and F1 Score for each class:

Class: Informative Sign
Precision: 100%
Recall: 83.3333%
F1 Score: 90.9091%

Class: Prohibitive Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Class: Warning Sign
Precision: 88.8889%
Recall: 100%
F1 Score: 94.1177%

Average Precision: 96.2963%
Average Recall: 94.4444%
Average F1 Score: 95.0089%
4. Confusion Matrix:
   Actual\Predicted     Informative Sign    Prohibitive Sign        Warning Sign
    Informative Sign                5                   0                   1
    Prohibitive Sign                0                   3                   0
        Warning Sign                0                   0                   8
```

Figure 5.4.3.3 Sample Output of Evaluation (KNN – Color Histogram)

5.5 Implementation Issues and Challenges

Done By: Khew Sei Fong

There are several issues and challenges faced in developing this project. In the segmentation phase, the various light conditions, complex backgrounds, and objects with colors similar to the traffic signs can significantly reduce the segmentation accuracy, making it difficult to segment traffic signs effectively. This leads to incorrect or incomplete feature extraction which will directly impact the performance of the models. Moreover, the limited size of the dataset poses a challenge as it restricts the model's ability to learn and generalize effectively. With fewer training samples, the models are more prone or overfitting where they perform well on the training data but fail to generalize to the unseen testing data.

Done By: Leong Yee Chung

The process of segmenting traffic signs using both color and shape-based techniques faced several challenges, primarily due to image quality, broken contours, and limitations in defining HSV color ranges. One major issue was poor image quality, which led to broken or incomplete contours, making shape segmentation less effective. Even with the use of techniques like convex hulls and contour approximation to smooth the boundaries, the system often failed to reconstruct the full shape of the traffic sign. This caused misclassification or incomplete segmentation, as shown in **Figures 5.5.1** and **5.5.2**, where the system mistakenly detected areas as similar to a triangle, leading to incorrect segmentation.

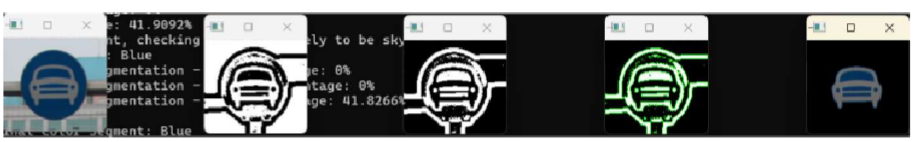

Figure 5.5.1 Shape Segmentation of Broken Contours

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

Figure 5.5.2 Shape Segmentation of Broken Contours 2

Another challenge was the difficulty in defining an optimal HSV range that could handle varying lighting conditions and image quality. Lighting variations greatly affected the accuracy of color classification, causing the system to misidentify traffic signs. In Figure 5.5.3, for instance, poor lighting conditions made it impossible for the system to detect enough yellow pixels, leading to a misclassification of the traffic sign.

Figure 5.5.3 Color Segmentation of Poor Lighting Condition

In Figure 5.5.4, although the system initially segmented the traffic sign correctly using color segmentation, it can't be segmented well due to overexposure and prevented accurate contour detection. The excessive lighting caused the contours to become fragmented, making it difficult for the system to perform accurate shape-based segmentation. This resulted in incomplete segmentation, even though the initial color segmentation was used correctly but not segmenting correctly.

Figure 5.5.4 Color Segmentation of Overexposure

5.6 Concluding Remark

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

Done by: Steffi Yim Kar Mun

In conclusion, the implementation of the traffic sign detection and classification system utilized a combination of hardware and software tools, including an HP Victus laptop, Microsoft Visual Studio, and OpenCV. The project demonstrated the effectiveness of machine learning models such as KNN, SVM, and Random Forest in identifying traffic signs based on features extracted using Colour Histograms and Hu Moments techniques. Despite challenges such as segmentation difficulties and a limited dataset, the system was able to achieve classification with measurable accuracy, precision, recall, and F1-score. These results highlight the potential of using machine learning and image processing techniques in traffic sign recognition, however further enhancements, especially in terms of dataset size and segmentation accuracy, are needed for real-world applications.

CHAPTER 6

System Evaluation and Discussion

6.1 System Testing and Performance Metrics

Done By: Khew Sei Fong

In the evaluation of this project, there are 4 evaluation metrics used for evaluating the performance of the models which are accuracy, precision, recall, and F1-score.

Accuracy is used to measure the overall correctness of the prediction of the model.

$$Accuracy = \frac{Correct\ Prediction}{Total\ Prediction}$$

Higher accuracy means that the more correct predictions are made by the model.

Precision is used to measure the accuracy of the positive predictions made by the model.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Higher precision indicates that among the instances the model predictions as positive, a large proportion are actually positive.

Recall is used to measure the ability of model to find all relevant positive instances.

$$Recall = \frac{True\ Positive}{True\ Positve\ +\ False\ Negative}$$

Higher recall means that the model correctly identified a large proportion of all the actual positive instances

F1-score is the harmonic mean of precision and recall and provides a single metric that balances both concerns.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Higher F1-score indicates a good balance between precision and recall, meaning that the model performs well in identifying positive instances with fewer false positives and false negatives.

6.2 Comparison Performance between Classifiers

Done By: Lip Zhen Yi

Classifier	Color Histogram			Hu Moments			Color Histogram + Hu Moments		
	K-NN	SVM	Random Forest	K-NN	SVM	Random Forest	K-NN	SVM	Random Forest
Accuracy (%)	94.12	100	100	70.59	52.94	94.12	70.59	100	100
Informative Sign Class Precision (%)	100	100	100	66.67	0	85.71	66.67	100	100
Informative Sign Class Recall (%)	83.33	100	100	33.33	0	100	40	100	100
Informative Sign Class F1-score (%)	90.91	100	100	44.44	0	92.31	50	100	100
Prohibitive Sign Class Precision (%)	100	100	100	33.33	27.27	100	80	100	100
Prohibitive Sign Class Recall (%)	100	100	100	66.67	100	66.67	100	100	100
Prohibitive Sign Class F1-score (%)	100	100	100	44.44	42.86	80	88.89	100	100
Warning Sign Class Precision (%)	88.89	100	100	100	100	100	66.67	100	100
Warning Sign Class Recall (%)	100	100	100	100	75	100	75	100	100
Warning Sign Class F1-score (%)	94.12	100	100	100	85.71	100	70.59	100	100

Table 6.2.1 Classifier Performance Comparison

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

In comparison of the three classifiers in table 6.2.1, Random Forest is much greater than K-NN and SVM in terms of overall accuracy. K-NN is not able to achieve 100% in all three feature sets. Although SVM has the same 100% as Random Forest in using only Color Histogram and the combination of Color Histogram and Hu Moments, it shared in Hu Moments is much lower than Random Forest which only gets 52.94%. On the contrary, Random Forest is almost ahead in all aspects. It achieved 100% in using only Color Histogram and the combination of Color Histogram and Hu Moments as well as achieved 94.12% in using only Hu Moments. This proved that Random Forest is more effective for traffic sign identification when using different feature sets.

The table also demonstrates that Random Forest offers consistently high precision, recall and also F1-score in almost all classes which are informative sign class, prohibitive sign class and warning sign class. The majority of classes achieve 100% in these parameters. For Hu Moments, Random Forest can also maintain are above 80%, only prohibitive sign class observing a little decline to 66.67% in recall. The performance of the other two classifiers, K-NN and SVM is not better than Random Forest, especially in terms of Hu Moments. The precision, recall and F1-score of the informative sign class using SVM even as low as 0%. This shows that Random Forest is more dependable and trustworthy when handling particular kinds of data.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

6.3 Error Analysis

Done By: Khew Sei Fong

The lower prediction accuracy when using Hu Moments as features, compared to other feature sets like Color Histogram or Combined Features, can be attributed to several factors. Hu Moments represents a limited set of seven invariant moments that capture shape information which may not be sufficient to distinguish between complex traffic sign shapes effectively. In contrast, Color Histogram and Combined Features provide richer information, capturing not just shape but also color, leading to better model performance.

Furthermore, the quality of the segmented images used for feature extraction is crucial in this project. The segmentation is not performed accurately, due to varying lighting conditions, complex backgrounds or similar colors between signs and surrounding objects. The extracted features may not correctly represent the traffic signs. This poor segmentation can lead to incomplete or noisy features which reduces the effectiveness of machine learning models during training and testing.

Additionally, the size of the dataset is limited and cause the low prediction accuracy for certain model. With only 84 images, split into 67 for training and 17 for testing, the dataset is too small to capture the variability and complexity of traffic sign effectively. This limited data may result in overfitting, where the model learns to perform well on training data but fails to generalize to unseen data.

6.4 Objectives Evaluation

Done by: Steffi Yim Kar Mun

The objective of this research was to develop a traffic sign detection and recognition system capable of improving autonomous vehicle navigation by meeting specific performance benchmarks, including a detection accuracy of 95% and a recognition accuracy of 95%, all while maintaining a processing speed of 30 milliseconds per image.

In evaluating the system's performance, we used three different classifiers: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest. Both the SVM and Random Forest models achieved a recognition accuracy of 100% after training and testing, surpassing the target of 95%. This demonstrates the system's robustness in correctly identifying traffic signs under various conditions.

For detection, most of the images were successfully segmented despite challenges such as varying lighting conditions. This indicates that the detection accuracy is on par with the 95% target, ensuring that most traffic signs can be isolated and recognized effectively.

Therefore, based on the data collected, we can conclude that the objectives of detection and recognition accuracy, were achieved, fulfilling the research's goals.

6.5 Concluding Remark

Done by: Steffi Yim Kar Mun

In conclusion, this research successfully achieved its primary objective of developing an advanced traffic sign detection and recognition system aimed at improving autonomous vehicle navigation. Through the evaluation of three classifiers, K-Nearest Neighbours (KNN), Support Vector Machine (SVM), and Random Forest, both SVM and Random Forest models achieved a recognition accuracy of 100%, surpassing the target of 95%. Additionally, despite challenges like varying lighting conditions, the system's detection accuracy was on par with the 95% benchmark, demonstrating its capability to isolate and recognize traffic signs under diverse conditions. This validates the system's effectiveness in enhancing the safety and efficiency of autonomous driving by ensuring accurate and timely traffic sign recognition within the required processing speed of 30 milliseconds per image.

67
Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

CHAPTER 7

CONCLUSION AND RECOMMENDATION

7.1 Conclusion

Done by: Lip Zhen Yi

This project successfully developed a system for detecting various traffic sign types using machine learning technique. In this project, various machine learning methods are used to extract the feature of traffic sign and classify them. To improve the accuracy of feature extraction and classification, the traffic signs are segmented to minimize environmental noise such as the variation in lightning. The extracted features used in this project are Hu Moments, Color Histogram and the combination of these two feature descriptors. Hu Moments are very useful in identifying various sign forms since they extract shape-based information from the traffic sign images, while Color Histogram offer important insights into the distribution of colors in the images. By combining these two feature descriptors, the system is able to provide better and more accurate traffic sign detection compared to using either feature descriptors alone. The machine learning methods used for classification are random forest, SVM and k-NN. All these classifiers contributed to the analysis of this project, but random forest is proved to be the best classifier in the end. It is the preferable option for this system as it predicts traffic sign images more accurately than both SVM and k-NN.

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

7.2 Recommendations

Done by: Khew Sei Fong

There are several improvements that will be implemented in the traffic sign detection and recognition system. Firstly, real-time capture using a camera will be integrated for real-time detection and recognition of traffic signs, making the system more effective in ADAS. Secondly, the training dataset will be expanded to include traffic signs from around the world, which will increase the system's accuracy in detecting and recognizing a broader range of signs. Lastly, a user-friendly interface will be developed to enhance user experience and make the system more intuitive for end-users.

Done by: Leong Yee Chung

One of the key improvements for future development of the traffic sign detection and recognition system is the use of deep learning models, especially Convolutional Neural Networks (CNNs). CNNs are very effective at handling image tasks like object detection and classification because they can automatically learn complex features from images. Compared to traditional methods, CNNs offer more accurate feature extraction and can improve the system's ability to detect and recognize traffic signs in different conditions. Additionally, using pre-trained models like YOLO or MobileNet through transfer learning can speed up development while keeping accuracy high. With CNNs, the system can achieve real-time detection and better recognition accuracy, making it ideal for use in autonomous driving and advanced driver assistance system

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

REFERENCES

[1] M. L. Cummings, "What self-driving cars tell us about AI risks," IEEE Spectrum, https://spectrum.ieee.org/self-driving-cars-2662494269 (accessed Sep. 7, 2024).

[2] C. Liu, S. Li, F. Chang and Y. Wang, "Machine Vision Based Traffic Sign Detection Methods: Review, Analyses and Perspectives," in IEEE Access, vol. 7, pp. 86578-86596, 2019, doi: 10.1109/ACCESS.2019.2924947.

[3] S. S. Md Sallah, F. A. Hussin, and M. Z. Yusoff, "Shape-based road sign detection and recognition for embedded application using MATLAB," *2010 International Conference on Intelligent and Advanced Systems*, Jun. 2010. doi:10.1109/icias.2010.5716193.

[4] H. Handoko, J. H. Pratama, and B. W. Yohanes, "Traffic sign detection optimization using color and shape segmentation as pre-processing system," Telkomnika, vol. 19, no. 1, p. 173-175, Feb. 2021

[5] D. Hema and S. Kannan, "Interactive Color Image Segmentation using HSV Color Space," Science and Technology Journal, vol. 7, no. 1, pp. 37–39, Jan. 2019

[6] S. B. Wali *et al.*, "Vision-based traffic sign detection and recognition systems: Current trends and challenges," *Sensors*, vol. 19, no. 9, p. 2093, May 2019. doi:10.3390/s19092093

[7] Y. Wu, YB. Liu, J. Li, H. Liu and X. Hu, "Traffic sign detection based on convolutional neural networks," The 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 2013, pp. 1-7, doi: 10.1109/IJCNN.2013.6706811.

[8] Y. Wu, Z. Li, Y. Chen, K. Nai, and J. Yuan, "Real-time traffic sign detection and classification towards real traffic scene," *Multimedia Tools and Applications*, vol. 79, no. 25–26, pp. 18201–18219, Mar. 2020. doi:10.1007/s11042-020-08722-y

[9] K. Xie et al., "Efficient Federated Learning With Spike Neural Networks for Traffic Sign Recognition," in IEEE Transactions on Vehicular Technology, vol. 71, no. 9, pp. 9980-9992, Sept. 2022, doi: 10.1109/TVT.2022.3178808.

[10] Saadna, Y., Behloul, A. An overview of traffic sign detection and classification methods. *Int J Multimed Info Retr* 6, 193–210 (2017). https://doi.org/10.1007/s13735-017-0129-8

[11] J. Cao, C. Song, S. Peng, F. Xiao, and S. Song, "Improved traffic sign detection and recognition algorithm for intelligent vehicles," *Sensors*, vol. 19, no. 18, p. 4021, Sep. 2019, doi: 10.3390/s19184021.

[12] A. Ellahyani, M. E. Ansari, and I. E. Jaafari, "Traffic sign detection and recognition based on random forests," *Applied Soft Computing*, vol. 46, pp. 805–815, Sep. 2016, doi: 10.1016/j.asoc.2015

APPENDIX A

Dataset used:

APPENDIX B

```
K: 1, Accuracy: 94.1176%
K: 3, Accuracy: 82.3529%
K: 5, Accuracy: 76.4706%
K: 7, Accuracy: 70.5882%
K: 9, Accuracy: 58.8235%

Best K: 1, Best Accuracy: 94.1176%
1. Accuracy: 94.1176%
2. Precision, Recall, and F1 Score for each class:

Class: Informative Sign
Precision: 100%
Recall: 83.3333%
F1 Score: 90.9091%

Class: Prohibitive Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Class: Warning Sign
Precision: 88.8889%
Recall: 100%
F1 Score: 94.1177%

Average Precision: 96.2963%
Average Recall: 94.4444%
Average F1 Score: 95.0089%
4. Confusion Matrix:
   Actual\Predicted     Informative Sign    Prohibitive Sign        Warning Sign
    Informative Sign                 5                  0                  1
    Prohibitive Sign                 0                  3                  0
        Warning Sign                 0                  0                  8
```

Evaluation result of KNN with color histogram

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

```
K: 1, Accuracy: 82.3529%
K: 3, Accuracy: 70.5882%
K: 5, Accuracy: 82.3529%
K: 7, Accuracy: 76.4706%
K: 9, Accuracy: 70.5882%

Best K: 1, Best Accuracy: 82.3529%

1. Accuracy: 70.5882%
2. Precision, Recall, and F1 Score for each class:

Class: Informative Sign
Precision: 66.6667%
Recall: 33.3333%
F1 Score: 44.4444%

Class: Prohibitive Sign
Precision: 33.3333%
Recall: 66.6667%
F1 Score: 44.4444%

Class: Warning Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Average Precision: 66.6667%
Average Recall: 66.6667%
Average F1 Score: 62.963%
4. Confusion Matrix:
   Actual\Predicted     Informative Sign    Prohibitive Sign        Warning Sign
    Informative Sign                2                  4                  0
    Prohibitive Sign                1                  2                  0
        Warning Sign                0                  0                  8
```

Evaluation result of KNN with hu moments

```
Running KNN Combined
K: 1, Accuracy: 88.2353%
K: 3, Accuracy: 70.5882%
K: 5, Accuracy: 58.8235%
K: 7, Accuracy: 64.7059%
K: 9, Accuracy: 64.7059%

Best K: 1, Best Accuracy: 88.2353%
1. Random Forest Accuracy: 70.5882%
2. Precision, Recall, and F1 Score for each class:

Class: Informative Sign
Precision: 66.6667%
Recall: 40%
F1 Score: 50%

Class: Prohibitive Sign
Precision: 80%
Recall: 100%
F1 Score: 88.8889%

Class: Warning Sign
Precision: 66.6667%
Recall: 75%
F1 Score: 70.5882%

Average Precision: 71.1111%
Average Recall: 71.6667%
Average F1 Score: 69.8257%
4. Confusion Matrix:
   Actual\Predicted     Informative Sign    Prohibitive Sign        Warning Sign
    Informative Sign                   2                   0                   3
    Prohibitive Sign                   0                   4                   0
        Warning Sign                   1                   1                   6
```

Evaluation result of KNN with combined features

```
C: 0.01, Gamma: 0.001, Accuracy: 17.6471%
C: 0.01, Gamma: 0.01, Accuracy: 17.6471%
C: 0.01, Gamma: 0.1, Accuracy: 17.6471%
C: 0.01, Gamma: 1, Accuracy: 17.6471%
C: 0.1, Gamma: 0.001, Accuracy: 17.6471%
C: 0.1, Gamma: 0.01, Accuracy: 17.6471%
C: 0.1, Gamma: 0.1, Accuracy: 17.6471%
C: 0.1, Gamma: 1, Accuracy: 17.6471%
C: 1, Gamma: 0.001, Accuracy: 52.9412%
C: 1, Gamma: 0.01, Accuracy: 94.1176%
C: 1, Gamma: 0.1, Accuracy: 76.4706%
C: 1, Gamma: 1, Accuracy: 23.5294%
C: 10, Gamma: 0.001, Accuracy: 88.2353%
C: 10, Gamma: 0.01, Accuracy: 100%
C: 10, Gamma: 0.1, Accuracy: 76.4706%
C: 10, Gamma: 1, Accuracy: 23.5294%
C: 100, Gamma: 0.001, Accuracy: 94.1176%
C: 100, Gamma: 0.01, Accuracy: 100%
C: 100, Gamma: 0.1, Accuracy: 76.4706%
C: 100, Gamma: 1, Accuracy: 23.5294%

Best C: 10, Best Gamma: 0.01, Best Accuracy: 100%
1. SVM Color Accuracy: 100%
2. Precision, Recall, and F1 Score for each class:

Class: Informative Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Class: Prohibitive Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Class: Warning Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Average Precision: 100%
Average Recall: 100%
Average F1 Score: 100%
4. Confusion Matrix:
   Actual\Predicted     Informative Sign    Prohibitive Sign        Warning Sign
    Informative Sign                  6                   0                   0
    Prohibitive Sign                  0                   3                   0
        Warning Sign                  0                   0                   8
```

Evaluation result of SVM with color histogram

```
C: 0.01, Gamma: 0.001, Accuracy: 17.6471%
C: 0.01, Gamma: 0.01, Accuracy: 17.6471%
C: 0.01, Gamma: 0.1, Accuracy: 17.6471%
C: 0.01, Gamma: 1, Accuracy: 17.6471%
C: 0.1, Gamma: 0.001, Accuracy: 17.6471%
C: 0.1, Gamma: 0.01, Accuracy: 17.6471%
C: 0.1, Gamma: 0.1, Accuracy: 17.6471%
C: 0.1, Gamma: 1, Accuracy: 17.6471%
C: 1, Gamma: 0.001, Accuracy: 17.6471%
C: 1, Gamma: 0.01, Accuracy: 17.6471%
C: 1, Gamma: 0.1, Accuracy: 17.6471%
C: 1, Gamma: 1, Accuracy: 17.6471%
C: 10, Gamma: 0.001, Accuracy: 17.6471%
C: 10, Gamma: 0.01, Accuracy: 17.6471%
C: 10, Gamma: 0.1, Accuracy: 17.6471%
C: 10, Gamma: 1, Accuracy: 17.6471%
C: 100, Gamma: 0.001, Accuracy: 17.6471%
C: 100, Gamma: 0.01, Accuracy: 17.6471%
C: 100, Gamma: 0.1, Accuracy: 17.6471%
C: 100, Gamma: 1, Accuracy: 52.9412%

Best C: 100, Best Gamma: 1, Best Accuracy: 52.9412%
1. SVM Hu Accuracy: 52.9412%
2. Precision, Recall, and F1 Score for each class:

Class: Informative Sign
Precision: 0%
Recall: 0%
F1 Score: 0%

Class: Prohibitive Sign
Precision: 27.2727%
Recall: 100%
F1 Score: 42.8571%

Class: Warning Sign
Precision: 100%
Recall: 75%
F1 Score: 85.7143%

Average Precision: 42.4242%
Average Recall: 58.3333%
Average F1 Score: 42.8571%
4. Confusion Matrix:
   Actual\Predicted     Informative Sign    Prohibitive Sign        Warning Sign
    Informative Sign                0                   6                   0
    Prohibitive Sign                0                   3                   0
        Warning Sign                0                   2                   6
```

Evaluation result of SVM with hu moments

```
C: 0.01, Gamma: 0.001, Accuracy: 23.5294%
C: 0.01, Gamma: 0.01, Accuracy: 23.5294%
C: 0.01, Gamma: 0.1, Accuracy: 23.5294%
C: 0.01, Gamma: 1, Accuracy: 23.5294%
C: 0.1, Gamma: 0.001, Accuracy: 23.5294%
C: 0.1, Gamma: 0.01, Accuracy: 23.5294%
C: 0.1, Gamma: 0.1, Accuracy: 23.5294%
C: 0.1, Gamma: 1, Accuracy: 23.5294%
C: 1, Gamma: 0.001, Accuracy: 35.2941%
C: 1, Gamma: 0.01, Accuracy: 100%
C: 1, Gamma: 0.1, Accuracy: 58.8235%
C: 1, Gamma: 1, Accuracy: 29.4118%
C: 10, Gamma: 0.001, Accuracy: 94.1176%
C: 10, Gamma: 0.01, Accuracy: 100%
C: 10, Gamma: 0.1, Accuracy: 58.8235%
C: 10, Gamma: 1, Accuracy: 29.4118%
C: 100, Gamma: 0.001, Accuracy: 94.1176%
C: 100, Gamma: 0.01, Accuracy: 100%
C: 100, Gamma: 0.1, Accuracy: 58.8235%
C: 100, Gamma: 1, Accuracy: 29.4118%

Best C: 1, Best Gamma: 0.01, Best Accuracy: 100%
1. SVM Combined Accuracy: 100%
2. Precision, Recall, and F1 Score for each class:

Class: Informative Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Class: Prohibitive Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Class: Warning Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Average Precision: 100%
Average Recall: 100%
Average F1 Score: 100%
4. Confusion Matrix:
   Actual\Predicted     Informative Sign    Prohibitive Sign        Warning Sign
    Informative Sign                   5                   0                   0
    Prohibitive Sign                   0                   4                   0
        Warning Sign                   0                   0                   8
```

Evaluation result of SVM with combined features

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

```
Max Depth: 5, Min Sample Count: 2, Max Categories: 2, Accuracy: 100%
Max Depth: 5, Min Sample Count: 2, Max Categories: 5, Accuracy: 100%
Max Depth: 5, Min Sample Count: 2, Max Categories: 10, Accuracy: 100%
Max Depth: 5, Min Sample Count: 5, Max Categories: 2, Accuracy: 100%
Max Depth: 5, Min Sample Count: 5, Max Categories: 5, Accuracy: 100%
Max Depth: 5, Min Sample Count: 5, Max Categories: 10, Accuracy: 100%
Max Depth: 5, Min Sample Count: 10, Max Categories: 2, Accuracy: 100%
Max Depth: 5, Min Sample Count: 10, Max Categories: 5, Accuracy: 100%
Max Depth: 5, Min Sample Count: 10, Max Categories: 10, Accuracy: 100%
Max Depth: 10, Min Sample Count: 2, Max Categories: 2, Accuracy: 100%
Max Depth: 10, Min Sample Count: 2, Max Categories: 5, Accuracy: 100%
Max Depth: 10, Min Sample Count: 2, Max Categories: 10, Accuracy: 100%
Max Depth: 10, Min Sample Count: 5, Max Categories: 2, Accuracy: 100%
Max Depth: 10, Min Sample Count: 5, Max Categories: 5, Accuracy: 100%
Max Depth: 10, Min Sample Count: 5, Max Categories: 10, Accuracy: 100%
Max Depth: 10, Min Sample Count: 10, Max Categories: 2, Accuracy: 100%
Max Depth: 10, Min Sample Count: 10, Max Categories: 5, Accuracy: 100%
Max Depth: 10, Min Sample Count: 10, Max Categories: 10, Accuracy: 100%
Max Depth: 15, Min Sample Count: 2, Max Categories: 2, Accuracy: 100%
Max Depth: 15, Min Sample Count: 2, Max Categories: 5, Accuracy: 100%
Max Depth: 15, Min Sample Count: 2, Max Categories: 10, Accuracy: 100%
Max Depth: 15, Min Sample Count: 5, Max Categories: 2, Accuracy: 100%
Max Depth: 15, Min Sample Count: 5, Max Categories: 5, Accuracy: 100%
Max Depth: 15, Min Sample Count: 5, Max Categories: 10, Accuracy: 100%
Max Depth: 15, Min Sample Count: 10, Max Categories: 2, Accuracy: 100%
Max Depth: 15, Min Sample Count: 10, Max Categories: 5, Accuracy: 100%
Max Depth: 15, Min Sample Count: 10, Max Categories: 10, Accuracy: 100%

Best Parameters -> Max Depth: 5, Min Sample Count: 2, Max Categories: 2, Best Accuracy: 100%
1. Random Forest Color Accuracy: 100%
2. Precision, Recall, and F1 Score for each class:

Class: Informative Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Class: Prohibitive Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Class: Warning Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Average Precision: 100%
Average Recall: 100%
Average F1 Score: 100%
4. Confusion Matrix:
   Actual\Predicted     Informative Sign    Prohibitive Sign       Warning Sign
    Informative Sign                   6                   0                  0
    Prohibitive Sign                   0                   3                  0
        Warning Sign                   0                   0                  8
```

Evaluation result of Random Forest with color histogram

```
Max Depth: 5, Min Sample Count: 2, Max Categories: 2, Accuracy: 82.3529%
Max Depth: 5, Min Sample Count: 2, Max Categories: 5, Accuracy: 94.1176%
Max Depth: 5, Min Sample Count: 2, Max Categories: 10, Accuracy: 100%
Max Depth: 5, Min Sample Count: 5, Max Categories: 2, Accuracy: 100%
Max Depth: 5, Min Sample Count: 5, Max Categories: 5, Accuracy: 100%
Max Depth: 5, Min Sample Count: 5, Max Categories: 10, Accuracy: 100%
Max Depth: 5, Min Sample Count: 10, Max Categories: 2, Accuracy: 94.1176%
Max Depth: 5, Min Sample Count: 10, Max Categories: 5, Accuracy: 88.2353%
Max Depth: 5, Min Sample Count: 10, Max Categories: 10, Accuracy: 100%
Max Depth: 10, Min Sample Count: 2, Max Categories: 2, Accuracy: 94.1176%
Max Depth: 10, Min Sample Count: 2, Max Categories: 5, Accuracy: 94.1176%
Max Depth: 10, Min Sample Count: 2, Max Categories: 10, Accuracy: 88.2353%
Max Depth: 10, Min Sample Count: 5, Max Categories: 2, Accuracy: 88.2353%
Max Depth: 10, Min Sample Count: 5, Max Categories: 5, Accuracy: 94.1176%
Max Depth: 10, Min Sample Count: 5, Max Categories: 10, Accuracy: 94.1176%
Max Depth: 10, Min Sample Count: 10, Max Categories: 2, Accuracy: 88.2353%
Max Depth: 10, Min Sample Count: 10, Max Categories: 5, Accuracy: 94.1176%
Max Depth: 10, Min Sample Count: 10, Max Categories: 10, Accuracy: 88.2353%
Max Depth: 15, Min Sample Count: 2, Max Categories: 2, Accuracy: 88.2353%
Max Depth: 15, Min Sample Count: 2, Max Categories: 5, Accuracy: 94.1176%
Max Depth: 15, Min Sample Count: 2, Max Categories: 10, Accuracy: 94.1176%
Max Depth: 15, Min Sample Count: 5, Max Categories: 2, Accuracy: 100%
Max Depth: 15, Min Sample Count: 5, Max Categories: 5, Accuracy: 94.1176%
Max Depth: 15, Min Sample Count: 5, Max Categories: 10, Accuracy: 94.1176%
Max Depth: 15, Min Sample Count: 10, Max Categories: 2, Accuracy: 94.1176%
Max Depth: 15, Min Sample Count: 10, Max Categories: 5, Accuracy: 88.2353%
Max Depth: 15, Min Sample Count: 10, Max Categories: 10, Accuracy: 94.1176%

Best Parameters -> Max Depth: 5, Min Sample Count: 2, Max Categories: 10, Best Accuracy: 100%
1. Random Forest Hu Accuracy: 94.1176%
2. Precision, Recall, and F1 Score for each class:

Class: Informative Sign
Precision: 85.7143%
Recall: 100%
F1 Score: 92.3077%

Class: Prohibitive Sign
Precision: 100%
Recall: 66.6667%
F1 Score: 80%

Class: Warning Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Average Precision: 95.2381%
Average Recall: 88.8889%
Average F1 Score: 90.7692%
4. Confusion Matrix:
   Actual\Predicted    Informative Sign    Prohibitive Sign       Warning Sign
    Informative Sign                  6                  0                  0
    Prohibitive Sign                  1                  2                  0
        Warning Sign                  0                  0                  8
```

Evaluation result of Random Forest with hu moments

```
Max Depth: 5, Min Sample Count: 2, Max Categories: 2, Accuracy: 100%
Max Depth: 5, Min Sample Count: 2, Max Categories: 5, Accuracy: 100%
Max Depth: 5, Min Sample Count: 2, Max Categories: 10, Accuracy: 100%
Max Depth: 5, Min Sample Count: 5, Max Categories: 2, Accuracy: 100%
Max Depth: 5, Min Sample Count: 5, Max Categories: 5, Accuracy: 100%
Max Depth: 5, Min Sample Count: 5, Max Categories: 10, Accuracy: 100%
Max Depth: 5, Min Sample Count: 10, Max Categories: 2, Accuracy: 100%
Max Depth: 5, Min Sample Count: 10, Max Categories: 5, Accuracy: 100%
Max Depth: 5, Min Sample Count: 10, Max Categories: 10, Accuracy: 100%
Max Depth: 10, Min Sample Count: 2, Max Categories: 2, Accuracy: 100%
Max Depth: 10, Min Sample Count: 2, Max Categories: 5, Accuracy: 100%
Max Depth: 10, Min Sample Count: 2, Max Categories: 10, Accuracy: 100%
Max Depth: 10, Min Sample Count: 5, Max Categories: 2, Accuracy: 100%
Max Depth: 10, Min Sample Count: 5, Max Categories: 5, Accuracy: 100%
Max Depth: 10, Min Sample Count: 5, Max Categories: 10, Accuracy: 100%
Max Depth: 10, Min Sample Count: 10, Max Categories: 2, Accuracy: 100%
Max Depth: 10, Min Sample Count: 10, Max Categories: 5, Accuracy: 100%
Max Depth: 10, Min Sample Count: 10, Max Categories: 10, Accuracy: 100%
Max Depth: 15, Min Sample Count: 2, Max Categories: 2, Accuracy: 100%
Max Depth: 15, Min Sample Count: 2, Max Categories: 5, Accuracy: 100%
Max Depth: 15, Min Sample Count: 2, Max Categories: 10, Accuracy: 100%
Max Depth: 15, Min Sample Count: 5, Max Categories: 2, Accuracy: 100%
Max Depth: 15, Min Sample Count: 5, Max Categories: 5, Accuracy: 100%
Max Depth: 15, Min Sample Count: 5, Max Categories: 10, Accuracy: 100%
Max Depth: 15, Min Sample Count: 10, Max Categories: 2, Accuracy: 100%
Max Depth: 15, Min Sample Count: 10, Max Categories: 5, Accuracy: 100%
Max Depth: 15, Min Sample Count: 10, Max Categories: 10, Accuracy: 100%

Best Parameters -> Max Depth: 5, Min Sample Count: 2, Max Categories: 2, Best Accuracy: 100%
1. Random Forest Combined Accuracy: 100%
2. Precision, Recall, and F1 Score for each class:

Class: Informative Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Class: Prohibitive Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Class: Warning Sign
Precision: 100%
Recall: 100%
F1 Score: 100%

Average Precision: 100%
Average Recall: 100%
Average F1 Score: 100%
4. Confusion Matrix:
   Actual\Predicted     Informative Sign    Prohibitive Sign       Warning Sign
    Informative Sign                   5                   0                  0
    Prohibitive Sign                   0                   4                  0
        Warning Sign                   0                   0                  8
```

Evaluation result of Random Forest with combined features

```
   Actual\Predicted     Informative Sign    Prohibitive Sign       Warning Sign
    Informative Sign              6                 0                  0
    Prohibitive Sign              0                 3                  0
        Warning Sign              0                 0                  8

Classifier          Feature Type   Accuracy  Precision      Recall         F1 Score
KNN                 color          94.1176   96.3           88.89          91.37
KNN                 combine        94.12     96.3           88.89          91.37
KNN                 hu             70.59     66.67          66.67          62.96
Random Forest       color          100       100            100            100
Random Forest       combine        100       100            100            100
Random Forest       hu             94.12     95.24          88.89          90.77
SVM                 color          88.24     86.67          90.28          86.41
SVM                 combine        88.24     86.67          90.28          86.41
SVM                 hu             52.94     42.42          58.33          42.86

Select a classifier:
1. KNN
2. SVM
3. Random Forest
4. Exit
5. Generate Reports for All Classifiers
Enter choice:
```

Overall classifier model's performance

Plagiarism Check Result

CHAPTER 7

APPENDIX

B
F

T7_G14_Leong Yee Chung.pdf

ORIGINALITY REPORT

15%	7%	10%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1 Submitted to Universiti Tunku Abdul Rahman
Student Paper

2 eprints.utar.edu.my
Internet Source

3 Ashok Kumar, Geeta Sharma, Anil Sharma, Pooja Chopra, Punam Rattan. "Advances in Networks, Intelligence and Computing - International Conference on Networks, Intelligence and Computing (ICONIC-2023)", CRC Press, 2024
Publication

4 "Proceedings of the 4th International Conference on Advances in Computational Science and Engineering", Springer Science and Business Media LLC, 2024
Publication

5 ejmcm.com
Internet Source

6 Chuns

3%

1%

<1%

<1%

<1%

<1%

13

14

7

8

9

10

11

12

Si
gn
D
et
ec
tio
n
M
et
ho
ds:
R
ev
ie
w,
An
al
ys
es
an
d
Pe
rs
pe
cti

ves", IEEE Access, 2019
Publication

www.frontiersin.org
Internet Source

Submitted to Manipal University
Student Paper

Parikshit N. Mahalle, Namrata N. Wasatkar, Gitanjali R. Shinde. "Data-Centric Artificial Intelligence for Multidisciplinary Applications", CRC Press, 2024
Publication

ebin.pub
Internet Source

www.mdpi.com
Internet Source

Wu, Yihui, Yulong Liu, Jianmin Li, Huaping Liu, and Xiaolin Hu. "Traffic sign detection based on convolutional neural networks", The 2013 International Joint Conference on Neural Networks (IJCNN), 2013.
Publication

Submitted to University of Lancaster
Student Paper

Anand Kumar Mishra, Amit Kumar Tyagi, Richa, Subhra Rani Patra. "Chapter 14

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

20

15

16

17

18

19

Introduction to Machine Learning and Artificial Intelligence in Banking and Finance", Springer Science and Business

Media LLC, 2024
Publication

Sneha Agrawal, Rahul Kumar Chaurasiya. "Ensemble of SVM for Accurate Traffic Sign Detection and Recognition", Proceedings of the International Conference on Graphics and Signal Processing - ICGSP '17, 2017
Publication

<1%

elibrary.tucl.edu.np
Internet Source

<1%

Submitted to Georgia Institute of Technology Main Campus
Student Paper

pdfs.semanticscholar.org
Internet Source

Subhash Chand Agrawal, Rajesh Kumar Tripathi, Ashutosh Gupta, Manish Sahu. "Traffic Sign Recognition System using CNN", 2023 World Conference on Communication & Computing (WCONF), 2023
Publication

<1%

<1%

Submitted to Universiti Teknologi Malaysia
Student Paper

<1%

www.sigma-computer.com

<

1

%

21 Internet Source

28

22 "Advances in Computational Collective Intelligence", Springer Science and Business

23

24

25

26

27

Media LLC, 2024

Publication

Internet Source

M. Emre Celebi, Teresa Mendonca, Jorge S. Marques. "Dermoscopy Image Analysis", CRC Press, 2019

Publication

<1%

<1%

Mohammad Shamsul Arefin, M Shamim Kaiser, ASM Kayes. "Applied Intelligence for Industry 4.0", CRC Press, 2023

Publication

<1%

Yasir Ali Al Khafaji, Nidhal K. El Abbadi. "Traffic Signs Detection and Recognition Using A combination of YOLO and CNN", 2022 Iraqi International Conference on Communication and Information Technologies (IICCIT), 2022

Publication

<1%

<1%

open_jicareport.jica.go.jp

dokumen.pub

Internet Source

fud.edu.ng

Internet Source

<1%

< 1%

<1%

35

29

36

30

31

32

33

34

www.astesj.com

Internet Source

www.researchgate.net

Internet Source

Anshul Verma, Pradeepika Verma. "Research Advances in Intelli gent Computing - Volume 2", CRC Press,

2024
Publication

<1%

Berke Cansiz, Coskuvar Utkan Kilinc, Gorkem Serbes. "Tunable Q-factor wavelet transform based lung signal decomposition and statistical feature extraction for effective lung disease classification", Computers in Biology and Medicine, 2024
Publication

<1%

<1%

R. N. V. Jagan Mohan, Vasamsetty Chandra Sekhar, V. M. N. S. S. V. K. R. Gupta. "Algorithms in Advanced Artificial Intelligence", CRC Press, 2024
Publication

<1%

eprints.staffs.ac.uk
Internet Source

scholar.archive.org
Internet Source

<1%

www.grafiati.com
Internet Source

<1%

< 1%

<1%

43

37

44

38

39

40

41

42

www.ijraset.com
Internet Source

Yuanyuan Long, Yong Fan, Ya Pan. "Metamorphic Testing for Traffic Sign Detection and Recognition", 2023

IEEE 23rd International Conference on Software Quality, Reliability, and Security Companion (QRS-C), 2023
Publication
<1%

www.isaacpub.org
Internet Source
<1%

Zhengzhi Zhang. "Classification of Bidens in wheat farms", International Journal of Computer Applications in Technology, 2010
Publication
<1%

centaur.reading.ac.uk
Internet Source
<1%

Anju J. Prakash, S. Sruthy. "Enhancing traffic sign recognition (TSR) by classifying deep learning models to promote road safety", Signal, Image and Video Processing, 2024
Publication
<1%

Submitted to De Montfort University
Student Paper
<1%

Hana Ben Fredj, Amani Chabbah, Jamel Baili, Hassen Faiedh, Chokri Souani. "An efficient
<1%

<1%

<1%

51

45

46

47

48

49

50

im
ple
me
nta
tio
n
of
tra
ffic
sig
ns
rec
og
niti
on
sys
te
m
usi
ng
CN
N",
Mi
cro
pro

cessors and Microsystems, 2023
Publication

Submitted to Swiss School of Business and Management - SSBM
Student Paper

<1%

Yassmina Saadna, Ali Behloul. "An overview of traffic sign detection and classification methods", International Journal of Multimedia Information Retrieval, 2017
Publication

<1%

patents.justia.com
Internet Source

tigerprints.clemson.edu
Internet Source

<1%

Emin Guney, Cuneyt Bayilmis, Batuhan Cakan. "An Implementation of Real-Time Traffic Signs and Road Objects Detection Based on Mobile GPU Platforms", IEEE Access, 2022
Publication

<1%

<1%

Submitted to Multimedia University
Student Paper

Submitted to University of Melbourne
Student Paper

<1%

< 1%

52 56

53

54

55

Ameur Zaibi, Anis Ladgham, Anis Sakly. "A Lightweigh

t Model for Traffic Sign Cla

ssification Based on Enhanced LeNet-5 Network", Journal of Sensors, 2021
Publication

José-Francisco Díez-Pastor, Pedro Latorre-Carmona, Álvar Arnaiz-González, Antonio Canepa-Oneto et al. "Towards automatic phytolith classification using feature extraction and combination strategies", Progress in Artificial Intelligence, 2024
Publication

Miguel Lopez-Montiel, Ulises Orozco-Rosas, Moisés Sánchez-Adame, Kenia Picos, Oscar Montiel. "Chapter 5 Evaluation of Deep Learning Algorithms for Traffic Sign Detection to Implement on Embedded Systems", Springer Science and Business Media LLC, 2021
Publication

Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security - Volume 2", CRC Press, 2023
Publication

Saravanan Krishnan, Ramesh Kesavan, B.

Surendiran, G. S. Mahalakshmi. "Handbook of

<
1
%

<1%

<
1
%

<1%

<
1
%

Artificial Intelligence in Biomedical Engineering", Apple Academic Press, 2021
Publication

57 arxiv.org
Internet Source

58 ijritcc.org
Internet Source

59 journals.uhd.edu.iq
Internet Source

60 "Computer Vision - ECCV 2016 Workshops", Springer Nature, 2016
Publication

61 AliAsgar Padaria, Aryan Alpesh Mehta, Nilesh Kumar Jadav, Sudeep Tanwar et al. "Traffic Sign Classification for Autonomous Vehicles Using Split and Federated Learning Underlying 5G", IEEE Open Journal of Vehicular Technology, 2023
Publication

62 Any Gupta, Ayesha Choudhary. "A Framework for Real-time Traffic Sign Detection and Recognition using Grassmann Manifolds", 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018
Publication

<1% <1%

<1%

<1% <1%

<1%

63

68

64

65

66

67

Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security", CRC Press, 2023
Publication

Chunsheng Liu, Faliang Chang, Zhenxue Chen. "High performance traffic sign recognition based on sparse representationand SVM classification", 2014 10th International Conference on Natural Computation (ICNC), 2014
Publication <1%

David A. Rosenbaum, Jonathan Vaughan, Brad Wyble. "MATLAB for Behavioral Scientists", Routledge, 2014
Publication <1%

Jong Wan Hu. "Advanced Materials and Structural Engineering", CRC Press, 2016
Publication

K. Aditya Shastry, H.A. Sanjay, G. Deexith. "Quadratic-radial-basis-function-kernel for classifying multi-class agricultural datasets with continuous attributes", Applied Soft Computing, 2017
Publication <1%

Lecture Notes in Computer Science, 2015.
Publication <1%

<1%

<1%

74

69

75

70

71

72

73

Lu Chen gEn. "An efficient image segmentation method with application to cell images", 2008 9th International

Conference on Signal Processing, 10/2020

08

Publication

<1%

Luis Unzueta. "Kinetic Pseudo-energy History for Human Dynamic Gestures Recognition", Lecture Notes in Computer Science, 2008

Publication

<1%

Shaina ., Kaan Kirant, Noman Nawaz, Olufemi Moses. "A Comparison among Machine Learning and Deep Learning Approaches for Text Classification", PsyArXiv, 2023

Publication

<1%

Zhishan Li, Mingmu Chen, Yifan He, Lei Xie, Hongye Su. "An Efficient Framework for Detection and Recognition of Numerical Traffic Signs", ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022

Publication

<1%

acikerisim.cumhuriyet.edu.tr

Internet Source

cn.technave.com

Internet Source

<1%

vuir.vu.edu.au

Internet Source

< 1%	<1%

76	"Ubiquitous Intelligent Systems", Springer Science and Business Media LLC, 2022 Publication	<1%
77	Altaf Alam, Zainul Abdin Jaffery. "Indian Traffic Sign Detection and Recognition", International Journal of Intelligent Transportation Systems Research, 2019 Publication	<1%
78	Wen, Chenglu, Jonathan Li, Huan Luo, Yongtao Yu, Zhipeng Cai, Hanyun Wang, and Cheng Wang. "Spatial-Related Traffic Sign Inspection for Inventory Purposes Using Mobile Laser Scanning Data", IEEE Transactions on Intelligent Transportation Systems, 2015. Publication	<1%
79	Yaqin Wu, Wangli Hao, Jianjun Niu, Hao Wang. "YOLO-based lightweight traffic sign detection algorithm and Mobile deployment",Research Square Platform LLC, 2024 Publication	<1%

Exclude quotes	On	Exclude assignment template	On
Exclude bibliography	On	Exclude matches	Off