Detection of Motorcycle Headlights Using YOLOv5 and HSV

Vessa Rizky Oktavia[1*], Ahmad Wali Satria Bahari Johan[2], Whisnumurty Galih Ananta[3], Fahril Refiandi[4], Muhammad Khuluqil Karim[5]

[1,2,3,4,5] Informatics Department, Institut Teknologi Telkom Surabaya, Surabaya, East Java, Indonesia
Email: [1*] vessa@ittelkom-sby.ac.id, [2] ahmadsatria13@ittelkom-sby.ac.id,
[3] whisnumurty.galih.21@student.if.ittelkom-sby.ac.id, [4] fahril.refiandi.21@student.if.ittelkom-sby.ac.id,
[5] muhammad.khuluqil.21@student.if.ittelkom-sby.ac.id

(Article received: 10 Aug 2023, revised: 25 Sep 2023, accepted: 27 Sep 2023)

Abstract

"Electronic Traffic Law Enforcement" (ETLE) denotes a mechanism that employs electronic technologies to implement traffic regulations. This commonly entails utilizing a range of electronic apparatuses like cameras, sensors, and automated setups to oversee and uphold traffic protocols, administer fines, and enhance road security. ETLE systems are frequently utilized for identifying and sanctioning infractions like exceeding speed limits, disregarding red lights, and turning off the headlights. In Indonesia, there is currently no dedicated system designed to detect traffic violation, especially regarding vehicle headlights. Therefore, this research was conducted to detect vehicle headlights using digital images. With the results of this study, it will be possible to develop a system capable of classifying whether vehicle headlights are on or off. This research employed the deep learning method in the form of the YOLOv5 model, which achieved an accuracy of 94.12% in detecting vehicle images. Furthermore, the white color extraction method was performed by projecting the RGB space to HSV to detect the Region of Interest (ROI) of the vehicle headlights, achieving an accuracy of 73.76%. The results of this vehicle headlight detection are influenced by factors such as lighting, image capture angle, and vehicle type.

Keywords: Digital Image Processing, Deep Learning, YOLOv5, HSV, ETLE.

I. INTRODUCTION

Several regulations regarding motorcycle vehicles are enforced in Indonesia to ensure the safety of riders, passengers, and other road users. Motorcycles pose higher risks compared to other vehicles due to their smaller size, higher speed, and lower stability. Therefore, regulations concerning motorcycle attributes and riders are implemented to reduce the number of accidents [1]. One of these regulations mandates that every motorcycle must have a headlight installed to illuminate the road and improve the motorcycle rider's visibility. However, some motorcycles still do not have their headlights properly installed.

Currently, the police are implementing computer technology to detect violations committed by drivers on the roads. The implementation of Electronic Traffic Law Enforcement (ETLE) in 34 regional police departments (Polda) facilitates traffic law enforcement efforts in Indonesia. ETLE assists the police in enforcing the law against traffic violations, and CCTV cameras can monitor 12 types of violations. However, only two types of violations, namely riders not wearing helmets and those who violate lane markings, can be identified automatically. For other violations, the police need to manually monitor CCTV cameras to detect incidents, such as riders not using headlights on their vehicles.

In detecting motorcycle headlights, the system first needs to detect the motorcycle object itself. Most research uses machine learning and deep learning models to detect vehicles and motorcycles. A study conducted by Jorge E. Espinosa utilized the deep learning CNN method with the faster R-CNN model to detect motorcycles [2]. Subsequently, many researchers experimented with deep learning methods to detect motorcycles using digital images. On the other hand, research conducted by Sutikno used back-propagation and Support Vector Machine methods to detect motorcycles [3]. This research is using conventional machine learning methods, indicating that machine learning can also be a solution to image detection when combined with neural networks. The models used for image detection vary depending on the object, and YOLO (You Only Look Once) has been recognized as a state-of-the-art model for various research projects [4][5][6][7], including vehicle image detection.

After obtaining the motorcycle object, the next step is to detect the headlights. Several studies have discussed headlight detection systems using digital images. For instance, Qingyan

Wang conducted research using the YOLO v4 method to detect traffic lights [8]. In our case, the system was developed to detect only the Region of Interest (ROI) of headlights using HSV color dimensions to minimize space usage. Compared to RGB, HSV is claimed to be more useful in detecting lights [9]. In images, the high intensity of lights can increase the contrast of the image and reduce the saturation to zero, and proper preprocessing is essential to avoid ground truth distortion [10]. HSV also represents colors in a more distinguishable dimension than RGB [11]. Many other activities, such as object localization and object classification, can also benefit from using HSV [12][13][14].

This research aims to detect motorcycle headlights based on digital image data obtained from CCTV installed on the roads. Researchers use the YOLOv5 method since YOLOv5 is optimized for real-time object detection, making it suitable for applications where you need to process images or video frames quickly, such as monitoring traffic. Several stages are involved in detecting motorcycle headlights. The first stage involves the system's ability to detect motorcycles using the YOLO v5 architecture, which we believe is better suited for this case than YOLO v4 [15][16]. In future research, other YOLO models might be used depending on model improvements [17][18]. In the subsequent stage, the system detects the presence of headlights based on the Region of Interest (ROI) images of the motorcycles obtained from the previous process using the white color extraction method. This research's development is the beginning of a larger system intended to ease Electronic Traffic Law Enforcement (ETLE). The hope is that this system can be developed into an integrated system that can help solve Electronic Traffic Law Enforcement (ETLE) issues, just like many other systems designed to detect various traffic law violations [19][20][21][22].

II. RESEARCH METHODOLOGY

The methodology used in this research consists of data collection, detection of Region of Interest (ROI) for two-wheeled motor vehicles, and detection of ROI for vehicle headlights. The dataset used in this study comprises videos captured using CCTV cameras. The detection of ROI for motor vehicles is performed using deep learning with YOLOv5 [16]. Subsequently, the detection of ROI for vehicle headlights is conducted using the HSV model. Figure 1 illustrates the steps of each method used in detecting motorcycle headlights.

A. Data Acquisition

The dataset used in this research was obtained from CCTV video recordings in Malang city at cctv.malangkota.go.id. The city government of Malang, East Java, monitors traffic conditions using CCTV cameras. The captured CCTV camera footage can be accessed directly and free of charge by internet users through a browser. The CCTV camera used in this research has a frame rate of 25 FPS. This frame rate is commonly used in regions that follow the PAL video standard, such as Europe and some parts of Asia, and it provides smooth video playback.

The main page displays a map image with camera icons indicating the locations of CCTV cameras used to monitor specific areas shown on the map. The displayed map represents the region of Malang City. To access live CCTV monitoring of a particular area, users can simply click on the camera icon near the desired area.

The cameras used in this research are CCTV cameras monitoring 'Terusan Sulfat' street. Video recordings were taken from these CCTV cameras and used as the dataset. The captured screens that will be used as the dataset should clearly show vehicles with visible headlights. For this purpose, the screens are cropped to focus only on the road area where the vehicles are heading towards the camera, not the other way around. The image size is adjusted to ensure that the vehicles are fully visible and not cut off.

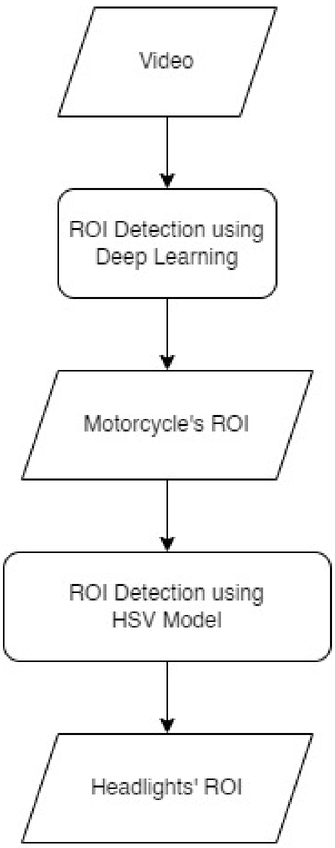

Figure 1. Research Methodology

B. YOLO Model

Deep learning is a method commonly applied in research involving images. Deep learning is used because it is considered fast and has a relatively high level of accuracy. In this research, deep learning is used to detect whether a passing vehicle is a two-wheeled motorcycle or not [4]. The dataset desired for this research consists of images of two-wheeled motorcycles taken from the front view only. The use of deep learning methods is typically packaged in the form of pre-trained models ready for use. These models can be obtained from Python libraries, ready to be adapted and modified for specific user needs. One of the trending deep learning models is You Only Look Once (YOLO).

YOLO was created in 2015 by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. YOLO is a real-time object detection algorithm that has become the state of the art. The YOLO model uses the COCO dataset for training and utilizes only CNN in its architecture, with just one fully connected layer. YOLO has evolved, with each version experiencing changes. The version used in this research is YOLOv5, which is an advancement of YOLOv3 built by Glenn Jocher and supported by Ultralytics. YOLOv5 is a single-stage object detector. It consists of three components: Backbone, Neck, and Head [18].

The backbone component consists of a pre-trained network used to extract the main features from the representation of an image. This helps reduce the spatial resolution of the image and enhance feature resolution. The neck component is utilized to extract features in a pyramid manner. This enables the model to identify features on objects of various sizes and scales. The head component is the finalization model, serving as the last stage of operations. It applies anchor boxes to the feature map and produces the final output, which includes class predictions, objective scores, and bounding boxes. The YOLO architecture, being a one-stage object detector, can be visualized as shown in Figure 2.

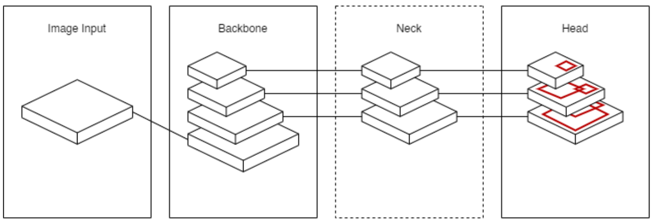

Figure 2. YOLO’s Architecture

In this research, we use the latest model of YOLOv5. The process begins with taking an input image and preparing it for detection. YOLOv5 resizes the image to a fixed size (1020 x 600 pixels) to ensure consistency. It also normalizes pixel values to bring them into a standardized range. This preprocessed image is then passed through the neural network.

The core of YOLOv5 is a deep convolutional neural network (CNN), which is responsible for feature extraction and object detection. This network typically includes a CSPDarknet53 backbone, which is a series of convolutional layers designed to capture hierarchical features from the input image. These layers are followed by additional convolutional and detection layers. The detection layers predict bounding boxes, class probabilities, and objectness scores at multiple scales. Specifically, for each grid cell in the feature map, the model predicts a set of bounding boxes (typically 3 to 5), each characterized by its coordinates (x, y, width, height), confidence score (indicating the likelihood of containing an object), and class probabilities (indicating the object's class). These predictions are made at different scales to handle objects of various sizes.

After the network's forward pass, YOLOv5 applies post-processing steps to refine the predictions. This includes removing bounding boxes with low confidence scores to filter out weak detections. It also performs non-maximum suppression (NMS) to eliminate duplicate and overlapping detections, ensuring that each object is represented by a single bounding box with the highest confidence score.

The final output of YOLOv5 is a list of bounding boxes that represent the detected objects in the input image, and the model has successfully localized and classified them.

C. Hue Saturation Value (HSV)

The Hue Saturation Value (HSV) model is a color model commonly used in image processing and object recognition. Unlike the RGB (Red, Green, Blue) color model, which is also widely used, HSV utilizes three main components: Hue, Saturation, and Value. The HSV model can be visualized in Figure 3.

Hue (H) represents the color attribute that can be easily identified, such as red, green, blue, yellow, and others. Conceptually, Hue can be represented in the form of a color circle, with red at one end and continuing in a clockwise direction until reaching red again. In this model, Hue is measured in degrees, typically ranging from 0 to 360.

Saturation (S) describes the level of color purity in the image. As Saturation approaches 0, the color becomes increasingly pale or tends towards gray. Higher Saturation values result in more intense or vivid colors. Saturation is measured as a percentage from 0 to 100%, with 0% producing a shade of gray and 100% yielding full color.

Value (V) represents the brightness or darkness of the color. As Value approaches 0, the color becomes darker or black, while higher Value values result in brighter colors. Value is also measured as a percentage from 0 to 100%.

The HSV model can be instrumental in object detection through various methods, including color segmentation, contrast enhancement, and edge detection. In the HSV model, the Hue component can be used to identify specific color ranges relevant to the object of interest [12]. For example, to detect a green tennis ball on a court, we can identify the Hue range corresponding to the green color and isolate the object within that range.

The Value component can be used to enhance the color contrast between the object and the background. By adjusting the Value, the color contrast between the object and the background can be accentuated, facilitating object detection. By combining information from the Hue and Saturation components, we can extract edges or boundaries between the object and the background. These edges often provide strong cues for object detection.

The use of HSV in this research rather than RGB is because when we want to detect objects based on their color, HSV can be more robust to changes in lighting conditions compared to RGB. In RGB, lighting changes affect all three color channels simultaneously, making it challenging to isolate color information. In HSV, changes in lighting mainly affect the value channel, which can be more easily compensated for.

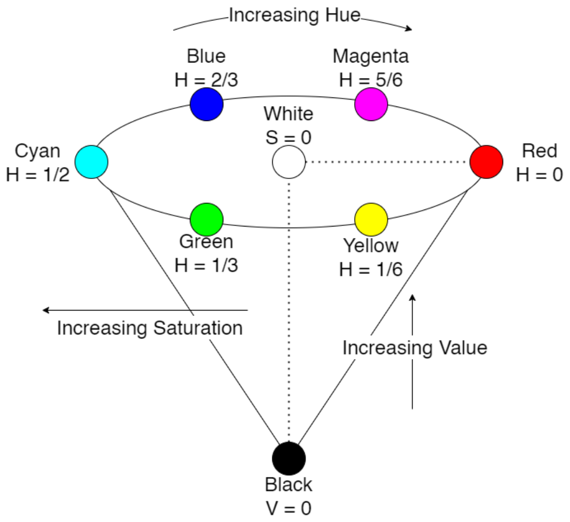

Figure 3. HSV Model Color Space

III. RESULT AND DISCUSSION

A. Dataset

The dataset used in this research is divided into two parts. The first dataset was obtained from videos containing images of streets and vehicles passing through a 1920 x 1080 pixel screen. Then, the images were cropped according to predefined frame sizes to obtain the first dataset, which is the dataset of two-wheeled vehicles. The first dataset consists of images containing multiple vehicles facing the camera. These images will be trained using the YOLOv5 model to obtain vehicle ROIs. The number of images in the first dataset is 102.

After obtaining the images of two-wheeled motor vehicles marked with ROIs, these images will be used as the second dataset. The second dataset consists of images of each vehicle that has been correctly detected as a two-wheeled motor vehicle. These images will serve as inputs for the process of detecting the ROI of the vehicles' headlights using the HSV method. The number of images used in the second dataset is 263.

B. Detection of Two-Wheeled Vehicle ROI using YOLOv5

Deep learning is a method commonly applied in research involving In detail, the YOLOv5 components utilize CSPDarknet53 for the Backbone, SPP, and PANet for the Neck, and the same architecture as YOLOv4 for the Head with modified formulas.

The YOLOv5 Backbone employed in this research comprises the CSPDarknet53 components. CSPDarknet53 is a Convolutional Neural Network (CNN) architecture developed based on Darknet53, which is a pre-trained model for object recognition.

The CSPDarknet53 architecture introduces the concept of Cross Stage Partial Network (CSP) to enhance object detection performance. In the CSPDarknet53 architecture, the input image first passes through several convolutional layers to extract high-level features. However, instead of directly proceeding to the next convolutional layers, these features are divided into two parallel paths. One parallel path is the main path that proceeds to the next convolutional layers unchanged. The second path, called the branch path, goes through a series of convolutional layers and combines them with features from the main path. This is done to enhance the learning of richer and more effective features.

CSPDarknet53 utilizes the two-path method to improve efficiency and accuracy in object detection. By combining features from the main path and the branch path, this architecture can extract more comprehensive information from the images and produce better object predictions.

The YOLOv5 Neck used in this research employs Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PANet). Spatial Pyramid Pooling (SPP) is used to capture spatial information at different scales in the image. SPP addresses the issues of object size variation and spatial layout by dividing the image into several regions and extracting features from each region independently.

The working principle of SPP involves dividing the input image into multiple predetermined levels or layers, each representing a different scale or resolution. At each level, pooling operations, such as max pooling, are conducted to capture the most prominent features within the region. The pooling operations combine features in each region, reducing dimensionality while retaining essential features.

The main idea behind SPP is to divide the image into multiple regions and pool features within each region. The network can capture fine details and global contextual information. This approach enables the network to effectively handle objects with different sizes and spatial configurations.

One main advantage of SPP is that it allows the use of fixed-size feature vectors regardless of the size of the input image. This fixed-size representation is beneficial for passing features to subsequent layers or for further processing and classification tasks.

Furthermore, the Neck section within YOLOv5 also uses PANet modified with Bottleneck CSP. PANet (Path Aggregation Network) is a component used in the YOLOv5 architecture to enhance feature representation and information flow in the network. PANet addresses the challenge of integrating multi-scale features from different levels in the network hierarchy for improved object detection performance.

The primary goal of PANet is to combine features from different spatial resolutions and merge them to capture local and global contextual information. The PANet structure consists of two main paths: the lateral path and the top-down path.

The lateral path takes feature maps from higher-resolution layers and reduces their spatial dimensions to match the resolution of the feature maps from lower-resolution layers. This is achieved by applying 1x1 convolutions to the higher-resolution feature maps. Lateral connections enable the merging of fine details from higher-resolution layers with coarser-resolution feature maps.

The top-down path takes feature maps from lower-resolution layers and upsamples them to match the resolution of the feature maps from higher-resolution layers. Upsampling is performed using bilinear interpolation or similar techniques.

The top-down path allows the propagation of global contextual information from lower-resolution layers to higher-resolution layers.

After the lateral and top-down paths process the feature maps, the merged feature map is obtained by element-wise summation or merging of the corresponding feature maps from both paths. This merging enables the integration of multi-scale features, capturing local details and global context.

The combination of PANet and Bottleneck CSP aims to enhance object detection performance by effectively integrating multi-scale features. By employing Bottleneck CSP, PANet can achieve improved feature merging from different spatial resolutions, resulting in better capturing of local and global contextual information.

In the modified PANet structure using Bottleneck CSP, the lateral and top-down paths are applied similarly to the original PANet. However, the difference lies in the utilization of Bottleneck CSP in each path. Bottleneck CSP is implemented in each lateral and top-down path to reduce feature dimensionality and enhance computational efficiency.

In the lateral path, Bottleneck CSP is used to reduce the feature dimensionality of the higher-resolution layer before merging it with the features from the lower-resolution layer. This helps retain important information in the feature representation with lower computational overhead.

In the top-down path, Bottleneck CSP is used to perform dimensionality reduction on the features from the lower-resolution layer before the upsampling process to match the feature resolution of the higher-resolution layer. By reducing dimensionality before upsampling, Bottleneck CSP improves the quality of the features spread to higher-resolution layers.

Through the application of Bottleneck CSP in PANet, the YOLOv5 model can obtain better feature representations with improved computational efficiency. By integrating features from various scales and retaining relevant information, PANet with Bottleneck CSP enhances the model's ability to detect objects with different sizes and spatial configurations.

The activation function used in all hidden layers within the YOLOv5 architecture is Leaky ReLU. Leaky ReLU (Rectified Linear Unit) is an activation function used in artificial neural networks to introduce non-linearity to the output of different layers. Leaky ReLU is similar to the standard ReLU, but it has a small positive parameter to allow for better gradient flow on negative values.

$$f(x) = max(0.01x, x) \tag{1}$$

The Leaky ReLU activation function anticipates negative values of x and replaces them with a very small value, namely 0.01, multiplied by the original x, as shown in Equation (1). By making this small modification, the gradient on the left side of the graph becomes non-zero. Therefore, during model execution, there will be no inactive neurons in any region.

The head section in the YOLOv5 architecture refers to the final layers of the network responsible for generating object location and class predictions.

The head in YOLOv5 consists of several convolutional layers and linear layers. After passing through the deeper and more complex convolutional layers in the network, the extracted features from the input image are transformed into more abstract representations. These convolutional layers are responsible for detecting objects at different scales and extracting more specific features.

After passing through the convolutional layers, these features are converted into a 1D representation using linear layers. These linear layers are used to map these features into a vector with dimensions corresponding to the number of object classes to be detected. Each element in this vector represents the object probability of each class.

Next, the output of these linear layers is processed using the sigmoid activation function to obtain the probabilities of the object classes, as described in Equation (2). YOLOv5 uses the sigmoid activation function on its output layer because this function is well-suited for mapping values into the probability range from 0 to 1. Additionally, the use of the sigmoid function on the output layer of YOLOv5 allows for the merging of prediction results from multiple bounding boxes within one image. Furthermore, predictions are also made for the bounding box coordinates representing the object locations in the image.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

After determining the use of the YOLOv5 model for training, the first step in detecting two-wheeled vehicles is to establish the frame size. The chosen frame size is half of the road width, since the road is divided into two lanes, one facing the CCTV camera and one moving away from it. Capturing images of vehicles facing the CCTV camera takes priority as the dataset, so the frame size is adjusted according to the width of the road where the vehicles are moving. The selected frame size is 1020 x 600 pixels.

Once the frame size is determined, the Region of Interest (ROI) for two-wheeled vehicles to be detected is defined. The estimated ROI size ranges from x pixels (320 to 450) and y pixels (590 to 790). These coordinates are used as the initial ROIs, which will later be used in training the YOLOv5 model to identify the ROIs for each passing two-wheeled vehicle.

Subsequently, YOLOv5 training is carried out to detect every passing vehicle. Initially, a variable is set to count the number of successfully detected two-wheeled vehicles, and its initial value is set to 0. If the ROI of a vehicle meets the initial conditions, it is recognized as a two-wheeled vehicle. Next, a check is performed to determine if the vehicle is indeed of the motorcycle type or not. If it is, the object is labeled as a motorcycle, and the count of successfully detected two-wheeled vehicles increases. This process continues until all vehicles passing through the frame have been successfully detected or the video has finished playing.

Successfully detected two-wheeled vehicles are marked with a blue rectangular border, forming what is known as the prediction box. This prediction box surrounds the objects predicted as two-wheeled vehicles, as shown in Figure 4.

Figure 4. Detected Two-wheeled Vehicles using YOLOv5

The next step is to evaluate the results of the detection process. The evaluation is conducted using accuracy calculations. Accuracy is calculated based on the values of True Positives (the number of successfully detected vehicles), True Negatives (the number of non-vehicle objects correctly not detected), False Positives (the number of non-vehicle objects incorrectly detected as vehicles), and False Negatives (the number of vehicles incorrectly not detected).

The accuracy metric is an essential measure to assess the performance of the detection model. True Positives represent the correct detections, where the model correctly identifies and labels a vehicle as a vehicle. True Negatives indicate the correct exclusions, where the model accurately recognizes and disregards non-vehicle objects as not vehicles. False Positives are instances where non-vehicle objects are falsely identified as vehicles by the model, while False Negatives occur when the model incorrectly fails to identify vehicles present in the frame.

By analyzing these metrics, the accuracy of the detection model can be determined, providing valuable insights into the model's ability to correctly detect and classify two-wheeled vehicles within the given frame. The evaluation results are essential for assessing the model's performance and identifying any areas for improvement, aiding in the refinement of the detection process.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

The accuracy value is calculated using Equation (3). The value of False Positives (FP) is 1 because the system predicted 1 image falsely as a motorcycle. The value of True Negatives (TN) is 0, as there are no vehicles other than two-wheeled motor vehicles that need to be detected.

As a result, out of a total of 102 two-wheeled motor vehicles that passed through, 96 of them were successfully detected as two-wheeled motor vehicles. Therefore, the number of True Positives (TP) is 96. There were 5 vehicles that were not successfully detected, marked by the absence of the blue box indicating the ROI frame on the objects. Thus, the number of False Negatives (FN) is 5. The confusion matrix can be seen in Figure 5. Consequently, the accuracy of the system for two-wheeled motor vehicle detection is calculated to be 94.12%.

Besides accuracy, we also need to calculate the precision and recall of the system. The precision value can be obtained using Equation (4), and the recall value can be obtained using Equation (5). Our system achieves a remarkable precision rate of 98.97% and a commendable recall rate of 95.05%, demonstrating its high accuracy and effectiveness in accurately detecting objects.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

These evaluation results indicate a high level of accuracy in the detection of two-wheeled motor vehicles, with a relatively small number of misclassifications. The system demonstrates robust performance in correctly identifying the majority of two-wheeled motor vehicles passing through the frame. The evaluation outcomes provide valuable insights into the effectiveness and reliability of the detection model, which are crucial for further improvements and practical implementations in real-world scenarios.

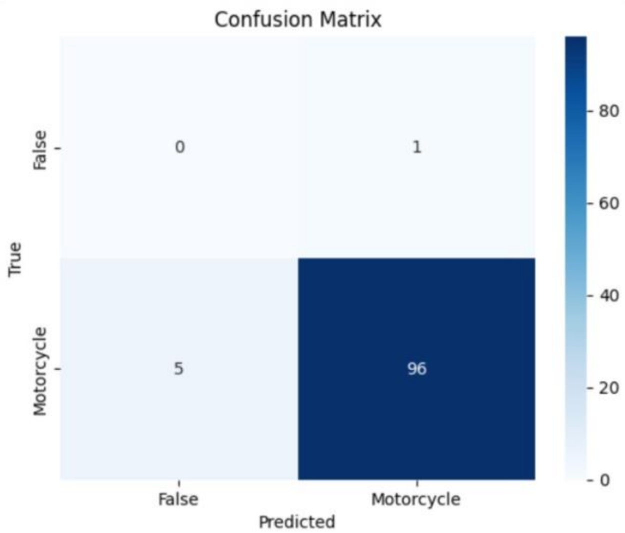

Figure 5. Confusion Matrix of the Motorcycle Detection

C. Detection of Headlight ROI using White Color Extraction (HSV)

The detection of headlights of two-wheeled vehicles using the white color extraction method (HSV) can be achieved by analyzing object characteristics such as contour size, coordinate location, and width-to-height ratio. This method aims to identify and separate white headlights from the background or other elements in vehicle images.

Firstly, the output from the previous step, which is the successfully detected images of two-wheeled vehicles, will be used as the dataset. This dataset consists of 102 images and is combined with other motorcycle image datasets, resulting in a total dataset of 263 images for this step.

Next, each vehicle will be classified according to its type, determined in this research by the brand of each vehicle. Some vehicles with larger headlights, such as scooters, are more easily detected in their headlight images compared to traditional motorcycles. After dividing them into several classes, variables for thresholding and contours will be determined. Each of these variables may vary depending on the type of vehicle.

Subsequently, a color space conversion will be performed on the images, transforming them from RGB format to HSV format. The HSV color space is chosen because it separates color, saturation, and value information more effectively. In the white color extraction method, the focus is given to the value (V) channel in the HSV image. By emphasizing the value channel, the headlight image will appear more contrasted compared to other objects in its surroundings.

The next step is to apply thresholding to the value (V) channel. Thresholding aims to separate areas with high brightness, which is a characteristic feature of headlights. An appropriate threshold value is selected to distinguish white headlights from the background or other elements in the image.

After the thresholding process, the contours of the resulting image are identified. Contour filtering is carried out based on desired object characteristics, such as contour size, coordinate location, and width-to-height ratio. For instance, headlights typically have relatively large contours, are positioned at a specific location on the two-wheeled vehicle (usually around the upper-middle part), and have a width-to-height ratio adapted to the type of motorcycle.

Once the contours meeting these criteria are obtained, the identification of headlights takes place. Contours considered as headlights are chosen based on the pre-defined object characteristics. As each type of two-wheeled vehicle has different headlight characteristics, the contour selection process varies for each vehicle type.

As with the previous method, the areas marked as ROIs will be inside prediction boxes. The prediction box used for headlight ROI detection is colored green to differentiate it from the prediction box for vehicle ROIs. The detection results can be observed in Figure 6.

Figure 6. Detected Headlights of Two-Wheeled Vehicle

Finally, evaluation and analysis are performed on the obtained detection results. The evaluation involves calculating the accuracy of the detection results. The detection results are evaluated to determine the effectiveness of the method used and identify any challenges in its implementation.

The evaluation is conducted similarly to the previous method, by calculating the accuracy using Equation (3). Out of the dataset consisting of 263 images, 194 images were successfully detected for headlight ROIs.

The confusion matrix can be seen in Table 1. The value of True Negative is zero. This is because the detected objects are headlights, and the system is designed to detect only the headlights of two-wheeled motor vehicles. There are no other objects detected as headlight, so the value of False Positive is zero. The obtained accuracy for headlight detection of two-wheeled motor vehicles is 73.76%.

Table 1. Confusion Matrix Values

	Actual Values	
Predictive Values	TP (194)	FP (0)
	FN (69)	TN (0)

The evaluation results indicate that the method achieves a moderate level of accuracy in detecting the headlights of two-wheeled motor vehicles. However, there is room for improvement in the detection performance, as indicated by the false negatives. Further analysis of the false negatives can provide insights into the challenges faced during the detection process, and potential modifications to the method can be explored to enhance accuracy.

D. Discussion

Previous research has been conducted to detect images of two-wheeled vehicles, but there has been no study specifically focusing on detecting images of vehicle headlights. Some studies that involve the detection or classification of various types of two-wheeled vehicles vehicles have already been conducted. A comparison of the accuracy achieved in this study with several previous studies is presented in Table 2. The methods employed in these studies involve various versions of the YOLO method. It is evident that YOLOv5, used in this research, outperforms the others. The proposed method differs from the approach outlined in [16] primarily in the way it determines the total number of pixels within the bounding box. In the proposed method, researchers applied a condition wherein, if the actual height divided by the actual width resulted in a value greater than 1.45, it was classified as a motorcycle.

The achieved high accuracy results can be attributed to several factors, such as the frontal angle of image capture, sufficient lighting conditions during the daytime, and the specific type of vehicles under consideration. Other potential methods that could further enhance accuracy include image segmentation and conversion to grayscale or black and white images.

The successful detection of headlight images in two-wheeled vehicles holds great promise for practical applications, such as road safety and traffic management. By accurately detecting and analyzing headlight patterns, traffic authorities can monitor and regulate vehicle movements more effectively.

Nevertheless, there are still challenges that need to be addressed in this research. For instance, the accuracy may vary under different lighting conditions, and there might be potential difficulties in detecting obscured or partially obscured headlights. These aspects should be carefully examined in future studies.

Furthermore, exploring the integration of machine learning algorithms and deep learning techniques could potentially yield improved accuracy in headlight detection. Such advancements can lead to more reliable and efficient systems for intelligent transportation and surveillance.

Table 2. Comparison With Other Method

Method	Used Model	Accuracy
Proposed Method	YOLOv5	94.12%
[5]	YOLOv3	90.8%
[7]	YOLOv1	85%
[15]	YOLOv4	90%
[16]	YOLOv5	90%

IV. CONCLUSION

In conclusion, the detection of two-wheeled motor vehicle images and the detection of vehicle headlights have been successfully accomplished. The main contribution of this research lies in utilizing the detected two-wheeled vehicle images as a dataset for headlight detection in vehicles. The results demonstrate an accuracy of 94.12% for the detection of two-wheeled motor vehicle images, and an accuracy of 73.76% for the detection of vehicle headlight images in two-wheeled motor vehicles. The recall and precision rates in motorcycle detection surpassed 90%, demonstrating the high level of accuracy and reliability achieved by the system. This indicates the system's proficiency in effectively identifying objects in images, instilling confidence in its performance and suitability for various applications.

The high accuracies obtained in both tasks confirm the effectiveness of the proposed methods in detecting two-wheeled motor vehicles and their headlights. These results are promising and pave the way for future research and practical applications in traffic management, road safety, and intelligent transportation systems.

It is worth noting that although the accuracy levels are satisfactory, there is still potential for improvement. Upcoming studies can explore additional techniques and refinements to further enhance accuracy and tackle specific challenges, such as varying lighting conditions and partial headlight occlusion.

Overall, this research provides valuable insights and establishes a foundation for further advancements in the fields of vehicle detection and headlight analysis. The successful detection of two-wheeled motor vehicles and their headlights has significant implications for road safety and traffic monitoring, ultimately contributing to the advancement of urban transportation systems.

REFERENCES

[1] D. Perhubungan R.I. Ditjen Perhubungan Darat, *Buku Petunjuk Tata Cara Bersepeda Motor Di Indonesia*. 2008.

[2] J. E. Espinosa, S. A. Velastin, and J. W. Branch, “Motorcycle detection and classification in urban Scenarios using a model based on Faster R-CNN,” in *9th International Conference on Pattern Recognition Systems*, Valparaiso: IET Digital Library, May 2018, pp. 91–96.

[3] Sutikno, H. Arif Wibawa, and R. Saputra, “Automatic Detection of Motorcycle on the Road using Digital Image Processing,” *Scientific Journal of Informatics*, vol. 6, no. 2, pp. 203–212, Nov. 2019, [Online]. Available: http://journal.unnes.ac.id/nju/index.php/sji

[4] J. H. Sri Wisna *et al.*, “Deteksi Kendaraan Secara Real Time Menggunakan Metode YOLO Berbasis Android,” *Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan*, vol. 09, no. 01, pp. 8–14, 2020.

[5] D. J. P. Manajang, S. Sompie R.U.A, and A. Jacobus, “Implementasi Framework Tensorflow Object Detection Dalam Mengklasifikasi Jenis Kendaraan Bermotor,” *Jurnal Teknik Informatika*, vol. 15, pp. 171–178, 2020.

[6] Asni B A, Amin, and Waruni K M, “Penerapan Metode Yolo Object Detection V1 Terhadap Proses Pendeteksian Jenis Kendaraan Di Parkiran,” *Jurnal Teknik Elektro UNIBA (JTE)*, vol. 6, no. 1, pp. 194–199, 2021.

[7] R. A. Utama and L. ETP, “Implementasi Metode Yolo Object Detector untuk Klasifikasi Jenis Kendaraan yang Melintas di Ruas Jalan,” *Jurnal Ilmiah Komputasi*, vol. 20, no. 4, pp. 601–608, 2021.

[8] Q. Wang, Q. Zhang, X. Liang, Y. Wang, C. Zhou, and V. I. Mikulovich, “Traffic lights detection and recognition method based on the improved yolov4 algorithm,” *Sensors*, vol. 22, no. 1, Jan. 2022, doi: 10.3390/s22010200.

[9] P. García-Sánchez, Merelo J.J, Calandria D., Pelegrina A. B., Palacio R. Morcillo F., and Garcia-Ortega R.H., “Testing the Differences of using RGB and HSV Histograms during Evolution in Evolutionary Art,” in *Conference: 5th International Joint Conference on Computational Intelligence*, 2013. [Online]. Available: http://www.processing.org/

[10] E. Prasetyo, R. Dimas Adityo, N. Suciati, and C. Fatichah, “Deteksi Wilayah Cahaya Intensitas Tinggi Citra Daun Mangga untuk Ekstraksi Fitur Warna dan Tekstur Pada Klasifikasi Jenis Pohon Mangga,” in *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI) 9*, Pekanbaru: Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau, May 2017, pp. 21–31.

[11] E. K. Kim, H. Lee, J. Y. Kim, and S. Kim, “Data Augmentation Method by Applying Color Perturbation of Inverse PSNR and Geometric Transformations for Object Recognition Based on Deep Learning,” *Applied Sciences (Switzerland)*, vol. 10, no. 11, Jun. 2020, doi: 10.3390/app10113755.

[12] R. Antika and M. Iqbal, "Using the Matlab App to Detect Objects By Color With Hsv Color Segmentation," *Jurnal Mantik*, vol. 5, no. 4, pp. 2702–2710, 2022.

[13] H. C. Kang, H. N. Han, H. C. Bae, M. G. Kim, J. Y. Son, and Y. K. Kim, "HSV Color-Space-Based Automated Object Localization for Robot Grasping without Prior Knowledge," *Applied Sciences (Switzerland)*, vol. 11, no. 16, Aug. 2021, doi: 10.3390/app11167593.

[14] D. Iskandar and Marjuki, "Classification Of Melinjo Fruit Levels Using Skin Color Detection With RGB And HSV," *Journal of Applied Engineering and Technological Science*, vol. 4, no. 1, pp. 123–130, 2022.

[15] F. Rofii, G. Priyandoko, M. I. Fanani, and A. Suraji, "Vehicle Counting Accuracy Improvement By Identity Sequences Detection Based on Yolov4 Deep Neural Networks," *TEKNIK*, vol. 42, no. 2, pp. 169–177, Aug. 2021, doi: 10.14710/teknik.v42i2.37019.

[16] D. I. Mulyana and M. A. Rofik, "Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOV5," *Jurnal Pendidikan Tambusai*, vol. 6, no. 3, pp. 13972–13982, 2022.

[17] R. B. S, A. Marium, G. N. Srinivasan, and S. A. Shetty, "Literature Survey on Object Detection using YOLO," *International Research Journal of Engineering and Technology*, vol. 7, no. 6, pp. 3082–3088, 2020, [Online]. Available: www.irjet.net

[18] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed Tools Appl*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023, doi: 10.1007/s11042-022-13644-y.

[19] A. Mashudi, F. Rofii, and M. Mukhsim, "Sistem Kamera Cerdas Untuk Deteksi Pelanggaran Marka Jalan," *JASEE Journal of Application and Science on Electrical Engineering*, vol. 1, no. 1, pp. 15–25, Feb. 2020, doi: 10.31328/jasee.v1i01.4.

[20] M. Iqbal, Karmilasari, Y. Karyanti, and dan Yulia Chalri, "Rancang Bangun Sistem Deteksi Pelanggaran Lalu Lintas dalam Simulator Mengemudi menggunakan Unity Game Engine," *Jurnal Format*, vol. 1, pp. 38–49, 2021.

[21] R. H. S., A. Rabi', and E. Sonalitha, "Sistem Pendeteksi Pelanggar Traffic Light pada Zebra Cross Menggunakan Raspberry Pi Berbasis Pengolahan Citra Digital," *Journal of Electrical Electronic Control and Automotive Engineering (JEECAE)*, vol. 5, no. 1, May 2020.

[22] Suhartono, S. G. Zain, and Sugiawan, "Sistem Object Recognition Plat Nomor Kendaraan Untuk Sistem Parkir Bandara," *Journal of Embedded System Security and Intelligent System*, vol. 3, no. 2, pp. 127–134, Nov. 2022, [Online]. Available: https://ojs.unm.ac.id/JESSI/index

[23]