

Article

3D Road Lane Classification with Improved Texture Patterns and Optimized Deep Classifier

Bhavithra Janakiraman ¹, Sathiyapriya Shanmugam ², Rocío Pérez de Prado ^{3,*}  and Marcin Wozniak ^{4,*} 

¹ Department of Computer Science and Engineering, Dr. Mahalingam College of Engineering and Technology, Pollachi 642003, India

² Department of Electronics and Communication Engineering, Panimalar Engineering College, Chennai 600123, India

³ Telecommunication Engineering Department, University of Jaén, 23700 Linares, Spain

⁴ Faculty of Applied Mathematics, Silesian University of Technology, 44-100 Gliwice, Poland

* Correspondence: rperez@ujaen.es (R.P.d.P.); marcin.wozniak@polsl.pl (M.W.)

Abstract: The understanding of roads and lanes incorporates identifying the level of the road, the position and count of lanes, and ending, splitting, and merging roads and lanes in highway, rural, and urban scenarios. Even though a large amount of progress has been made recently, this kind of understanding is ahead of the accomplishments of the present perceptual methods. Nowadays, 3D lane detection has become the trending research in autonomous vehicles, which shows an exact estimation of the 3D position of the drivable lanes. This work mainly aims at proposing a new technique with Phase I (road or non-road classification) and Phase II (lane or non-lane classification) with 3D images. Phase I: Initially, the features, such as the proposed local texton XOR pattern (LTXOR), local Gabor binary pattern histogram sequence (LGBPHS), and median ternary pattern (MTP), are derived. These features are subjected to the bidirectional gated recurrent unit (BI-GRU) that detects whether the object is road or non-road. Phase II: Similar features in Phase I are further classified using the optimized BI-GRU, where the weights are chosen optimally via self-improved honey badger optimization (SI-HBO). As a result, the system can be identified, and whether it is lane-related or not. Particularly, the proposed BI-GRU + SI-HBO obtained a higher precision of 0.946 for db 1. Furthermore, the best-case accuracy for the BI-GRU + SI-HBO was 0.928, which was better compared with honey badger optimization. Finally, the development of SI-HBO was proven to be better than the others.

Keywords: bidirectional gated recurrent unit; local Gabor binary pattern histogram sequence; local texton XOR pattern; median ternary pattern; road lane classification; self-improved honey badger optimization



Citation: Janakiraman, B.; Shanmugam, S.; Pérez de Prado, R.; Wozniak, M. 3D Road Lane Classification with Improved Texture Patterns and Optimized Deep Classifier. *Sensors* **2023**, *23*, 5358. <https://doi.org/10.3390/s23115358>

Academic Editor: Marco Leo

Received: 17 April 2023

Revised: 31 May 2023

Accepted: 1 June 2023

Published: 5 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Transportation is becoming an essential part of contemporary society's everyday routine. As the number of vehicles has rapidly increased in recent decades, the frequency of traffic accidents and the provocation of traffic jamming have become increasingly noticeable [1–3]. The entire country suffers tremendous economic losses as a result of road traffic accidents, which pose a major threat to people's safety. In this situation, more and more drivers are opting for the lane maintenance system [4–6]. In a lane-keeping system, lane line recognition and lane offset estimation are strongly tied to the implementation of the lane-keeping function, which has a direct impact on the system's resilience and real-time performance [6,7].

Road lane recognition using image processing and machine learning approaches has been a popular topic of study in both the advanced and developing worlds [8–10]. Since the number of vehicles has increased, several clever technologies have been developed to assist drivers in driving safely. Lane recognition is a significant feature of any driver assistance system. Scientists working on lane detection are now confronting numerous

important obstacles, including achieving dependability in the face of sunlight and backdrop clutter [11,12]. The advancement of image processing methods and the development of cheap visual sensor devices have opened the way for a variety of autonomous road lane recognition approaches in recent years. The actuality that the textures are alike makes automated road-lane-detecting techniques feasible. The fact that the textures of lanes are distinguishable from the background of the concrete surface makes automatic road lane recognition techniques feasible [13–15].

Due to the following scenarios, lane detection becomes difficult: With less understanding of the road geometry, lane-marking alternatives develop the presence of surrounding impediments, which obscure lane markings or are misinterpreted as lane markings. On-road signage or writing, as well as shadow effects, are frequently recommended as lane feature locations. The variation in illumination affects color and intensity. Several specialists have approved lane detection algorithms in the previous two decades. Feature-based or model-based approaches are used in traditional lane-detecting systems [16,17]. To extract lane line information, the feature-oriented method primarily employs the color and gradient variation in lane lines. The detection quality of the lane line identification schemes that depend on conventional techniques is easily influenced by climate changes owing to the limitations of the feature extraction technique [18]. When the assumption of flat ground is violated, conventional algorithms become inaccurate while recognizing lanes in the image domain and projecting them into the 3D environment in both elevation and lane curvature. Based on the recent success of CNNs in detecting lanes in 3D with the monocular depth estimation, more models are aiming to produce a series of 3D curves in camera coordinates from a single front-facing camera image, with each curve expressing either a lane delimiter or a lane center line. Moreover, the aforementioned approaches need lane line detection before calculating the lane offset for cars, the end-to-end lane offset estimation, which is not conceivable in the existing approaches. In order to overcome the above-stated limitations, the contributions are defined as follows:

1. The primary goal of this work is to propose a 3D road lane classification with improved texture patterns and an optimized deep classifier that includes Phases I (road or non-road classification) and II (lane or non-lane classification);
2. Initially, features such as the local texton XOR pattern (LTXOR), local Gabor binary pattern histogram sequence (LGBPHS), and median ternary pattern (MTP) are determined. These features are further classified using the bidirectional gated recurrent unit (BI-GRU), which determines whether there is a lane or not under various environmental conditions;
3. To improve its performance with the targets of lowering the complexity and minimizing the error, the weights in the BI-GRU are optimized by self-improved honey badger optimization (SI-HBO). Thus, the lane line problem in multi-lane scenes is efficiently recognized, and once the vehicles change lanes, the current lane scene is easily identified.

The structure of this study is specified as follows: Section 2 describes the existing research work. Section 3 explains the developed technique, and Section 4 describes the proposed features. Section 5 elaborates on the concept of two-phase classification. Section 6 presents the results and discussion and a comparative analysis. Finally, Section 7 explains the conclusion.

2. Literature Review

In 2021, Satish et al. [19] proposed a novel strategy to alert the driver when the car crosses the road border lanes using machine learning methods to prevent road accidents and ensure safe driving. The dataset's performance was measured by the production of experimental findings. The proposed technique outperformed the existing lane recognition algorithms regarding precision and accuracy. In 2021, Malik et al. [20] presented a CNN for lane offset estimation and lane line identification, which turned lane line detection difficulties into instance segmentation problems in a complicated road environment. The

network builds its example for every line in response to changes in the lane process mechanism. The global scale perceptual optimization mechanism was intended to deal with the problem, particularly as the lane path length narrows as it approaches the vanishing point. In addition, an estimation network was employed to achieve multi-tasking processing and boost performance.

In 2020, Tiago et al. [21] presented a novel road representation as well as a workflow of processes for combining two concurrent DL schemes, based on two ENet model modifications. The findings revealed that the total solution copes with the many failures in every approach, resulting in a more reliable road detection result than each technique alone. In 2021, Ting et al. [22] used a two-level CNN to create an LCR model using visual technologies. To compare with the LCR approach, a new CNN based on the AlexNet was presented. For the two models, all samples were separated into a training dataset and testing dataset. Two machine networks were compared in terms of performance. With the LCR model, the average training accuracy was over 94.6 percent. The LCR model surpassed the AlexNet model, which averaged just a 73.97 percent accuracy.

In 2020, Jau et al. [23] built a DL-based embedded road-border-detecting system. A CAE with noise removal and reconstructing features was deployed to eliminate all items in the photos excluding lane markings to generate an image with clear lane markings. The lane line's feature points were then retrieved, and a hyperbolic model was used to fit the lane line. Finally, for lane tracking, a particle filter was applied. According to the testing findings, the suggested lane-detecting method was 90.02 percent accurate for both structured and unstructured roadways. In 2019, Wang et al. [24] provided a straight-curve model-based curve identification technique, which has high application in most curved-road circumstances. By assessing the basic properties of the road images, the approach split the road image into the ROI and the road backdrop region. The ROI was further separated into two sections: straight and curved. Simultaneously, the mathematical formalism of a straight curve was constructed. Finally, the curve and straight detection and identification were accomplished, and the road lane line was recreated.

In 2021, Luo et al. [25] presented a unique and resilient multiple-lane recognition approach depending on road construction data that included five complementary specifications: length, parallel, distribution, pair, and uniform width. To choose lane candidates, all five limitations were merged into a cohesive framework based on HT. Furthermore, a dynamic programming technique to discover the most reasonable solutions from the remaining choices was provided. This technique may successfully cope with multi-lane detection's combination of complexity and interferences. In 2022, Ye et al. [26] described a scheme to extract lane characteristics from MLS point clouds on curving highways. There were four phases in the suggested technique. A road edge recognition technology was used to discern road curbs and retrieve road surfaces following data pre-processing. Then, symbols, arrows, and phrases were discovered to alert drivers in critical situations. According to the research, the proposed approaches achieved greater accuracy and resilience than most current methods.

Table 1 reviews the work on road lane detection. Here, ref. [19] used a CNN that offered high accuracy and high precision. However, a cost analysis should be made. The CNN used in [20] undergoes less loss and reliable recall, but forward collision warning is not a concern. The ENet used in [21] achieves higher reliability and less scalability, but a special network is required for fusing purposes. The LCR used in [22] offers a high detection rate and high accuracy. However, it requires the use of more datasets. Moreover, ref. [24] used a particle filter, which provided less error with high accuracy. However, the collision-avoiding model needs to be involved. In [24], an improved Hough transform was used, in which a higher accuracy and minimal time utilization were established. However, a cost analysis should be made. The Hough transform used in [25] provided higher accuracy and fewer false alarms. However, it needs more consideration of the time usage. The Gaussian distribution used in [26] offered high precision and a high F1-score. However, the spiral curves of roads were not considered.

Table 1. Review of road-lane-detecting models.

Author	Adopted Methods	Features	Challenges
Satish et al. [19]	CNN	The method is used to extract the edge features High precision	Needs consideration of the stability and computational time analysis.
Malik et al. [20]	CNN	Less loss High recall	Needs consideration of forward collision warning policy.
Tiago et al. [21]	ENet-based model	High reliability High scalability	A special network is required for fusing purposes.
Ting et al. [22]	LCR	High detection rate Higher accuracy	Requires the use of more datasets.
Jau et al. [23]	Particle filter	Less error High accuracy	A collision-avoiding model needs to be involved.
Wang et al. [24]	Improved Hough transform	High accuracy Minimal time utilization	Cost analysis should be made.
Luo et al. [25]	Hough transform	Fewer false alarms High accuracy	Needs more consideration of time usage.
Ye et al. [26]	Gaussian distribution	High precision High F1-score	Spiral curves of the road are not considered.

The aforementioned technique, however, does give a general concept as to how deep learning technology might be used to identify lane lines, but it is still unable to resolve end-to-end lane line detection in challenging environments. In spite of the challenging circumstances of lane line occlusion, high exposure, and unstructured roads, there are not many academics working on lane line identification at the moment. Additionally, the end-to-end lane offset estimation was not achieved in any of the aforementioned methods because they all used lane line detection before calculating the lane offset for cars. To overcome all these challenges, the proposed method can be used.

3. Description of Proposed Technique

This work developed Phase I (road or non-road classification) and Phase II (lane or non-lane classification) technology with 3D images. Initially, a 3D image is converted to a 2D image. Then, the detection process proceeds:

- At first, the proposed LTXOR-, LGBPHS-, and MTP-based features are extracted;
- Then, road or non-road is classified using the BI-GRU model;
- Further, these extracted features are provided for the optimal BI-GRU for lane or non-lane classification;
- For optimizing the weights in the BI-GRU, SI-HBO is deployed in this work.

The overall architecture of the proposed model on road lane classification is shown in Figure 1.

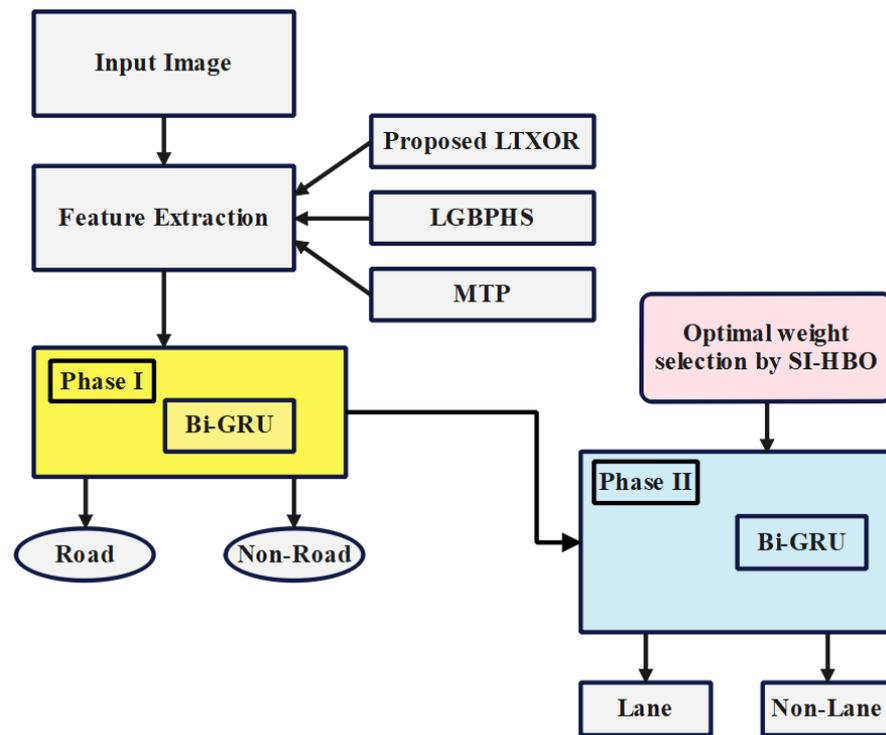


Figure 1. Overall architecture of the proposed model for road lane classification.

4. Extraction of Proposed Features

The proposed LTXOR-, LGBPMS-, MTP-based features are derived from the input image (I_{im}).

4.1. Proposed LTXOR Features

The proposed LTXOR is the improved version of the existing LXTOR features by mapping the texton shapes into a Gaussian plane. The Gaussian plane can provide a reliable estimation of their own uncertainty. Hence, the proposed features can represent the lane with more information to assist the road/non-road classifier. In the LTXOR pattern [27], seven different texton shapes were deployed for producing texton images. Here, the image (I_{im}) is divided into overlapping 2×2 sub-blocks indicated by B_1 . The positions of grey value are indicated by P, Q, R, S for analysis [27]. As per the texton shape, the sub-blocks are implied as in Equation (1):

$$Tx(Y, Z) = \begin{cases} 1, & B_1(P) = B_1(R) \& B_1(Q) \neq B_1(S) \\ 2, & B_1(Q) = B_1(P) \& B_1(S) \neq B_1(R) \\ 3, & B_1(R) = B_1(P) \& B_1(S) \neq B_1(Q) \\ 4, & B_1(P) = B_1(Q) \& B_1(R) \neq B_1(S) \\ 5, & B_1(P) = B_1(Q) \& B_1(S) \neq B_1(R) \\ 6, & B_1(Q) = B_1(P) \& B_1(R) \neq B_1(S) \\ 7, & B_1(P) = B_1(R) \& B_1(Q) = B_1(S) \\ 0, & B_1(P) \neq B_1(R) \& B_1(Q) \neq B_1(S) \end{cases} \quad (1)$$

The center of every pixel and the nearby neighbors are collected on the texton picture, following the computation of the texton on the image. The XOR function (\otimes) is executed among the center texton and neighbor. Conventionally, Equation (2) determines the LTXOR patterns [27]. As per the proposed concept, the LTXOR is modeled and updated as in Equation (3), in which $G(y, z)$ implies the Gaussian function for two dimensions, and σ implies the standard deviation, the initial value of which is taken as 0.1. σ is used for calculating the Gaussian values, those values only used to increase the LTXOR performance. In $Tx(Y, Z)$,

x represents texton shapes, which are considered for the texton image generation; Y and Z imply the distances from the origin to the horizontal and vertical axes, respectively. $Tx(b_l)$ and $Tx(b_a)$ imply the shape of the texton for the neighbor pixel (b_l) and center pixel (b_a), and \otimes implies the XOR operation among the variables. The standard deviation is the distance between the horizontal and vertical:

$$LTXOR_{G,L} = \sum_{l=1}^G 2^{(l-1)} \times \tilde{f}_3(Tx(b_l) \otimes Tx(b_a)) \quad (2)$$

$$PLTXOR_{G,L} = \sum_{l=1}^G 2^{(l-1)} \times \frac{\tilde{f}_3(Tx(b_l) \otimes Tx(b_a))}{G(y,z)} \quad (3)$$

$$G(y,z) = \frac{1}{2\pi\sigma^2} e^{-\frac{y^2+z^2}{2\sigma^2}} \quad (4)$$

$$\tilde{f}_3(y \otimes z) = \begin{cases} 1 & y \neq z \\ 0 & else \end{cases} \quad (5)$$

$$f_2(y,z) = \begin{cases} 1 & y = z \\ 0 & else \end{cases} \quad (6)$$

Furthermore, the specific texton image is transferred to LTXOR maps with values ranging from 0 to $2^p - 1$, where p represents the number of neighbors. The value of m is selected between 0 and $2^p - 1$. After computing the pattern for each pixel (j, k), the histogram construction can be derived, as shown in Equation (7):

$$His_{PLTXORP}(\tilde{m}) = \sum_{\tilde{j}=1}^{T_1} \sum_{\tilde{k}=1}^{T_2} \tilde{f}_2(PLTXORP(j,k), m); m \in [0, (2^p - 1)] \quad (7)$$

where the size of the input image is $T_1 \times T_2$.

4.2. MTP Features

The MTP [28] combines the measurement (I_{im}) of the image pixels with the integration of the median. This strategy is more resistant to speckle variation (smooth or high-textured). The arithmetic means that the intensity of nine pixels is computed after a 3×3 neighbor is formed around each pixel. This is proven quantitatively in Equation (8), with MC, t, V implying the local median, user-specified threshold, and neighbor grey level, respectively.

$$f^{MTP} = \begin{cases} 1 & V > MC + t \\ 0 & MC - t \leq V \leq MC + t \\ -1 & V < MC - t \end{cases} \quad (8)$$

Every MTP code is further split into its respective negative and positive parts, which are regarded as two binary patterns known as pos^{MTP} and neg^{MTP} . This is precisely exposed in Equations (9)–(12), where pix represents the pixel count:

$$pos^{MTP} = \sum_{pix=0}^7 f_{pos}(f_{MTP}(i_{pix})) * 2^{pix} \quad (9)$$

$$f_{pos}(V) = \begin{cases} 1 & V = 1 \\ 0 & else \end{cases} \quad (10)$$

$$neg^{MTP} = \sum_{pix=0}^7 f_{neg}(f_{MTP}(i_{pix})) * 2^{pix} \quad (11)$$

$$f_{neg}(V) = \begin{cases} 1 & V = -1 \\ 0 & else \end{cases} \quad (12)$$

4.3. LGBPHS Features

The following approach [29] uses the local histogram feature to summarize the area attribute of the LGBP patterns. To begin, each LGBP map is separated into many non-overlapping sections. After that, each region's histogram is retrieved. Lastly, to represent the provided image (I_{im}), all the histograms predicted from the areas of all the LGBP maps are combined into a unified histogram series. The following is a description of the aforementioned procedure: The histogram (H) for the image $f(p, q)$ with a grey level between $[0, Z - 1]$ is shown in Equation (13), wherein i implies the i th grey level, and h_i implies the pixel count in the image with a grey level (i) [29]:

$$h_i = \sum_{p,q} \chi\{f(p, q) = i\}, \quad i = 0, 1 \dots Z - 1 \quad (13)$$

$$\chi(f(p, q)) = \begin{cases} 1, & f(p, q) = i \\ 0, & f(p, q) \neq i \end{cases} \quad (14)$$

At last, every histogram piece calculated from every 40 LGBP maps was combined into a histogram series, where \mathfrak{R} refers to the final representation of the image, which is mentioned as $\mathfrak{R} = (G_{0,0,0}, \dots, G_{0,0,m-1}, \dots, G_{0,1,0}, \dots, G_{0,1,m-1}, \dots, G_{7,4,m-1})$ [29]. The derived feature sets, including the PLTXOR, MTP, and LGBPHS, are together appended and determined as the final feature set (fe), as in Equation (15):

$$fe = \left[\mathfrak{R} \quad f^{MTP} \quad His_{PLTXORP}(\tilde{m}) \right] \quad (15)$$

5. Two-Phase Classification Using BI-GRU + SI-HBO

5.1. Two-Phase Classification

- In Phase I, the features (fe) are subjected to BI-GRU for classification as road or non-road;
- In Phase II, the same features are then subjected to the optimized BI-GRU, which trains with the SI-HBO to determine whether they are lane or non-lane.

5.2. BI-GRU

The BI-GRU [30] includes unique gates called reset (rt) and update (ut) gates that diminish the gradient dispersal with fewer losses. The (ut) substitutes the forget and input gate of LSTM and portrays the conservation degree of former data, as in Equation (16):

$$ut = \mu(W_u \cdot [R_{t-1}, fe_t] + f_u) \quad (16)$$

In Equation (16), μ points out the sigmoid activation function between 0 and 1; fe_t stands for the input matrix at the time step (t); R_{t-1} stands for the hidden state at the prior time step ($t - 1$); W_u stands for the weight matrix of ut ; and f_u stands for the bias matrix of ut . The (rt) regulates how much chronological data have to be ignored, which is revealed in Equation (17), wherein W_r characterizes the weight matrix of rt , and f_r symbolizes the bias matrix of rt :

$$rt = \mu(W_r \cdot [R_{t-1}, fe_t] + f_r) \quad (17)$$

The hidden state of the candidate is exposed in Equation (18), wherein \tanh stands for the tanh activation function, f_R and W_R stand for the bias matrix and weight matrix of the

new cell state, respectively, and $*$ stands for the dot multiplication function. The output (R_t) shows linear intermission amid (\tilde{R}_t) and R_{t-1} :

$$\tilde{R}_t = \tanh(W_R \cdot [R_{t-1} * rg, fe_t] + f_R) \tag{18}$$

$$R_t = (1 - ut) * R_{t-1} + ut * \tilde{R}_t \tag{19}$$

The backward and forward BI-GRUs hold the prior and forthcoming details of the input data, respectively. The BI-GRU is modeled as in Equation (20). Here, \overleftarrow{R}_t and \overrightarrow{R}_t correspond to the hidden states of the backward and forward BI-GRUs, respectively; Ct refers to the combination of the outputs in two directions (for example, the multiplication function, average function, and summation function), where Yt is referred to as the output data:

$$Yt = Ct\left(\overrightarrow{R}_t, \overleftarrow{R}_t\right) \tag{20}$$

5.3. SI-HBO Model for Tuning Bi-GRU

The tuning of weights will be under the fixation of the objective, defined as in Equation (23), and the diagrammatic representation is shown in Figure 2. The explanation of SI-HBO is given below:

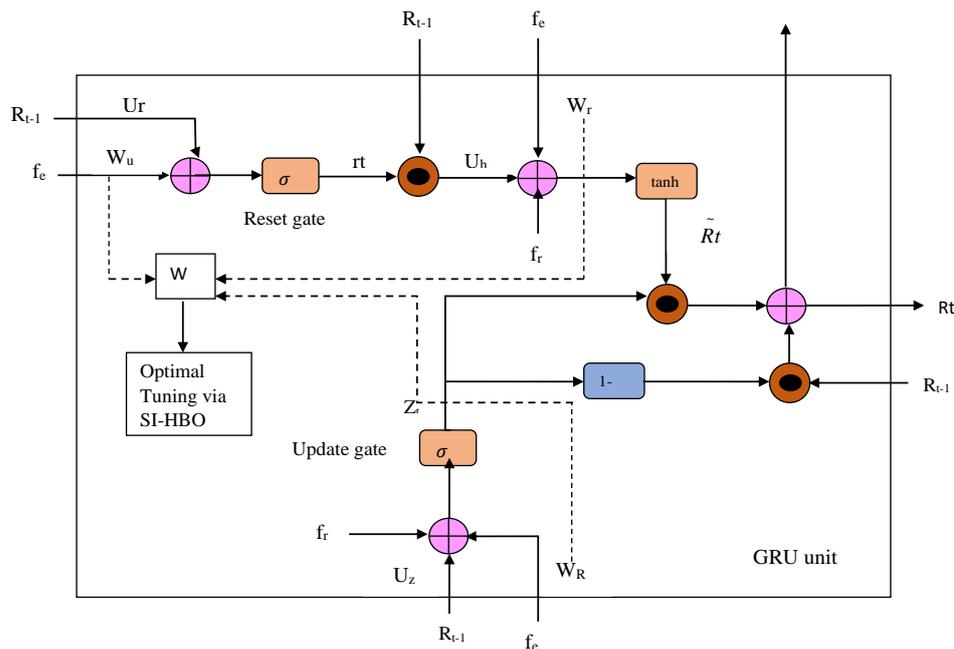


Figure 2. Optimal tuning of weights in Bi-GRU classifier.

The SI-HBO approach is improved from the HBO [31] model depending on the foraging behavior of honey badgers. Self-development is better in varied optimization schemes to minimize the root mean square error, for which the improved team optimizer is used [32]. In different fields, swarm intelligence has attracted researchers [33]. The group search algorithm is nature-inspired [34]. To combine the high-level and low-level features, the semantic embedding branch is used [35]. To solve multi-objective problems, the differential evolutionary algorithm is used [36]. The coyote optimization algorithm is used for its good tracking characteristics [37]. The ring toss game-based optimization algorithm is a population-based optimization algorithm [38]. The supply-demand optimization algorithm is competitive compared to other algorithms [39]. The grasshopper optimization

algorithm (GOA) is swarm-based [40]. The developed SI-HBO includes two chief phases: the digging phase and the honey phase.

Initialization: The populations (P) are initialized, where N implies the population size and D implies the dimension. This is exposed in Equations (21) and (22):

$$P = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1D} \\ p_{21} & p_{22} & \dots & p_{2D} \\ \dots & \dots & \dots & \dots \\ p_{N1} & p_{N2} & p_{N3} & p_{ND} \end{bmatrix} \tag{21}$$

$$p_i = LB_i + r1 * (UB_i - LB_i) \tag{22}$$

The lower and upper limits are implied by UB_i and LB_i , respectively. Moreover, a random number ($r1$) lies between 0 and 1. itr implies the current iteration, and \max^{itr} implies the maximal iteration. Using Equation (23), the searching agent’s fitness is determined.

As mentioned in Figure 2 (Phase II), the lane or non-lane classification is performed by the optimized Bi-GRU. Initially, the weight values are assigned as random values, where the weights (W) that consider $\{W_u, W_r, \text{ and } W_R\}$ are given in Equations (16)–(18). All these weight values are given as input to the Bi-GRU. The results (output) from the Bi-GRU are considered as error values, and these values are tuned optimally by the SI-HBO algorithm.

The major aim is to reduce the error (err) between the actual and predicted value from the Bi-GRU, which is defined in Equation (23):

$$obj = \min(err) \tag{23}$$

If the error is high, then the same steps as mentioned above need to be processed; in the case of vice versa, if the error is less, then these values are tuned optimally by SI-HBO. Figure 3 shows the fitness (error values) vs. iteration for the proposed SI-HBO algorithm and other existing algorithms (DOA, DA, AOA, HBO, BWO, and HBO).

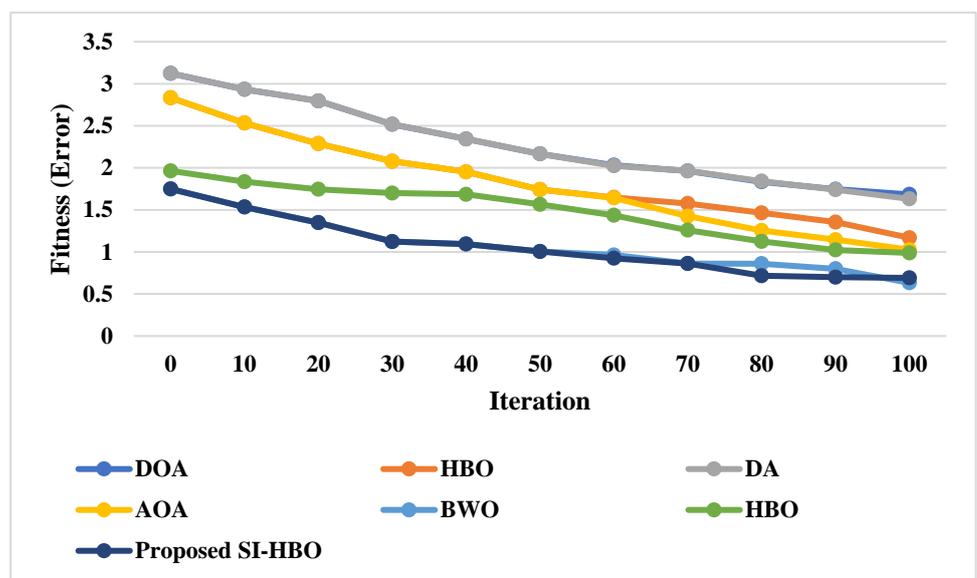


Figure 3. Fitness (error) vs. iteration for proposed SI-HBO.

The fitness prey is referred to as fit_{prey} , and the best position is implied by p_{prey} . Ensure the termination principle: $itr \leq \max^{itr}$. Use Equation (24) to update the decreasing

factor (α) that minimizes by iterations to decrease randomization with time. Here, C refers to a constant, which is taken as $C = 2$:

$$\alpha = C * \exp\left(\frac{-itr}{\max itr}\right) \quad (24)$$

For $i = 1$ to N , compute the solution's intensity (I_i), as in Equations (25)–(27), in which I_i implies the prey's scent power, and $rand2$ implies a random number between 0 and 1. The distance between the prey and the i th search agent is implied as D_i , whereas the source strength is implied as S :

$$I_i = rand2 * \left(\frac{S}{4\pi D_i^2}\right) \quad (25)$$

$$S = (p_{itr} - p_{itr+1})^2 \quad (26)$$

$$D_i = p_{prey} - p_i \quad (27)$$

Create an arbitrary integer (r) from 0 to 1. Digging phase: If $r < 0.5$, then update p_{new} as per Equation (28), in which Sp implies the speed and $time = 2$, and $rand3$, $rand4$, and $rand5$ imply random integers between 0 and 1.

$$p_{new} = p_{prey} + flag * \beta * I * p_{prey} + flag * rand3 * \alpha * D_i * \cos(2\pi rand4) * [1 - \cos(2\pi rand5)] \quad (28)$$

In the honey phase,

$$p_{new} = \frac{p_{prey} + flag * \beta * I * p_{prey} + flag * rand3 * \alpha * D_i * \cos(2\pi rand4) * [1 - \cos(2\pi rand5)]}{Sp} \quad (29)$$

$$Sp = \frac{dis \tan ce}{time} \quad (30)$$

Here, p_{prey} implies the best prey's position and $flag$ changes the searching direction. If $1 - r \geq 0.5$, then update the position as in Equation (31). Then, update the solution, as shown in Equation (32), in which da implies the diameter of the prey and honey badger side to side, rad implies the radius, and α implies the time-varying search influence form:

$$p_{new} = p_{prey} + flag * rand7 * \alpha * D_i \quad (31)$$

From Equation (31), it is detected that a honey badger executes the search near to the prey location (p_{prey}) depending on the distance (D_i). Now, the search behavior is inspired and varied in terms of time (α). Furthermore, a honey badger realizes some disturbance, which is eliminated by using updated Equation (32):

$$p_{newupdate} = p_{prey} + \frac{flag * rand7 * \alpha * D_i}{da} \quad (32)$$

$$da = 2rad \quad (33)$$

Here, $rand7$ lies between 0 and 1.

Calculate a novel position and allocate it to $p_{newupdate}$. If $f_{new} \leq f_i$, then set $p_i = p_{new}$ and $f_i = f_{new}$, and if $f_{new} \leq f_{prey}$, then set $p_{prey} = p_{new}$ and $f_{prey} = f_{new}$, and execute an arithmetic crossover.

The pseudocode of the proposed algorithm is shown in Algorithm 1.

Algorithm 1: Pseudocode of SI-HBO algorithm

```

Initialize the population with a random position
Fitness evaluation
Save best position  $p_{prey}$ 
While  $t \leq t_{max}$  do
    Update the decreasing factor  $\alpha$  using Equation (24)
    For  $i = 1$  to  $N$  do
        compute the solution's intensity  $I_i$  using Equation (25)
    If  $r < 0.5$  then
        update  $p_{new}$  as per Equation (26),
    else
        update the solution as shown in Equation (32),
    End if
    Calculate novel position and allocate it to  $p_{newupdate}$ .
    If  $f_{new} \leq f_i$ , & if  $f_{new} \leq f_{prey}$ , &  $f_{prey} = f_{new}$ ,
        execute arithmetic cross over
    End if

```

6. Results and Discussion**6.1. Simulation Setup**

The offered road-lane-detecting scheme using (BIGRU + SI-HBO) was performed in “MATLAB 2020a” on an 11th Gen Intel(R) Core (TM) i3-1115G4 @ 3.00 GHz, 3.00 GHz, 64-bit operating system, $\times 64$ -based processor, and 8.00 GB RAM. The performance of the BI-GRU + SI-HBO method was calculated over the DOA, DA, AOA, BWO, HBO, ENet [21], LSTM, CNN [20], RNN, DBN, SVM, CNN-LD [19], RF, proposed image, and conventional image. Here, the examination was made with the database denoted as db, mentioned in [41], and named as the third lane dataset. The dataset was downloaded from https://drive.google.com/file/d/1Kisxoj7mYl1YyA_4xBKTE8GGWiNZVain/view (accessed on 6 November 2022). We can randomly generate every single modeled component, from the scene's 3D geometry to the different object classes, according to the programmable methodology. The main road's lane configuration is chosen. Next, we decide whether a secondary road will exist and how many lanes it will have. The secondary road junction is seen as either a merging or a split, depending on the later orientation of the camera in the image.

The sample representation of the extracted lane images is shown in Figure 4.

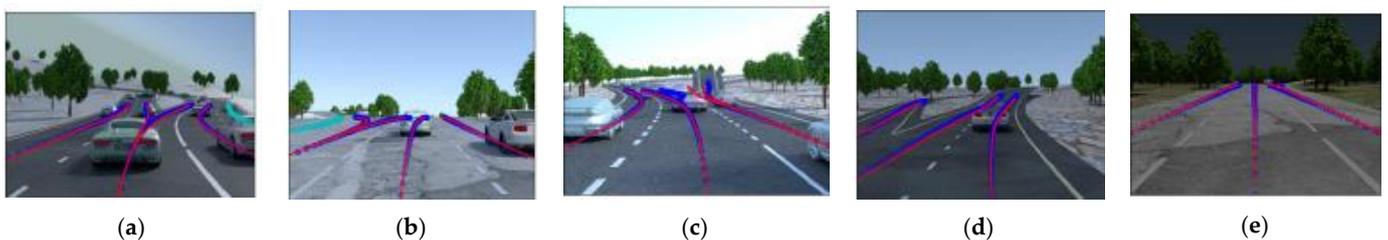


Figure 4. Three-dimensional sample representation of extracted lane images: (a) left split with divider; (b) straight road; (c) right split with divider; (d) road with divider; (e) straight road with right split.

6.2. Performance Analysis

The analysis of the suggested BI-GRU + SI-HBO was calculated over traditional schemes on disparate metrics. The assessment of the BI-GRU + SI-HBO was performed over traditional models, such as the DOA, DA, AOA, BWO, HBO, E-Net [21], LSTM, CNN [20], CNN-LD [19], RNN, DBN, and SVM, and the RF models are presented in Figures 5–7 for LRs from 60, 70, 80, and 90. Here, Figures 5–7 explain the evaluation of the BI-GRU + SI-HBO over the traditional BI-GRU + DOA, BI-GRU + DA, BI-GRU + AOA, BI-GRU + BWO, and BI-GRU + HBO for db 1 to determine whether it is lane or non-lane.

Table 2 explains the estimation of the BI-GRU + SI-HBO over the traditional ENet [21], LSTM, CNN [20], RNN, DBN, SVM, and RF for db 1 and db 2 to determine whether it is road or non-road. Here, the offered BI-GRU + SI-HBO offered superior outputs to the BI-GRU + DOA, BI-GRU + DA, BI-GRU + AOA, BI-GRU + BWO, BI-GRU + HBO, ENet [21], LSTM, CNN [20], RNN, DBN, SVM, and RF. In Figure 5b, the accuracy for the BI-GRU + SI-HBO is higher at the 90th LR than at the 60th, 70th, and 80th LRs. At the 60th LR, the BI-GRU + SI-HBO has a lesser accuracy than at the 70th, 80th, and 90th LRs. Likewise, from Table 2, the BI-GRU + SI-HBO had the best outcome for precision at 0.946 for db 1. Here, the accuracies for the SVM and RF are much lower, whilst the RNN has a high accuracy next to the BI-GRU + SI-HBO. This is a novel method for recognizing 3D road lanes that extracts many features, including the suggested LTXOR, LGBPFS, and MTP. Then, using an efficient BI-GRU classification procedure, it can be determined whether the object is a road or not. Then, with the aid of the SI-HBO algorithm, the optimal weights for the BI-GRU are determined. By optimally tuning the weights, precise and accurate detection is achieved. Thus, the advantage of BI-GRU + SI-HBO is established over BI-GRU + DOA, BI-GRU + DA, BI-GRU + AOA, BI-GRU + BWO, BI-GRU + HBO, E-Net [21], LSTM, CNN [20], RNN, DBN, SVM, and RF.

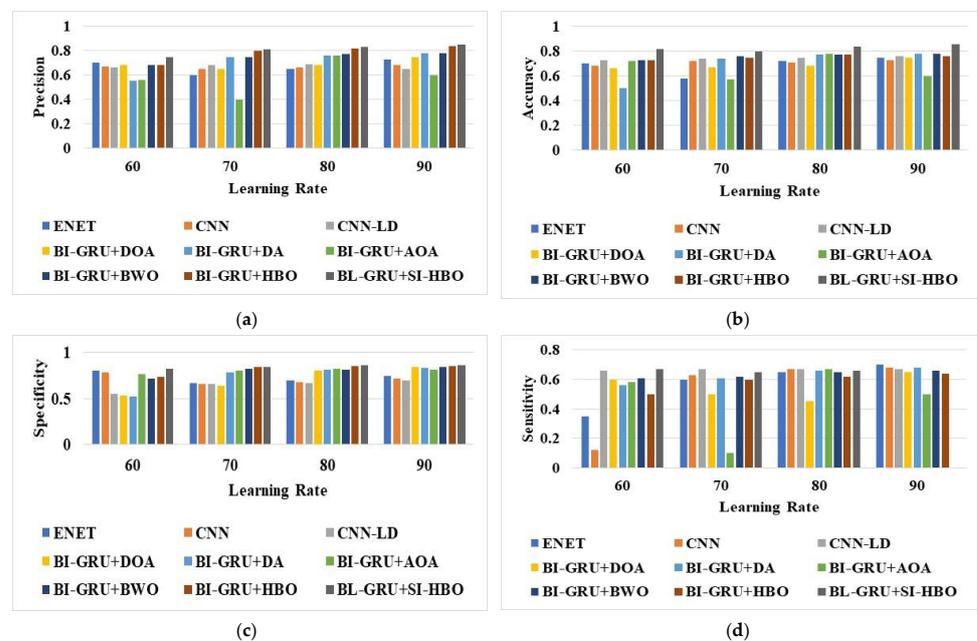


Figure 5. Analysis via BI-GRU + SI-HBO over other schemes ([19–21]) for (a) precision, (b) accuracy, (c) specificity, and (d) sensitivity for db 1.

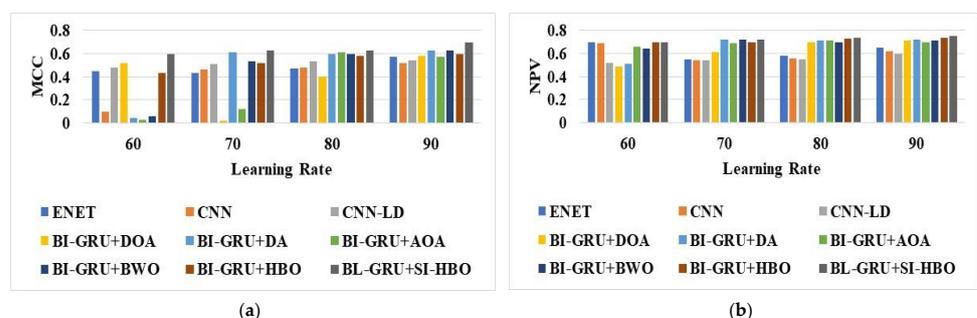
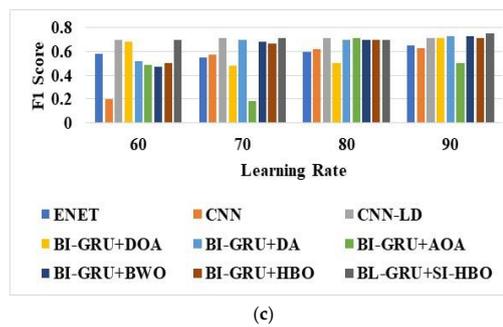
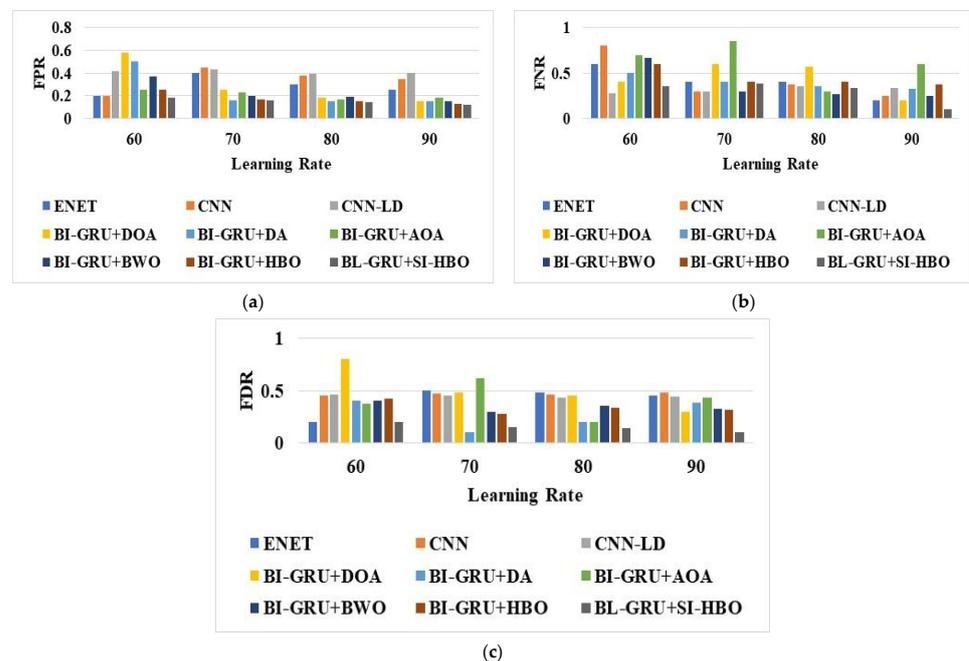


Figure 6. Cont.



(c)

Figure 6. Analysis via BI-GRU + SI-HBO over other schemes ([19–21]) for (a) MCC, (b) NPV, and (c) F1-score for db 1.



(a)

(b)

(c)

Figure 7. Analysis via BI-GRU + SI-HBO over other schemes ([19–21]) for (a) FPR, (b) FNR, and (c) FDR for db 1.

Table 2. Analysis via BI-GRU over other classifier schemes using db 1 and db 2.

	BI-GRU + SI-HBO	ENet [21]	CNN-LD [19]	LSTM	CNN [20]	RNN	DBN	SVM	RF
Sensitivity	0.910	0.264	0.275	0.037	0.226	0.245	0.075	0.528	0.094
Accuracy	0.928	0.631	0.658	0.552	0.596	0.640	0.56	0.552	0.552
NPV	0.945	0.928	0.754	0.905	0.918	0.913	0.933	0.573	0.908
Specificity	0.945	0.930	0.983	0.725	0.918	0.903	0.903	0.573	0.910
F1-Score	0.928	0.465	0.492	0.072	0.342	0.388	0.137	0.437	0.138
FNR	0.089	0.735	0.739	0.962	0.773	0.754	0.924	0.471	0.905
Precision	0.946	0.823	0.723	0.915	0.705	0.928	0.887	0.518	0.625
FPR	0.054	0.079	0.945	0.275	0.081	0.163	0.163	0.426	0.091
MCC	0.855	0.301	0.678	0.143	0.202	0.347	0.143	0.101	0.088
FDR	0.053	0.176	0.156	0.092	0.294	0.071	0.286	0.481	0.375

6.3. Statistical Analysis on Accuracy

Table 3 highlights the statistical study conducted via the employed BI-GRU + SI-HBO over conventional models (BI-GRU + DOA; BI-GRU + DA; BI-GRU + AOA; BI-GRU + BWO; and BI-GRU + HBO) on accuracy. The metaheuristic schemes are stochastic, and to substantiate their fair evaluation, each model was analyzed quite a lot of times to

accomplish high accuracy. An accuracy of 0.928 was gained with the BI-GRU + SI-HBO for the best case, whilst the BI-GRU + DOA, BI-GRU + DA, BI-GRU + AOA, BI-GRU + BWO, and BI-GRU + HBO achieved lesser accuracies for the best case. Similarly, superior outputs were obtained for the BI-GRU + SI-HBO for the mean case. These enhancements are owing to the incorporated, enhanced LT-XOR and optimized BI-GRU concepts.

Table 3. Statistical study on accuracy.

Metrics	Standard Deviation	Worst	Variance	Mean	Best
BI-GRU + DOA	0.108	0.580	0.011	0.655	0.816
BI-GRU + DA	0.108	0.580	0.011	0.655	0.815
BI-GRU + AOA	0.173	0.407	0.030	0.622	0.833
BI-GRU + BWO	0.088	0.647	0.007	0.767	0.855
BI-GRU + HBO	0.047	0.722	0.002	0.775	0.829
BI-GRU + SI-HBO	0.051	0.815	0.002	0.868	0.928

6.4. Comparative Analysis

An improved specificity of 0.93 is noted for the BI-GRU + SI-HBO, which is better than the BI-GRU+ SI-HBO + conventional LT-XOR, the BI-GRU+ SI-HBO without feature extraction, and the suggested method without optimization. Next to the BI-GRU + SI-HBO, the developed model without feature extraction revealed better values than the BI-GRU+ SI-HBO + conventional LT-XOR and the suggested method without optimization. This development is owing to the enhanced LT-XOR and SI-HBO concepts. The image results of the proposed and conventional methods are shown in Figure 8. From the results, it is proven that the proposed algorithm is more efficient at detecting the road and lane when compared to other methods, such as ENet [21], CNN [20], and CNN-LD [19].

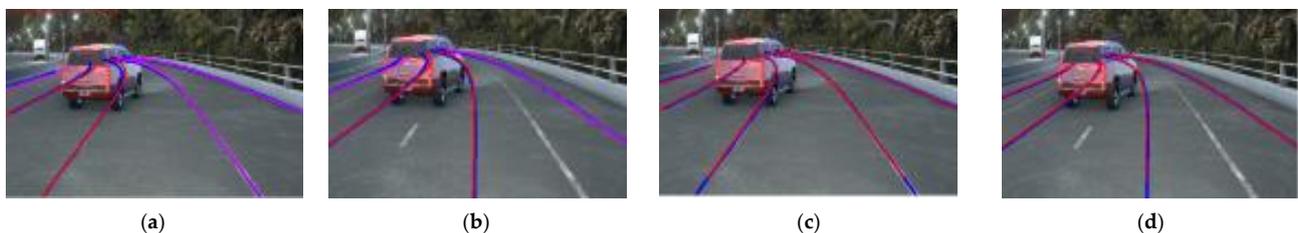


Figure 8. Image results of proposed and conventional road lane detection algorithms: (a) BI-GRU+SI-HBO; (b) ENet [21]; (c) CNN [20]; (d) CNN-LD [19].

The developed BI-GRU + SI-HBO technique was analyzed with the recommended scheme with conservative LT-XOR, the suggested method without feature extraction in Table 4, using db 1 and db 2. Likewise, the developed BI-GRU + SI-HBO technique is analyzed with the BI-GRU+ SI-HBO + conventional LT-XOR, the BI-GRU+ SI-HBO without feature extraction, and the suggested method without optimization in Table 5 using db 1.

Table 4. Comparison of proposed model with conventional ones using db 1 and db 2.

Metrics	Proposed Model	Proposed Model with Conventional LTXOR	Proposed Model without Feature Extraction
Sensitivity	0.730	0.823	0.678
Accuracy	0.912	0.763	0.843
FPR	0.067	0.476	0.225
Specificity	0.932	0.523	0.733
Precision	0.919	0.679	0.755

Table 4. Cont.

Metrics	Proposed Model	Proposed Model with Conventional LTXOR	Proposed Model without Feature Extraction
FNR	0.269	0.172	0.321
F1-Score	0.814	0.809	0.808
FDR	0.081	0.320	0.121
MCC	0.674	0.597	0.721
NPV	0.902	0.523	0.245

Table 5. Comparison of the proposed model with conventional ones using db 1.

Metrics	BI-GRU + SI-HBO	BI-GRU + SI-HBO with Conventional LTXOR	BI-GRU + SI-HBO without Feature Extraction	Proposed Model without Optimization
Sensitivity	0.910	0.893	0.853	0.793
Accuracy	0.928	0.771	0.815	0.719
FPR	0.0540	0.464	0.344	0.516
Specificity	0.945	0.535	0.655	0.483
Precision	0.946	0.690	0.716	0.619
FNR	0.089	0.124	0.142	0.216
F1-Score	0.928	0.816	0.834	0.764
MCC	0.855	0.608	0.685	0.54
FDR	0.053	0.309	0.283	0.381
NPV	0.945	0.535	0.655	0.484

6.5. Discussion

Currently, achieving reliability in changes in lighting and background clutter is one of the main challenges confronting researchers working on road lane detection. A number of automatic road lane detection techniques have emerged in recent years as a result of improvements in image processing techniques and the availability of low-cost visual sensing devices. The fact that lane textures can be easily distinguished from the pavement surface backdrop contributes to the viability of the automatic road lane detection method. Applying learning techniques and image processing techniques has increased the accuracy and productivity in recent years. Although these methods concentrate on identifying the lane from a single frame, they typically offer performances that are potentially unsatisfactory when dealing with some extreme circumstances, such as lane line degradation, large shadows, significant vehicle occlusion, noisy image inputs, etc. Practically, lanes should be continuous line formations on the road. As a result, information from earlier frames can be used to extrapolate the location of a lane that cannot be exactly detected in the live frame.

From the overall analysis, research on 3D lane identification, which provides an accurate estimate of the 3D position of the drivable lanes, is becoming popular. The primary goal of this work is to propose a novel method using 3D and Phases I (road or non-road classification) and II (lane or non-lane classification). Particularly, the BI-GRU + SI-HBO obtained the greatest result for db 1 with 0.946 precision. The SVM and RF accuracies in this situation were significantly reduced; however, the RNN's accuracy increased when compared to the BI-GRU + SI-HBO. The best-case accuracy for the BI-GRU + SI-HBO is 0.928, while the best-case accuracies for the DOA, DA, AOA, BWO, and HBO were less. Further, the analysis of the BI-GRU for the classification of road or non-road using db 1 and db 2 was performed for varied LRs. The positive metrics attained improved outcomes, while the negative metrics attained worse outcomes. From this analysis, the BI-GRU exposed a high specificity of 0.93 at the 90th LR, while at the 60th LR, the BI-GRU exposed a comparatively lesser specificity of 0.6279. Thus, better outcomes were attained at the 90th LR. Finally, the convergence of the SI-HBO scheme over the DOA, DA, AOA, BWO, and HBO for diverse iterations was analyzed in this study. From the convergence

analysis, the SI-HBO gained less cost from the 10th to 50th iterations. A lesser convergence of 1.076 was accomplished using SI-HBO rather than the DOA, DA, AOA, BWO, and HBO. Thus, improved results are achieved with the SI-HBO method.

7. Conclusions

This work suggests a novel technique with Phase I (road or non-road classification) and Phase II (lane or non-lane classification). Initially, the features, such as the proposed LTXOR, LGBPHS, and MTP, were derived, which were then categorized via the BI-GRU, which detected whether the object was road or non-road. The similar features in a phase were then categorized via the optimal BI-GRU, wherein the weights were chosen via SI-HBO. Thus, it could be determined whether the object was lane or non-lane. The BI-GRU + SI-HBO, especially, gained the best precision at 0.946 for db 1. Here, the accuracies for the SVM and RF were much lower, whilst the RNN gained a high accuracy next to the BI-GRU + SI-HBO. An accuracy of 0.928 was gained with the BI-GRU + SI-HBO for the best case, whilst the DOA, DA, AOA, BWO, and HBO achieved lesser accuracies for the best case. In the future, effective perception via advanced methods will be proposed to study the efficient integration of sensors to minimize the computation time and cost.

Author Contributions: The paper investigation, resources, data curation, writing—original draft preparation, writing—review and editing, and visualization were performed by B.J. The paper conceptualization and software were conducted by S.S. The validation and formal analysis, methodology, supervision, project administration, and funding acquisition of the version to be published were conducted by R.P.d.P. and M.W. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge contributions to this project from the Rector of the Silesian University of Technology under pro quality grant no. 09/010/RGJ23/0076. In addition, this research is supported by Spanish Research Projects P18-RT-4040 and PID2020-119082RB-C21.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

Abbreviation	Description
AOA	Arithmetic Optimization Algorithm
BWO	Black Widow Optimization
BI-GRU	Bidirectional Gated Recurrent Unit
CNN	Convolutional Neural Network
CAE	Convolution Auto-Encoder
DL	Deep Learning
DOA	Dingo Optimization
DA	Dragonfly Algorithm
DBN	Deep Belief Network
HBO	Honey Badger Optimization
HT	Hough Transform
LCR	Lane-Changing Recognition
LSTM	Long Short-Term Memory
LGBPHS	Local Gabor Binary Pattern Histogram Sequence
LTXOR	Local Texton Xor Pattern
LR	Learning Rate

MTP	Median Ternary Pattern
MLS	Mobile Laser Scanning
MCC	Matthews Correlation Coefficient
NPV	Negative Predictive Value
ROI	Region Of Interest
RNN	Recurrent Neural Network
RF	Random Forest
SVM	Support Vector Machine
SI	Self-Improved

References

- Dewangan, D.K.; Sahu, S.P. Driving Behavior Analysis of Intelligent Vehicle System for Lane Detection Using Vision-Sensor. *IEEE Sens. J.* **2021**, *21*, 6367–6375. [\[CrossRef\]](#)
- Lin, C.-Y.; Lian, F.-L. System Integration of Sensor-Fusion Localization Tasks Using Vision-Based Driving Lane Detection and Road-Marker Recognition. *IEEE Syst. J.* **2020**, *14*, 4523–4534. [\[CrossRef\]](#)
- Feng, Z.; Li, M.; Stolz, M.; Kunert, M.; Wiesbeck, W. Lane Detection with a High-Resolution Automotive Radar by Introducing a New Type of Road Marking. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 2430–2447. [\[CrossRef\]](#)
- Lin, C.; Guo, Y.; Li, W.; Liu, H.; Wu, D. An Automatic Lane Marking Detection Method with Low-Density Roadside LiDAR Data. *IEEE Sens. J.* **2021**, *21*, 10029–10038. [\[CrossRef\]](#)
- Lu, P.; Xu, S.; Peng, H. Graph-Embedded Lane Detection. *IEEE Trans. Image Process.* **2021**, *30*, 2977–2988. [\[CrossRef\]](#)
- Zhang, Y.; Lu, Z.; Ma, D.; Xue, J.-H.; Liao, Q. Ripple-GAN: Lane Line Detection with Ripple Lane Line Detection Network and Wasserstein GAN. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 1532–1542. [\[CrossRef\]](#)
- Lopez-Guede, J.M.; Izquierdo, A.; Estevez, J.; Graña, M. Active learning for road lane landmark inventory with V-ELM in highly uncontrolled image capture conditions. *Neurocomputing* **2021**, *438*, 259–269. [\[CrossRef\]](#)
- Arya, D.; Maeda, H.; Ghosh, S.K.; Toshniwal, D.; Mraz, A.; Kashiyama, T.; Sekimoto, Y. Deep learning-based road damage detection and classification for multiple countries. *Autom. Constr.* **2021**, *132*, 103935. [\[CrossRef\]](#)
- Fang, J.; Qu, B.; Yuan, Y. Distribution equalization learning mechanism for road crack detection. *Neurocomputing* **2021**, *424*, 193–204. [\[CrossRef\]](#)
- Grinias, I.; Panagiotakis, C.; Tziritas, G. MRF-based segmentation and unsupervised classification for building and road detection in peri-urban areas of high-resolution satellite images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *122*, 145–166. [\[CrossRef\]](#)
- Xu, F.; Hu, B.; Chen, L.; Wang, H.; Xia, Q.; Sehdev, P.; Ren, M. An illumination robust road detection method based on color names and geometric information. *Cogn. Syst. Res.* **2018**, *52*, 240–250. [\[CrossRef\]](#)
- Jayaprakash, A.; KeziSelvaVijila, C. Feature selection using Ant Colony Optimization (ACO) and Road Sign Detection and Recognition (RSDR) system. *Cogn. Syst. Res.* **2019**, *58*, 123–133. [\[CrossRef\]](#)
- Geng, L.; Sun, J.; Xiao, Z.; Zhang, F.; Wu, J. Combining CNN and MRF for road detection. *Comput. Electr. Eng.* **2018**, *70*, 895–903. [\[CrossRef\]](#)
- Liu, Y.; Xu, W.; Dobaie, A.M.; Zhuang, Y. Autonomous road detection and modeling for UGVs using vision-laser data fusion. *Neurocomputing* **2018**, *275*, 2752–2761. [\[CrossRef\]](#)
- Li, Y.; Ding, W.; Zhang, X.; Ju, Z. Road detection algorithm for Autonomous Navigation Systems based on dark channel prior and vanishing point in complex road scenes. *Robot. Auton. Syst.* **2016**, *85*, 1–11. [\[CrossRef\]](#)
- Ochman, M. Hybrid approach to road detection in front of the vehicle. *IFAC-Pap.* **2019**, *52*, 245–250. [\[CrossRef\]](#)
- Xiao, L.; Wang, R.; Dai, B.; Fang, Y.; Liu, D.; Wu, T. Hybrid conditional random field based camera-LIDAR fusion for road detection. *Inf. Sci.* **2018**, *432*, 543–558. [\[CrossRef\]](#)
- Zhu, M.; Liu, Y.; Zhuang, Y.; Hu, H. Visual Campus Road Detection for an UGV using Fast Scene Segmentation and Rapid Vanishing Point Estimation. *IFAC Proc. Vol.* **2014**, *47*, 11898–11903. [\[CrossRef\]](#)
- Satti, S.K.; Devi, K.S.; Dhar, P.; Srinivasan, P. A machine learning approach for detecting and tracking road boundary lanes. *ICT Express* **2021**, *7*, 99–103. [\[CrossRef\]](#)
- Haris, M.; Hou, J.; Wang, X. Multi-scale spatial convolution algorithm for lane line detection and lane offset estimation in complex road conditions. *Signal Process. Image Commun.* **2021**, *99*, 116413. [\[CrossRef\]](#)
- Almeida, T.; Lourenço, B.; Santos, V. Road detection based on simultaneous deep learning approaches. *Robot. Auton. Syst.* **2020**, *133*, 103605. [\[CrossRef\]](#)
- Xu, T.; Zhang, Z.; Wu, X.; Qi, L.; Han, Y. Recognition of lane-changing behaviour with machine learning methods at freeway off-ramps. *Phys. A Stat. Mech. Its Appl.* **2021**, *567*, 125691. [\[CrossRef\]](#)
- Perng, J.W.; Hsu, Y.W.; Yang, Y.Z.; Chen, C.Y.; Yin, T.K. Development of an embedded road boundary detection system based on deep learning. *Image Vis. Comput.* **2020**, *100*, 103935. [\[CrossRef\]](#)
- Wang, H.F.; Wang, Y.F.; Zhao, X.; Wang, G.P.; Huang, H.; Zhang, J. Lane Detection of Curving Road for Structural Highway with Straight-Curve Model on Vision. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5321–5330. [\[CrossRef\]](#)
- Luo, S.; Zhang, X.; Hu, J.; Xu, J. Multiple Lane Detection via Combining Complementary Structural Constraints. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 7597–7606. [\[CrossRef\]](#)

26. Ye, C.; Zhao, H.; Ma, L.; Jiang, H.; Li, H.; Wang, R.; Chapman, M.A.; Junior, J.M.; Li, J. Robust Lane Extraction from MLS Point Clouds Towards HD Maps Especially in Curve Road. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1505–1518. [[CrossRef](#)]
27. Bala, A.; Kaur, T. Local texton XOR patterns: A new feature descriptor for content-based image retrieval. *Eng. Sci. Technol. Int. J.* **2016**, *19*, 101–112. [[CrossRef](#)]
28. Khan, A.; Bashar, F.; Ahmed, F.; Kabir, H. Median ternary pattern (MTP) for face recognition. In Proceedings of the International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, Bangladesh, 17–18 May 2013. [[CrossRef](#)]
29. Zhang, W.; Shan, S.; Gao, W.; Chen, X.; Zhang, H. Local Gabor binary pattern histogram sequence (LGBPHS): A novel non-statistical model for face representation and recognition. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05), Beijing, China, 17–21 October 2005.
30. Li, X.; Ma, X.; Xiao, F.; Xiao, C.; Wang, F.; Zhang, S. Time-series production forecasting method based on the integration of Bidirectional Gated Recurrent Unit (Bi-GRU) network and Sparrow Search Algorithm (SSA). *J. Pet. Sci. Eng.* **2022**, *208*, 109309. [[CrossRef](#)]
31. Hashim, F.A.; Houssein, E.H.; Hussain, K.; Mabrouk, M.S.; Al-Atabany, W. Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems. *Math. Comput. Simul.* **2022**, *192*, 84–110. [[CrossRef](#)]
32. Dehghani, M.; Trojovský, P. Teamwork Optimization Algorithm: A New Optimization Approach for Function Minimization/Maximization. *Sensors* **2021**, *21*, 4567. [[CrossRef](#)]
33. Dhawan, S.; Gupta, R.; Rana, A.; Sharma, S. Various Swarm Optimization Algorithms: Review, Challenges, and Opportunities. *Soft Comput. Intell. Syst.* **2021**, *2021*, 291–301.
34. Abualgah, L. Group search optimizer: A nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. *Neural Comput. Appl.* **2021**, *33*, 2949–2972. [[CrossRef](#)]
35. Li, J.; Jiang, F.; Yang, J.; Kong, B.; Gogate, M.; Dashtipour, K.; Hussain, A. Lane-deeplab: Lane semantic segmentation in automatic driving scenarios for high-definition maps. *Neurocomputing* **2021**, *465*, 15–25. [[CrossRef](#)]
36. Jamali, A.; Mallipeddi, R.; Salehpour, M.; Bagheri, A. Multi-objective differential evolution algorithm with fuzzy inference-based adaptive mutation factor for Pareto optimum design of suspension system. *Swarm Evol. Comput.* **2020**, *54*, 100666. [[CrossRef](#)]
37. Diab, A.A.Z.; Sultan, H.M.; Do, T.D.; Kamel, O.M.; Mossa, M.A. Coyote Optimization Algorithm for Parameters Estimation of Various Models of Solar Cells and PV Modules. *IEEE Access* **2020**, *8*, 111102–111140. [[CrossRef](#)]
38. Doumari, S.A.; Hadi, G.; Mohammad, D.; Om, P.K. Ring toss game-based optimization algorithm for solving various optimization problems. *Int. J. Intell. Eng. Syst.* **2021**, *14*, 545–554.
39. Ginidi, A.R.; Shaheen, A.M.; El-Sehiemy, R.A.; Elattar, E. Supply demand optimization algorithm for parameter extraction of various solar cell models. *Energy Rep.* **2021**, *7*, 5772–5794. [[CrossRef](#)]
40. Meraihi, Y.; Gabis, A.B.; Mirjalili, S.; Ramdane-Cherif, A. Grasshopper optimization algorithm: Theory, variants, and applications. *IEEE Access* **2021**, *9*, 50001–50024. [[CrossRef](#)]
41. Garnett, N.; Cohen, R.; Pe'Er, T.; Lahav, R.; Levi, D. 3D-LaneNet: End-to-End 3D Multiple Lane Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.