

Nighttime Motorcycle Detection for Sparse Traffic Images Using Machine Learning

Pov Vandeth¹, Jimmy Tirtawangsa², and Hertog Nugroho^{3*}

¹Digital Project Management at Hattha Bank
Phnom Penh, Cambodia, 120606

²School of Computing, Telkom University
Bandung 40257, Indonesia

³Electric Engineering Department, Bandung State of Polytechnic
Bandung 40559, Indonesia

Email: ¹pov.vandeth@gmail.com, ²jimmytirtawangsa@telkomuniversity.ac.id, ³hertog@polban.ac.id

Abstract—Traffic accidents often occur at night. It is understandable, since at night, people have low visibility. Many efforts to develop tools to detect nearby vehicles to avoid crashes have been reported. However, most of them worked only on detecting cars. The research aims to detect motorcycles at night, to complement the previous studies, which mainly focused on cars. The research introduces four features which are extracted from the red pixel and edge map. The algorithm to extract the features has also been developed. They are applied to three commonly used classifiers: Artificial Neural Network (ANN), Decision Tree, and Support Vector Machine (SVM) classifiers to validate the effectiveness of the features. Since the public dataset related to the research is not available yet, the nighttime videos from YouTube have been collected. The datasets contain all the various levels of darkness. They are divided into an 80-20 ratio for training and testing sets to support the experiment and measure the validity of the proposed method. As the best result, the detection using ANN can detect motorcycle proposals with accuracy of 72.71%, precision of 65.10% and recall of 73.33%. Furthermore, during the experiment, the classification can perform consistently in 0.04 seconds per image. Therefore, the method is suitable for use in a real-time system.

Index Terms—Nighttime Motorcycle Detection, Sparse Traffic Images, Machine Learning

I. INTRODUCTION

INTELLIGENT Transportation Systems (ITS) are a collection of advanced application services for various modes of traffic and transport. It is used in many smart city infrastructures to promote self-driving and driving assistance and solve many other traffic issues. One main concern to improve the performance of ITS is the ability to collect drivers' behavior leading to

Received: April 12, 2022; received in revised form: July 26, 2022; accepted: July 28, 2022; available online: March 17, 2023.

*Corresponding Author

TABLE I
DATA OF TRAFFIC ACCIDENTS BASED ON THE VEHICLES' TYPE
IN THE DKI JAKARTA IN 2021.

No	Vehicle's type	Count	Percentage (%)
1	Public vehicle (taxi, bus, and others)	138	2
2	Small-size bus	741	12
3	Truck and pick-up	485	8
4	Car	128	2
5	Motorcycle	4,507	74
6	Bicycle	96	2
Total		6,095	

traffic accidents. Road injuries and the death tolls due to traffic accidents have terrible impacts on individuals, communities, and countries [1].

In ASEAN countries, motorcycles are a normal mode of transportation and are subject to road crashes more than other modes of transportation. According to a recent report on road safety [2], 74% of deaths are from 2-wheel riders (see Table I). It is the highest rate of traffic accidents in Indonesia. One crucial cause is the nighttime conditions. Nighttime condition is prone to traffic accidents. Therefore, nighttime vehicle detection capability is essential in ITS.

The vehicle's features are easily recognized in the daytime, such as color, shape, texture, and others. It is also easily differentiated from the background scene. However, such features cannot be easily identified at night, as the images taken from the night scene have low contrast, brightness, and signal-to-noise ratio. Meanwhile, vehicle lights, including turn lights, can be used at night to identify the vehicle since the lights are turned on. Hence, the research concentrates on detecting motorcycles at night to complement the previous studies, which have mainly focused on cars.

II. LITERATURE REVIEW

Previous study proposes a method to provide a set of proposals for nighttime images. It introduces the Bayes Saliency-based generator to propose a vehicle object [3]. Three features were extracted from an image: local contrast, taillight, and luminance to serve for saliency detection. Edge detection through prior estimation is used to separate the background and salient objects. The prior estimation also determines the threshold for feature maps and the Bayes rule. Moreover, the taillights are detected by luminance contrast saliency using information from the LAB color system, and procedures to analyze and verify the pair of taillights are developed [4].

Meanwhile in [5], the Harr-like-based classifier detected vehicle candidates from dark and bright environments using light pair identification to check whether they belong to a vehicle. It proposes Region of Interests (ROIs) of a pair of turn lamps. In the Hue, Saturation, Value (HSV) color system, the lamps' luminance (H channel) was found to be invariant in the frequency domain and has significant energy within the 0–3 Hz range. Therefore, the luminance channel was transformed into a frequency domain using the Discrete-time Fourier transform (DFT), and Adaboost classifier trains its magnitude distribution to detect the turn lamps.

Moreover, a method to detect and track two-way vehicles with occlusion is proposed [6]. The headlights of oncoming vehicles and the taillights of preceding ones are classified by a multiclass AdaBoost classifier which is pre-trained offline using Haar features extracted from gray images and a color feature extracted from LAB color space. The procedures to classify and track paired lights are also developed to handle occlusion.

In [7], an algorithm is proposed to detect the preceding vehicle taillight using a linear Support Vector Machine (SVM). For the training procedure, it divides the images into standard-size non-overlapping blocks and extracted features, such as Histogram of Gradient (HoG), Linear Binary Pattern (LBP), and Four Direction Feature (FDF). Then, it constructs feature vectors through tensor construction and decomposition and ranks the features. Several features with the highest ranks are selected to be stored as objects, while the rest are stored as non-objects. The candidate proposal generators (bounding boxes) using EdgeBoxes, local contrast, and image region similarity are developed for the testing procedure. HoG, LBP, and FDF features are extracted from the selected boxes to be fed to the linear SVM classifier.

Meanwhile, a different approach is also proposed [8]. Instead of focusing on one vehicle, the

research collects videos from cameras installed on nearby vehicles under an integrated ITS and develops a context-based vehicle detection and tracking based on Kalman Filter, which can help predict missing vehicle taillights. In the detection step, The Converted Red Green Blue (CRGB) color space is proposed. It extracts maximally stable extremal regions from the CRGB map to get candidate proposals. Then, the Kalman filter is implemented in the tracking step to match and predict taillight locations in consecutive frames. It also develops a procedure to fine-tune tracking results by leveraging information taken from other vehicles.

Another research designs a real-time vision-based blind spot warning system to detect cars and motorcycles [9]. In the daytime, it uses the optical flow of video and the intersection of vertical and horizontal edges to propose the ROI and extracted Speeded-Up Robust Features (SURF) for the linear SVM classification. For the nighttime environment, it detects the vehicle headlight through image color. With rule-based algorithms, the image is converted to a binary one to find the contour of the vehicle headlight. Pairing the light is done to group the contour for cars and motorcycles before analyzing the headlight's symmetrical position. All regions of interest are converted to 3D objects to measure their distance. Finally, the algorithm decides the object classes (car and then motorcycle), and it notifies the position and the type of the vehicle to the driver.

Recently, with the increasing popularity of deep-learning tools, the concepts have been adopted to address the mentioned problems. For example, YOLO 3 has been adopted as a backbone machine to detect vehicles at night [10]. The objects to be detected are the whole body of the car. The learning and recognition processes are entirely entrusted to the machine. The input images (in training and the validation) are enhanced by the optimal Multi-Scale Retinex (MSR) algorithm to increase the recognition capability. With a similar scheme, another research proposes an M-YOLO network built by a combination of MobileNet V.2 as the backbone network for feature extraction and multi-scale prediction head of YOLO V.3 as the detection part [11]. Meanwhile, a modified network is proposed to detect nighttime vehicles based on CornerNet [12]. A special pooling layer called "Direction Attention Pooling" generates candidate proposals and fine-tunes them by the Bayes corner localization algorithm derived from the fusion of the Nakagami image and Hue, Saturation, Intensity (HSI) segmentation.

In addition, previous research concentrates on detecting objects under insufficient and nighttime illumination conditions [13]. It proposes data collection procedures, covering occluded objects, blurry objects,

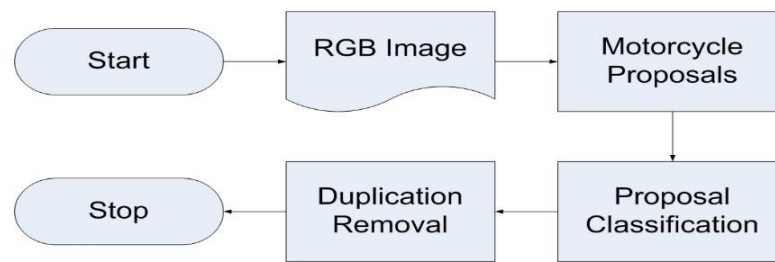


Fig. 1. The proposed design of motorcycle detection.

small objects, data labeling, and transfer learning techniques. It adopts pre-trained VGG16 and ResNet101 as feature extraction models and Faster Region-based Convolutional Neural Networks (R-CNN) for the detector. Similarly, using vehicle-reflected lights on the body is also proposed as additional visual clues to accompany the traditional vehicle lamp lights for vehicle nighttime detection [14]. The combination of the lights is called ‘vehicle highlights’. It develops a vehicle detection procedure based on the vehicle highlight. At first, it proposes vehicle highlight detection using a novel iterative label assignment, generating vehicle highlight masks and their labels. The results are combined with corresponding original images to train the network. In the experiment, the scheme is validated on several well-known networks, such as MSCNN, Faster R-CNN, YOLOv2, SiNet, and MobileNet. Another different approach is discussed [15]. The idea is based on the argumentation that most datasets containing vehicles are recorded during the daytime, while nighttime vehicle datasets are scarce. Therefore, it proposes AugGAN, a structure-aware unpaired image-to-image translation network. The network is used to transfer labeled daytime vehicle datasets to the nighttime domain while retaining labeled image objects and structural information of original images. The detector is trained by translated or actual nighttime vehicle datasets.

The methods in previous studies [3–6, 8, 9] provide the fundamental concept for generating the ROIs through pairing the vehicle taillights at the primary step. However, they do not apply it to fully solve the motorcycle detection problem since motorcycles have a single taillight and are smaller than cars. Meanwhile, the other studies [10–15] can be applied to many objects. There are no specific efforts to exploit vehicles’ characteristics at night. The objects to be trained are covered by bounding boxes and labeled, while during the detection process, the boxes were evaluated. Only in one previous research [9], motorcycle detection is considered.

III. RESEARCH METHOD

The proposed method is to detect motorcycles at night, as depicted in Fig. 1. Initially, a set of ROI proposals are generated from an HSV image. From each proposal, four features are extracted to be fed to a classifier. After the proposals have been identified, their positions are analyzed for duplication. The contributions are found in the feature extraction procedures. The research exploits the potential light from the taillight of the motorcycle. The red channel and the edges provide the vital attributes to build four features (red map, edge map, edge ROI, and object contour). The features form a feature vector for the classification.

A. Motorcycle Proposal

The motorcycle proposal is the introductory module to work directly on the raw image to expose the ROIs. Two essential data related to the ROI of a motorcycle are the taillight area and the reflection of a motorcycle. A taillight’s radiant is a vital hint to let humans and robots alike realize there is a vehicle. It also helps drivers to identify whether the vehicle is a car, a motorcycle, or something else. The taillight contributes significantly to this proposal process. The reflection of the motorcycle adds more weight to the verification process. The taillight, standardized by the vehicle manufacturer, plays a significant role in generating the proposal region. Starting from a raw image, as shown in Fig. 2, the RGB image is converted to the HSV color system, and H and V channels are filtered. Sample data shows that the H threshold is in two ranges: $(0 \leq H \leq 50)$ and $(150 \leq H \leq 179)$. The intensity value is also essential to verify the taillight. It is in the V channel of HSV. The threshold is set to $(200 \leq V \leq 255)$ because the taillight is also the bright spot or the dominant light. Each pixel is

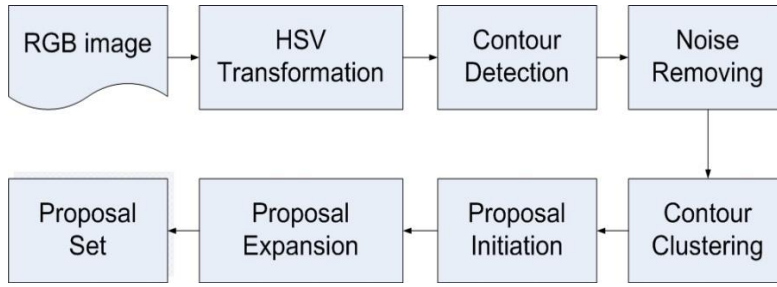


Fig. 2. The proposed design of the proposal generator.

evaluated according to Eq. (1).

$$f(x, y) = \begin{cases} 1 & [(0 \leq H(x, y) \leq 50) \\ & \vee \\ & (150 \leq H(x, y) \leq 179)] \\ & \wedge \\ & (200 \leq V(x, y) \leq 255) \\ 0 & \text{else} \end{cases} \quad (1)$$

Function $f(x, y)$ checks whether the pixel belongs to a bright spot at the x and y positions in the HSV image. It shows $H(x, y)$ and $V(x, y)$ as the hue and the value channels for each pixel. Afterward, the research applies the method [16] to find all possible contours within a binary output. All very short contours containing less than three connected points are discarded. Besides taillights, there are also some reflections and turn signals around them. Some parts of the rider's body and the lower part of a motorcycle may also be visible. Therefore, contours close to one another are clustered as one set. Suppose C is a set of contours. Each contour c_i is enclosed by a circle with a center point (x_i, y_i) and radius (r_i) . Two contours c_i and c_j will be joined if they meet the criteria defined by Eq. (2). It shows (x_i, y_i) and r_i as a center point and radius of contour c_i , (x_j, y_j) and r_j as a center point and radius of contour c_j , and m as the minimum rate of missing distance compared with the Euclidean distance between the contour c_i and c_j . From observation, its optimal value is 0.04.

$$(r_i + r_j)^2 \geq (1 - m)^2 \times ((x_i - x_j)^2 + (y_i - y_j)^2). \quad (2)$$

After the contours of taillights has been clustered, the largest local area is set and all contours inside the area are clustered to become a set of points. Those points are approximated by the convex hull. The research uses the algorithm from previous research [17] to work for the convex hull. Each convex hull is further simplified as a rectangle and known as the preliminary proposal. Each rectangle defines the final proposal.

B. Feature Extraction

Dark or blurred images at nighttime are not easy to perceive. It also poses a low-intensity gradient. Thus, the possible information used as features from a nighttime image is usually much less than from a daytime image. However, some features are found to be useful to identify the motorcycle through the motorcycle's taillight and its reflection. Therefore, four features to represent a motorcycle are proposed, as shown in Fig. 3. They are the red map, edge map, edge ROI, and object contour.

1) *Red Map*: The area around the taillight typically emits red, and some parts of the motorcycle will also reflect the red light. Under normal road conditions, the motorcycle images should emit more red pixels than images with no motorcycles. As typical images from digital cameras are in the RGB color system, red pixels are higher than the other two colors. Eq. (3) is used for identifying the red map. It shows $RM(x, y)$ as the mapping function to decide whether it is a red pixel at the x and y coordinates and $Red(x, y)$, $Blue(x, y)$, and $Green(x, y)$ as the color mapping functions to get its color value at x and y coordinates.

$$RM(x, y) = \begin{cases} 1 & \begin{matrix} Red(x, y) > Blue(x, y) \\ \wedge \\ Red(x, y) > Green(x, y) \end{matrix} \\ 0 & \text{else} \end{cases} \quad (3)$$

2) *Edge Map*: Although the nighttime environment provokes edges to become unclear because of low contrast and brightness, the edges are still notable in a nighttime object [2]. For example, the road lamp may be confused with a taillight, but on a motorcycle, there are other objects around the taillight. Below the taillight, a license plate area, tires, turn lamps, and other motorcycle parts can be found. Above the taillight, it has the rider's body and helmet. All these clues provide numerous unique edges compared with pixels surrounding the road lamp or other noise. So, the actual object of the motorcycle should have more edges than the noise in the region of interest. In the research,

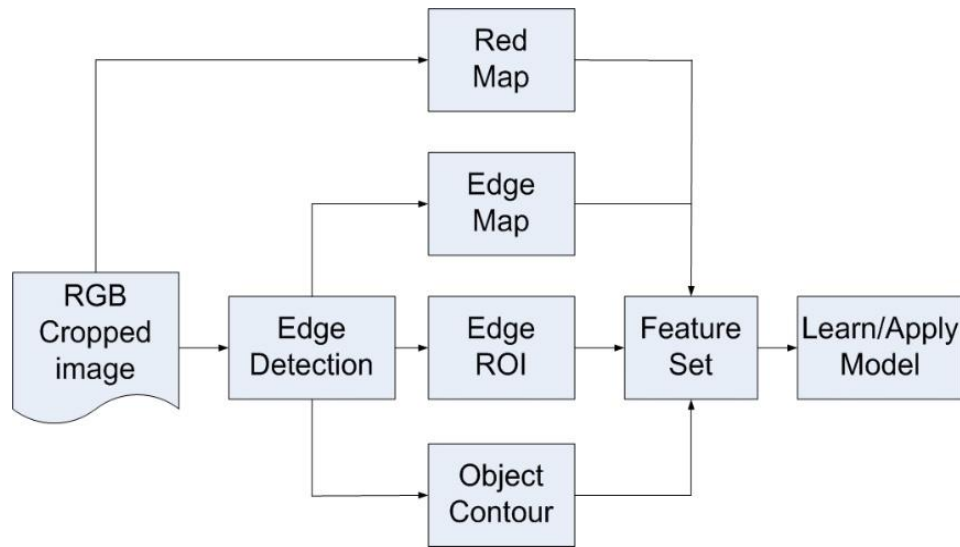


Fig. 3. The proposed new features for motorcycle classification.

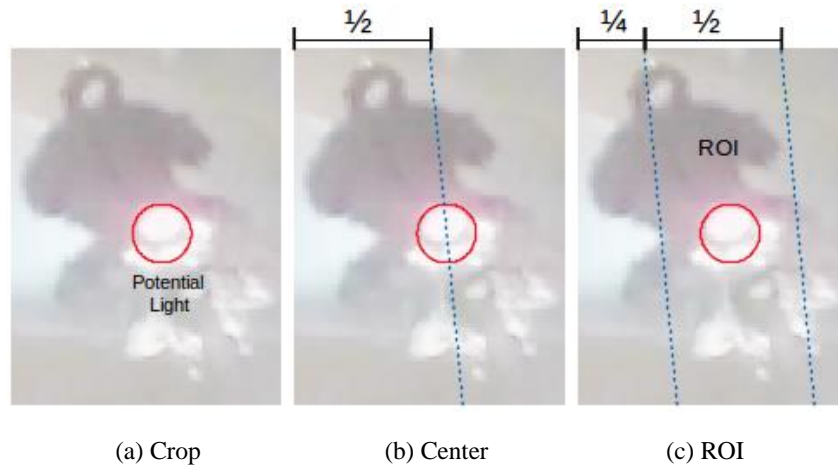


Fig. 4. The ROI of the motorcycle.

the Sobel operator is used to compute the edge map of the input image [18]. The criteria to determine an edge are set by Eq. (4). The $EM(x, y)$ is denoted as a mapping function to check the edge at position x and y axes. It also has $E_{sobel}(x, y)$ as an intensity value at x and y axes from the edge map. It finds that $th = 60$ should be sufficient as the lower threshold value.

$$EM(x, y) = \begin{cases} 1 & E_{sobel}(x, y) \geq th \\ 0 & else \end{cases}. \quad (4)$$

3) *Edge ROI*: The position of a motorcycle does not always appear in the center of the ROI. Sometimes, it is located at a bit left or right-hand side of the camera. The centerline should be rotated to guarantee that the object, including the rider and the bottom part of the

motorcycle, is in the center of ROI. It passes through a center of a taillight and the mid-width of the bounding box, as shown in Fig. 4b. The adaptive orientation of ROI makes it flexible to accommodate the motorcycle's position. As shown in Fig. 4c, the centerline is used to cover 1/4 of the width of the bounding box to the left and right to get the refined ROI. It means it takes up almost half the area of the bounding box.

The refined ROI covers all pixels within a 25% width of the bounding box. It is highly expected to contain the actual shape of a rider and the backside of a motorcycle. Therefore, all edges in the refined ROI will be counted. The assignment of a pixel value within the ROI is expressed in Eq. (5). The $ER(x, y)$ is a mapping function to check the edge value at

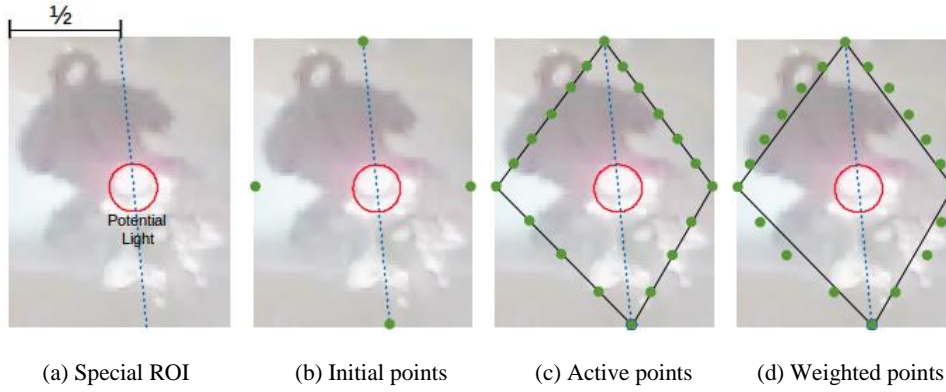


Fig. 5. Object contour of the motorcycle.

position (x, y) . Then, $E_{sobel}(x, y)$ is an intensity value at position (x, y) in the edge map. It is the edge pixel whenever its intensity is greater than the threshold.

$$ER(x, y) = \begin{cases} 1 & E_{sobel}(x, y), (x, y) \in ROI \\ 0 & else \end{cases} \quad (5)$$

The features described previously are the number of pixels in the image against the criteria postulated by $f(x, y)$ and normalized by Eq. (6). It shows W and H as the width and height of the image. Then, $f(x, y)$ represents a criteria pixel of a feature for the motorcycle.

$$\text{Feature}(\text{Image}) = \frac{\left[\sum_{x=0}^{W-1} \sum_{y=0}^{H-1} f(x, y) \right]}{W \times H} \quad (6)$$

4) *Object Contour*: Looking at Fig. 4, it seems impossible to obtain the actual shape of an object in the nighttime environment. However, there are more edges around and within the motorcycle. The outer edges can be exploited to form a closed polygon enveloping the object, which, in this case, is a motorcycle and its rider. Therefore, a method to build a closed object contour is developed.

The object contour feature is an area of a polygon representing the contour of the object. Assuming the polygon has the same amount of corner points around the object, the problem is to determine the positions of the points representing the object's outer shape.

Given a bounding box, the position of the points is defined with the following procedures. First, the center line is drawn from the top center of the bounding box through the center point of the potential light and extended downward until it touches the bottom border of the bounding box, as shown in Fig. 5a. Then, a horizontal line is drawn from the center point of potential light to both sides until it touches the borders of its bounding box. Four points will form the corner of a polygon, as shown in Fig. 5b. The horizontal line

divides the polygon area into upper and lower sides. Then, the lines connecting the four points are drawn, as shown in Fig. 5c. On each upper side, five new points are generated along the lines and distributed evenly. Since the position of the top point is in the center of the bounding box, the points will have the same distances as their neighbors.

Second, on each lower side, three points are generated along the lines and distributed evenly. However, since the position of the bottom point is not in the center, the distance between the two closest points on the lower left may not be the same as those on the lower right. Overall, from top to bottom of ROI, eight new points will be generated on each side (excluding the four points to make the polygon), as shown in Fig. 5c. This set of points is the initial position to search for object contour.

Third, the weights on the x-direction on some important points are adjusted to fit with the ellipse shape. For five points at the top left, an x-interval in the positive direction is added to the third point, half for the second and fourth, and one-third of the x-interval to the first and fifth points. Similar but in the opposite direction, x-intervals are added to the top right. Three points at the bottom right side are added by x-interval for the second point and half for the first and third points. Again similar but in the opposite direction for the bottom left side (see Fig. 5d), the x-intervals are determined by different areas of the points, as described by Eqs. (7)–(9).

In Eq. (7), it shows $x_{top\ interval}$ as the x-interval for points at the top side which the gap size is between 11 points spread uniformly along the image width. Then, $W.x_{bottomleft}$ (Eq. (8)) and $x_{bottomright}$ (Eq. (9)) are intervals for points at the bottom left and bottom right. They are normalized by four to fit three points on each side. Next, $bottom(x)$ is the distance between leftcorner point and crosspoint between the bottom line

Algorithm 1 Feature extraction algorithm by using active contour.

```

1: procedure ACTIVECONTOUR(CropImg, CircleTaillight)
2:  $\blacktriangleright$  X is a rectangle set and a prediction set of the motorcycle proposals. (See Fig. 5a)
3:  $\Delta x \leftarrow \text{Width}(\text{CropImg})/2$ 
4:  $\Delta y \leftarrow \text{CircleTaillight.getY}()/6$   $\blacktriangleright$  For 7 active points (See Fig. 5a and b)
5:  $P_{0,x} \leftarrow \text{Width}(\text{CropImg})/2$ 
6:  $P_{0,y} \leftarrow 0$ 
7:  $i \leftarrow 1$ 
8: while  $i < 7$  do  $\blacktriangleright$  Initiate the active points of top right & left (See Fig. 5c and d)
9:    $P_i \leftarrow P_{i-1} + \text{Point}(\Delta x, \Delta y)$ 
10:   $P_{20-i} \leftarrow P_i - \text{Point}(2 * i * \Delta x, 0)$ 
11:   $i \leftarrow i + 1$ 
12:  $\text{BottomX} \leftarrow \text{Colinear}(P_{0,x}, \text{CircleTaillight.x}, \text{Height}(\text{CropImg}))$ 
13:  $\Delta x \leftarrow (P_{7-x} - \text{BottomX})/4$   $\blacktriangleright$  For 5 active points
14:  $\Delta y \leftarrow (\text{Height}(\text{CropImg}) - P_{1-y})/4$ 
15:  $P_{10} \leftarrow \text{Point}(\text{BottomX}, \text{Height}(\text{CropImg}))$ 
16: while  $i < 10$  do  $\blacktriangleright$  Initiate the active points of bottom right & left
17:    $P_i \leftarrow P_{i-1} + \text{Point}(-\Delta x, \Delta y)$ 
18:    $P_{n+1} \leftarrow P_{n+3} - \text{Point}(\text{BottomX}/4, \Delta y)$ 
19:  $\text{EclipseWeightAdjustment}(P_0, P_1, \dots, P_{19})$ 
20:  $\text{Status} \leftarrow \text{StartProgressiveIdent}(P_0, \dots, P_{19}, \text{CircleTaillight})$ 
21: if  $\text{Status} = \text{"No obstacle"}$  then
22:    $\text{Feature} \leftarrow \text{Area}(\text{CircleTaillight}) / \text{Area}(\text{CropImg})$ 
23: else
24:    $\text{Contour} \leftarrow \text{ApproximatePolygonCurve}(P_0, P_1, \dots, P_{19})$ 
25:    $\text{Feature} \leftarrow \text{Area}(\text{Contour}) / \text{Area}(\text{CropImg})$ 

```

Algorithm 2 Duplication Removal Algorithm.

```

1: procedure Removal(X, Class)
2:  $\blacktriangleright$  X is a rectangle set and a prediction set of the motorcycle proposals.
3:  $n \leftarrow \text{Size}(X)$ 
4:  $M \leftarrow \text{Matrix}[n \times n]$ 
5:  $M_{i,j} \leftarrow \text{Equation (10)}$ 
6:  $Y_i \leftarrow \text{Equation (11)}$ 
7: if  $Y_i = 0$  then
8:    $\text{Image} \leftarrow \text{ExportProposal}(X_i, \text{Class})$ 
9:    $\text{RemoveRowColumn}(M, \text{Class}, X, i)$ 
10:   $n' \leftarrow \text{Size}(X)$ 
11:  if  $n' > 2$  then
12:     $\text{maxRetry} \leftarrow n' / 2$ 
13:    while  $\text{maxRetry} > 0$  do
14:       $Y_i \leftarrow \text{Equation (11)}$ 
15:       $t \leftarrow \text{FindMax}(Y)$ 
16:      if  $\text{Class}_t \neq 1$  then
17:         $\text{RemoveRowColumn}(M, \text{Class}, X, t)$ 
18:      else
19:         $t' \leftarrow \text{FindMaxNeighbor}(M, Y, t)$ 
20:        if  $\text{Class}_{t'} \neq 1$  OR  $M_{t,t'} > M_{t,t}$  then
21:           $\text{Image} \leftarrow \text{ExportProposal}(X_t, \text{Class})$ 
22:           $\text{RemoveRowColumn}(M, \text{Class}, X, t')$ 
23:        else
24:           $\text{RemoveRowColumn}(M, \text{Class}, X, t)$ 
25:         $\text{maxRetry} \leftarrow \text{maxRetry} - 1$ 
26:       $\text{Image} \leftarrow \text{ExportAllProposal}(X, \text{Class})$   $\blacktriangleright$  To export all the remaining proposals

```

and centerline (see Fig. 5b–d).

$$x_{\text{topinterval}} = \frac{W}{12}, \quad (7)$$

$$x_{\text{bottomleft}} = \frac{\text{bottom}(x)}{4}, \quad (8)$$

$$x_{\text{bottomright}} = \frac{(W - \text{bottom}(x))}{4}. \quad (9)$$

Fourth, those points are iteratively shifted horizontally, one pixel at a time to the centerline of ROI except for the top and bottom points, starting from their initial positions. Each point will stop if it finds any edge or centerline (which, in this case, it finds no edge). After all points have been run, the research applies

the Douglas Peucker algorithm [19] to approximate the contour and compute its area using Green's theorem [20]. Thus, the feature is produced in the range [0, 1] as the result of normalizing the contour area against the bounding box size. By using object contour, the object's whole shape, including the taillight, the rider, and the backside of a motorcycle, can be found. Algorithm 1 describes the procedure.

C. Duplication Removal

Since noises have been removed by some steps beforehand in the proposal-generating process, there is a possibility that several bounding boxes covered one object of a motorcycle, as shown in Fig. 6a. Logically, only the best one is needed. Therefore, an adaptable solution has been developed as shown in Algorithm 2. It has two backbone formulas to solve this problem. First, Eq. (10) is applied to measure the overlap ratio of each proposal. Second, Eq. (11) sums up the result of the overlapping ratio between each proposal against the proposal set. It has i and j as the indexes of the motorcycle proposal set and x_i as a bounding box of the proposal at the i^{th} index.

$$f_{i,j}(x) = \begin{cases} \frac{|x_i \cap x_j|}{x_i} & i \neq j; 0 \leq i, j \leq n-1 \\ 0 & \text{else} \end{cases}, \quad (10)$$

$$f_i(x) = \sum_{j=0}^{n-1} x_{i,j}. \quad (11)$$

IV. RESULTS AND DISCUSSION

The experiments are performed on the operating system of Ubuntu 16 LTS 64-bits, OpenCV library, and C++ programming language. This software runs on Lenovo ThinkPad-T430s with Intel Core i7-3520M (2.90GHz x 4 CPU) processors, Intel Ivy bridge Mobile graphic module, and 8GB memory.

Nighttime traveling consists of multiple situations from a city to the outskirts. Moreover, the brightness level ranges from brightest to dark in many areas. Therefore, the dataset must contain all the various conditions to support and measure the system experiment. The nighttime videos are collected from YouTube as listed in Table II, accessed on July 6, 2019, 15:15 (GMT+7) and taken by a nighttime rider from one city to another in Java Island, Indonesia. There are 1,111 images (1280×720 in width and height) extracted from frames of the 10 videos about nighttime traffic. The videos cover various levels of darkness. The ground truths are manually created and labeled. There are 3,337 detected objects, both motorcycles (class=1) and non-motorcycles (class=0). The dataset is divided into an 80-20 ratio for training and testing sets for experiment purposes.



Fig. 6. Before and after applying the duplication-removing algorithms.

TABLE II
LIST OF VIDEO SOURCES FOR THE DATA COLLECTION.

No	YouTube Channel	Video URL	Published Date	Range in Image Set
1	Aljabar Rider	https://youtu.be/V4HpZc3ftTc	January 1, 2018	1–17
2	Mega Pro Tapai	https://youtu.be/w5uZZICMJu4	September 9, 2017	18–54
3	2WHEELZ LUNATIC	https://youtu.be/ty35_ZrlknU	October 28, 2017	55–59
4	Guntur Wibowo	https://youtu.be/EjuBqZdyN4o	January 15, 2018	60–74
5	Guntur Wibowo	https://youtu.be/6JjAxQMvoiE	August 29, 2017	94–112
6	Gainur Nugraha	https://youtu.be/M4cuXpYE4Yg	July 28, 2015	113–214
7	Doyan Motor	https://youtu.be/ofx1D4tdR0U	March 27, 2016	215–467
8	Noval Rachmat	https://youtu.be/SDoD9LpmVdQ	December 5, 2013	468–782
9	Noval Rachmat	https://youtu.be/ajlohOLZtwQ	December 5, 2013	783–1,008
10	Muhammad Ilyas Nugraha Natadiwidjaja	https://youtu.be/iZ0le0aEG24	July 23, 2017	1,009–1,111

A. Survey of Classification Algorithms

Experiments have been conducted in various environments to see the performance of the proposed features. Extracted features become the training and testing vectors for some classification devices such as Artificial Neural Network (ANN) [21], Decision Tree (DTree) [22], and SVM [23]. Neural Network and Decision Tree were efficient learning for non-linear decision classifiers in the 1980s [24]. For the case of ANN, the research selects the following configurations: 1 input layer with 4 neurons, 2 hidden layers with 16 and 8 neurons, and 1 output layer with 2 neurons (1 = motorcycle, 0 = non-motorcycle). Table III shows the result of 15 experimental scenarios. Most of the result has performance accuracy above 50%.

Using the ANN classifier, six of them have accuracy above 80%. One comes from the combinations of only red pixel and object contour features. Meanwhile, others are from the combinations of three and four features. The highest accuracy is presented by full combinations of the proposed features which are the red pixel, the edge pixel, the edge ROI, and the object contour.

Table IV shows the application of PCA [25] on several combinations of the proposed features. It shows that all of them produce results above 80% accuracy after 15 test iterations using the ANN classifier. DTree

improves the results, although they are still below 80% and achieve similar highest results compared to Table III. Therefore, the results in Tables III and IV on the classification accuracy have shown that the proposed features are effective for machine learning in motorcycle detection during nighttime. The object contour is found to be the best feature of observing the experiment results. It always contributes to high-accuracy detection. The less effective feature is edge ROI, contributing to lower detection accuracy.

Most experiment results above 80% accuracy come from the combinations of three or four features. Most of the high-accuracy results are provided by ANN, while the Decision Tree and SVM cannot achieve any results higher than 80% accuracy as shown in Tables III and IV. The lowest accuracy rate from ANN is 80.64% in Table III. Meanwhile, the maximum accuracy is 78.35% for the Decision Tree and 62.34% for SVM. The kernel-based method like SVM does not present good performance for the features even with different kernels, such as polynomial, Radial basis, Sigmoid, exponential Chi^2 , and Histogram intersection kernels. According to all high accuracy results from those tables, the combinations of four features (red pixel, edge pixel, edge ROI, and object contour) have the best accuracy under the classification of ANN.

TABLE III
PERFORMANCE ACCURACY OF MACHINE LEARNING ON THE PROPOSED FEATURES.

No	RP	EP	ER	OC	ANN (%)	DTree (%)	SVM (%)
1	✓				64.48	53.20	62.04
2		✓			69.51	47.56	37.50
3			✓		59.14	33.53	51.37
4				✓	78.81	58.23	78.81
5	✓	✓			72.56	67.07	69.51
6	✓		✓		64.93	63.71	62.04
7	✓			✓	82.16	75.91	77.13
8		✓	✓		77.89	71.79	65.09
9		✓		✓	79.57	71.95	23.78
10			✓	✓	79.57	68.14	78.96
11	✓	✓	✓		80.64	73.32	62.34
12	✓	✓		✓	81.09	76.98	60.82
13	✓		✓	✓	82.92	77.43	40.39
14		✓	✓	✓	82.77	74.08	22.56
15	✓	✓	✓	✓	84.14	76.82	35.21

Note: Red Map (RM), Edge Map (EM), Edge ROI (ER), and Object Contour (OC).

TABLE IV
PERFORMANCE ACCURACY OF THE CLASSIFIERS ON THE PROPOSED FEATURES WITH PCA.

No	RP	EP	ER	OC	PCA	ANN (%)	DTree (%)	SVM (%)
1	✓	✓	✓		✓	80.64	74.69	62.34
2	✓	✓		✓	✓	81.40	77.28	61.28
3	✓		✓	✓	✓	82.92	75.76	40.49
4		✓	✓	✓	✓	82.16	73.17	22.56
5	✓	✓	✓	✓	✓	83.84	78.35	35.21

Note: Red Map (RM), Edge Map (EM), Edge ROI (ER), and Object Contour (OC).

B. Training Process

Next, the research compares the proposed design, including the ANN for classification, against other deep learning methods, such as the VGG Net, on the dataset. The VGG Net is one of the popular deep learning frameworks and was proposed by a group of Oxford researchers in 2014 for image classification and detection [26]. Among several versions, the VGG-16 with 5 convolutional layers and 2 connected layers is used in this work. There are 1,597 samples of positive and 1,084 samples of negative data in the training set. From the training set, the extraction process produces a feature set represented by a set of multiple vectors. The learning algorithm builds its model from those vectors and the provided classes. Initially, it is pre-trained to generalize the weight of the network. After that, the last layer of the pre-trained network is fine-tuned by reducing the learning rate using the Stochastic Gradient Descent (SGD) optimizer. Fine-tuning takes a longer time than pre-training. Finally, the last weighted model is used to predict the class of an object.

C. Result of the Nighttime Motorcycle Detection

From the dataset, 20% of it are randomly selected as the testing data, which equals to 218 raw images. Also, 388 samples of the actual motorcycles are manually labelled based on the human perspective. From the

218 images of testing data, the algorithm finds 689 ROIs based on the potential bright spots. Some of those lights are reflected by non-motorcycle objects, such as banners, traffic lights, road lamps, and others. To expose the ROIs, the research groups the bright pixels in the raw images in the solution design. Hence, from the 218 images, the algorithm proposes 689 ROIs. However, there are only 388 motorcycle objects. It means that there are many overlapped proposals. Therefore, bright images may cause more ROI creations. A more aggressive algorithm to remove the duplicated ROIs is needed to refine the proposals.

The system tracks all the 689 ROIs against the 388 labels of the motorcycles to construct the confusion matrix and to support the statistical analysis. Each discarded proposal is counted as a false-positive (FP) or a true-negative (TN) in the confusion matrix. Likewise, the matched motorcycle proposal is counted as a false-negative (FN) or a true-positive (TP).

Manual observation finds that 337 discarded ROIs are judged as noises due to no overlapping areas with any ground truth, shown as red boxes in Fig. 7b. From those images containing the motorcycles, 36 ROIs are missed due to taillight with no other detected features, as shown in Fig. 7a. Motorcycles with broken, dim, or too-small taillights are not considered in the experiments. It is likely hard for the system to find since taillight detection is the preliminary process of



Fig. 7. The result of true negative and false negative. Red and blue boxes represent the method’s proposals of noises and motorcycles. Then, green boxes are the ground truth.

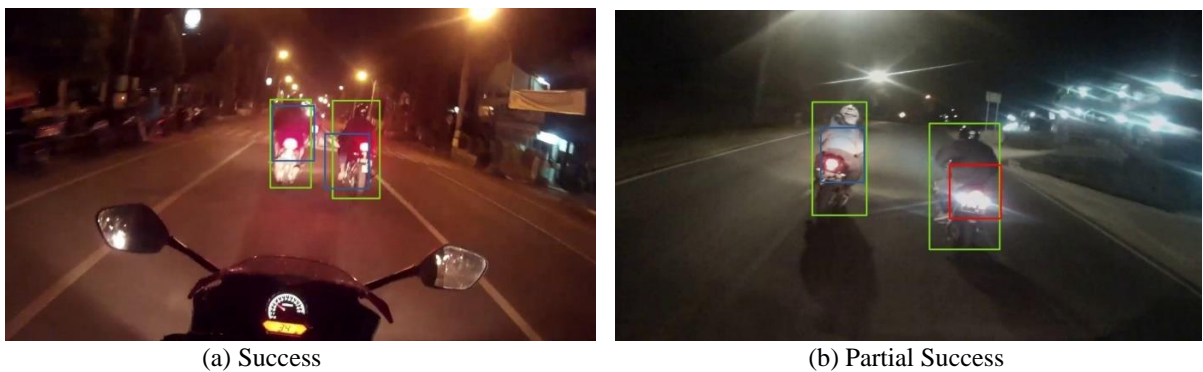


Fig. 8. The result of true positive and false positive.

generating the proposals.

The result of ANN classification is shown in Table V. Of 494 proposals containing no or partial only motorcycles, 292 are successfully detected as no motorcycles. Red boxes in Fig. 8b show some examples of proposals that are too small to be classified as a motorcycle. Of 285 proposals containing motorcycles, 209 of them are correctly classified. Figure 8a shows some examples of correctly classified proposals as motorcycles.

By finding the taillight area to propose the ROI of a motorcycle, the system can detect the taillights of the actual motorcycles with a 65.1% precision and 73.33% recall rate. This rate is much higher than the 53.6% recall by previous research [5]. Furthermore, the proposed features work well under significant feature variances of both motorcycles and non-motorcycles. The False-Positive Rate (FPR) is only 27.72%, from the failure to classify the proposals already known as noise. From the 218 testing images, the system also achieves 56.2% Average Precision (AP). Compared with the previous research [9], it detects only the motorcycle headlight to localize the object using a

similar method for car detection, while this method attempts to detect motorcycles from a more realistic view, which is their taillights. The method is also without prefiltering the car objects and is designed for various levels of darkness and motorcycle dimness. As shown in Table VI, the research can detect 285 out of the 388 motorcycles from behind, compared to previous research [9], which can detect 38 out of the 45 motorcycles using the brighter headlight of the front side.

It is worth noting that although CNN-based methods, such as VGG net [26], can achieve higher accuracy, up to 83%, as shown in Table V, it takes 76.01 seconds to finish. Meanwhile, the method in the research completes in close to one second achieving 72.71% accuracy. The research can achieve a much better speed than CNN-based methods because of the number of valuable features. The research uses only four features, while the usual CNN-based methods may have high dimensional parameters.

The true-positive or recall predicts how often the system can predict correctly that the proposal contains a correct motorcycle object. Light sources and their

TABLE V
THE RESULT OF MOTORCYCLE CLASSIFICATION ON THE PROPOSALS.

Classifier	TN	FP	FN	TP	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Prediction Time (s)
ANN + Proposed Features	292	112	76	209	72.71	65.10	73.33	68.97	1.01
VGG [26] + Resized proposals	323	74	43	249	83.02	77.09	85.27	80.98	76.01

Note: true-negative (TN) , false-positive (FP), false-negative (FN), and true-positive (TP).

TABLE VI
COMPARISON OF THE RESEARCH WITH ANOTHER PREVIOUS RESEARCH.

Method	Motorcycle	Time (s)	Overlap	View	Key Light
The research	388	0.04	285	Back	Taillight
Previous research [9]	45	0.03	38	Front	Headlight

reflection play a significant role in object detection. In this domain, a motorcycle is expected to have an excellent red light source from its taillight.

Edge pixels, edge ROIs, and object contours are features produced from the taillights and the light reflections from the motorcycle parts, such as the license plate, tires, rider’s body, and turn signals. Good motorcycle images have more edges than noise. If there is no taillight, dim red lights, or low reflection, the image may produce lower-quality edges because the difference between the motorcycle object and the noise will be low. It may lead to lower-quality proposals and failure to detect the motorcycle in the image and classify them as false negatives.

Bounding box size is also a factor in generating a good quality proposal. When the object is too small that the box has to up-scale, the background-to-foreground ratio will be more significant. The features tend to be closer to the noise than the motorcycle, and their classification counts as false negatives.

Another factor in generating a good proposal is the environment. For example, when the image is too bright or has high contrast, many reflections will be treated as taillights. Then, the negative (non-motorcycle) proposals will automatically increase by producing more false positives.

V. CONCLUSION

The research proposes a novel approach to detecting motorcycles in nighttime traffic images by processing the taillights and motorcycle reflections. The experiment shows that this approach can classify the object with 72.71% accuracy, 65.10% precision, and 73.33% recall. Furthermore, during the experiment, the classification can perform consistently in 0.04 seconds per image. Hence, the method is suitable for use in a required real-time system.

Based on observation from experiment, the research finds out that if the motorcycle has no taillight, dim red

lights, or low reflection, the image may produce lower-quality edges. The difference between the motorcycle object and the noise will be low. Hence, it may lead to lower-quality proposals and failure to detect the motorcycle in the images and classify them as false negatives.

As this topic is quite significant in developing vehicle detection applications, further study to improve its accuracy is needed. Further exploration of light reflections caused by motorcycles on the road can boost the accuracy of motorcycle identification. Many noise proposals are radiated from roadsides and non-road areas so that good road boundary detection will reduce the processing areas. It can reduce the noise proposals and improve the processing time further. An effective and yet more efficient classification method than the deep learning approach should also be investigated to make it even more practical for real-time application.

REFERENCES

- [1] M. Choudhary, “What is intelligent transport system and how it works?” 2019. [Online]. Available: <https://bit.ly/2YnQ7ys>
- [2] Badan Pusat Statistik Provinsi DKI Jakarta, “Jumlah korban kecelakaan lalu lintas menurut jenis kendaraan bermotor di provinsi DKI Jakarta 2021,” 2021. [Online]. Available: <https://bit.ly/3ZIWcFJ>
- [3] H. Kuang, K. F. Yang, L. Chen, Y. J. Li, L. L. H. Chan, and H. Yan, “Bayes saliency-based object proposal generator for nighttime traffic images,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 814–825, 2017.
- [4] F. Gao, Y. Ge, S. Lu, and Y. Zhang, “On-line vehicle detection at nighttime-based tail-light pairing with saliency detection in the multi-lane intersection,” *IET Intelligent Transport Systems*, vol. 13, no. 3, pp. 515–522, 2019.

- [5] C. L. Jen, Y. L. Chen, and H. Y. Hsiao, "Robust detection and tracking of vehicle taillight signals using frequency domain feature based Adaboost learning," in *2017 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*. Taipei, Taiwan: IEEE, June 12–14, 2017, pp. 423–424.
- [6] T.-A. Pham and M. Yoo, "Nighttime vehicle detection and tracking with occlusion handling by pairing headlights and taillights," *Applied Sciences*, vol. 10, no. 11, pp. 1–18, 2020.
- [7] H. Kuang, L. Chen, L. L. H. Chan, R. C. C. Cheung, and H. Yan, "Feature selection based on tensor decomposition and object proposal for night-time multiclass vehicle detection," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 71–80, 2018.
- [8] X. Zhang, B. Story, and D. Rajan, "Night time vehicle detection and tracking by fusing sensor cues from autonomous vehicles," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. Antwerp, Belgium: IEEE, May 25–28, 2020, pp. 1–7.
- [9] C. Fernández, D. F. Llorca, M. A. Sotelo, I. G. Daza, A. M. Hellín, and S. Álvarez, "Real-time vision-based blind spot warning system: Experiments with motorcycles in daytime/nighttime conditions," *International Journal of Automotive Technology*, vol. 14, no. 1, pp. 113–122, 2013.
- [10] Y. Miao, F. Liu, T. Hou, L. Liu, and Y. Liu, "A nighttime vehicle detection method based on YOLO v3," in *2020 Chinese Automation Congress (CAC)*. Shanghai, China: IEEE, Nov. 6–8, 2020, pp. 6617–6621.
- [11] S. Huang, Y. He, and X. A. Chen, "M-YOLO: A nighttime vehicle detection method combining Mobilenet v2 and YOLO v3," *Journal of Physics: Conference Series*, vol. 1883, pp. 1–6, 2021.
- [12] D. Huang, Z. Zhou, M. Deng, and Z. Li, "Night-time vehicle detection based on direction attention network and bayes corner localization," *Journal of Intelligent & Fuzzy Systems*, vol. 41, no. 1, pp. 783–801, 2021.
- [13] H. K. Leung, X.-Z. Chen, C. W. Yu, H. Y. Liang, J. Y. Wu, and Y. L. Chen, "A deep-learning-based vehicle detection approach for insufficient and nighttime illumination conditions," *Applied Sciences*, vol. 9, no. 22, pp. 1–27, 2019.
- [14] Y. Mo, G. Han, H. Zhang, X. Xu, and W. Qu, "Highlight-assisted nighttime vehicle detection using a multi-level fusion network and label hierarchy," *Neurocomputing*, vol. 355, pp. 13–23, 2019.
- [15] C. T. Lin, S. W. Huang, Y. Y. Wu, and S. H. Lai, "GAN-based day-to-night image style transfer for nighttime vehicle detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 951–963, 2020.
- [16] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.
- [17] J. Sklansky, "Finding the convex hull of a simple polygon," *Pattern Recognition Letters*, vol. 1, no. 2, pp. 79–83, 1982.
- [18] P. Danielsson and O. Seger, "Generalized and separable sobel operators," in *Machine vision for three-dimensional scenes*. Elsevier, 1990, pp. 347–379.
- [19] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.
- [20] J. Stewart, *Calculus*. Thomson, 2008.
- [21] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks," *Journal of Big Data*, vol. 7, no. 1, pp. 1–41, 2020.
- [22] I. D. Mienye, Y. Sun, and Z. Wang, "Prediction performance of improved decision tree-based algorithms: A review," *Procedia Manufacturing*, vol. 35, pp. 698–703, 2019.
- [23] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine learning*. Elsevier, 2020, pp. 101–121.
- [24] F. Anowar, S. Sadaoui, and B. Selim, "Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, T-SNE)," *Computer Science Review*, vol. 40, pp. 1–13, 2021.
- [25] D. M. Vida, "Improving decision tree and neural network learning for evolving data-streams," Ph.D. dissertation, Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, 2019.
- [26] M. Pak and S. Kim, "A review of deep learning in image recognition," in *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*. Kuta Bali, Indonesia: IEEE, Aug. 8–10, 2017, pp. 1–3.