LANE AND VEHICLE DETECTION USING HOUGH TRANSFORM AND YOLOv3

Subash Kumar
Department of Computer Science & Engineering
School of Engineering & Technology
Sharda University
Greater Noida, India
2018000119.subash@ug.sharda.ac.in

Kartikeya
Department of Computer Science & Engineering
School of Engineering & Technology
Sharda University
Greater Noida, India
2018007496.kartikeya@ug.sharda.ac.in

Sunil Kumar
Department of Computer Science & Engineering
School of Engineering & Technology
Sharda University
Greater Noida, India
sunil.kumar7@sharda.ac.in

Nikhil Gupta
Department of Computer Science & Engineering
School of Engineering & Technology
Sharda University
Greater Noida, India
2019006638.nikhil@ug.sharda.ac.in

Agrima Yadav
Department of Computer Science & Engineering
School of Engineering & Technology
Sharda University
Greater Noida, India
2018006287.agrima@ug.sharda.ac.in

Abstract - **Object tracking at dark is critical to minimizing the number of nocturnal traffic crashes. This paper presents a deep convolutional neural network dubbed M-YOLO to enhance the precision of nocturnal object recognition and to be suited for limited contexts (also including microcontrollers in automobiles). To begin, track line images are separated into other * 2S panels based on the features of uneven spatial and temporal dispersion densities. Additionally, the sensor frequency has been limited to four measurement levels, making it even more suited for tiny source localization, like lateral distance measurement. Thirdly, to optimize the connectivity, a fully connected layer throughout the basic Yolo v3 method is reduced by 53 to 49 levels. Lastly, characteristics like cluster center radius and backpropagation are enhanced.**

Keywords—Deep Learning, Vehicle Detection, Hough Transform, LaneRTD Algorithm, Canny Edge Sensor

I. INTRODUCTION

Estimating vehicle flow factors are critical in the development, engineering, and management of comprehensive road infrastructures. Furthermore, Intelligent Transportation Systems (ITS) need the monitoring and regulation of travel times and flow velocity in legitimate with greater precision and efficiency than those in the prior [1]. Many concepts and tools are available for collecting traffic information, such as "static point assessment" (i.e., deductive approach, hydraulic pipes, sensors, camcorders, etc.) as well as "gauge automobile information" techniques (i.e., vehicle trajectory information FCD), which are both frequently utilized.

Numerous Deep Learning-based strategies may be used in static point measuring approaches for automobile tracking and identification procedures [2–5]. With the fast evolution of China's automotive sector as well as the constant advancement of machine learning systems in subsequent generations, automobiles with ancillary commuting features and the automated vehicular feature will constitute the majority of the percentile rank of motor automobiles just in the business sector by 2020 [1]. The establishment of a smart automotive sector may indeed cause moral indignation, the stress distribution of vehicle riding, and tackle the issues of environmental depletion of natural resources induced by transportation jams.

The automated recognition of cycle lanes just on the roadway is indeed the technological underpinning for automobile-aided navigation as well as an essential piece of such theory was developed in autonomous cars [2]. Somewhere at the moment, the major mechanisms of traffic signs used domestically and overseas are as follows: the technique of recovering route characteristics using computer vision technology, the way of constructing a roadway design for recognition, and indeed the technique of multi-sensor fusing sensing [3]. As the number of cars on the road has grown throughout the years, so has the number of casualties. It kind of endangers people's well-being but also generates massive financial damages. Unless a computer network can indeed be utilized to observe the immediate data in real-time while operating and detect potential hazards as soon as feasible, traveling efficiency might be considerably enhanced. Traffic monitoring, like that of the foundation and basic function of autonomous navigation systems, has piqued the interest of academics.

Automobile recognition is particularly critical at dark when the illumination and clarity are low and indeed the likelihood of a vehicle crash is greater. There are two kinds of important transport recognition strategies: those associated with classical image retrieval and those based on deep neural networks. Conventional object recognition systems employ low-grade characteristics for edge detection [1], notably automobile offers breath taking at night time. Hsia detected the lane markings using the Hough transform, and then after acquiring the led source information for every track, the automobile number was calculated based on the viewing angle as well as dispersion of the background light [2]. Jong provided an enhanced multi-scale retina (MSR) contrast augmentation technique, and a two-stage predictor relying on BoF and CNN was employed to eliminate false alarms [3]. In the above dark automobile recognition approach is directly persuaded by roadway sceneries because it is reliant on automobile beam attributes and algorithms. As a result, it is dependent on human background subtraction.

II. LITERATURE SURVEY.

A. SUMMARY OF THE LANERTD ALGORITHM

The LaneRTD approach has the benefit of just using just a solitary Imaging system. To record the whole front driver's view, these lenses must be positioned mostly on the vehicle's front-windshield reflector. Besides presuming that perhaps the standard is sufficiently level, the skyline is guaranteed to just be aligned towards the X-axis mostly in the snapshot. Nevertheless, if indeed the skyline is indeed not aligned to that same X-axis, then the device's measurement information might be analysed to correct the alignment. To be clear, the road borders are typically organized in sets of markers that are often square in shape (or approximate). LaneRTD (Lane Real-Time Detection) seems to be another sort of technique stated [5]. The usage of this method in real life is indeed a benefit. The key operation of this technique is the reduction of signal-to-noise ratio as well as the recognition of vehicle channels' borders using Canny edge identification [6]. Canny detecting sides enable others to delete borders that do not "please" them in order to achieve the maximum representation of the shape of such seen item. It really is mainly used on black-and-white photos with such a Gaussian filter added [7], which enables them to eliminate clutter. It really is vital to note that now the Hugh transformation [8] is used to identify the locations of something like the driving lane, although, from the dynamic array results, it is feasible to show the car lane just on video sequence using a few more techniques. This project is focused here on the form of channel identification and tracking. Specific parameters have such a three-channel memory. As shown in Figure 2, H, S, V, and H, S, L indicate the color tone, S color immersion, V color depth, and L color brightness. Furthermore, several other color intensity schemes, including such HSV (Hue, Saturation, Value) as well as HSL (Hue, Saturation, Value) Saturation, Crispness, could be stated to produce the lane of traffic whatever is easier identified using incoming films. Filters have such a three-channel memory. As shown in Figure 2, H, S, V as well as H, S, L indicate the shades, S color saturation, V color change, and L color luminosity. When employing such places, you may come into the need to define a barrier so that roadways are more protected and feasible. Specific parameters have such a three-channel memory. As shown in Figure 2, H, S, V, and H, S, L indicate the color tone, S color immersion, V color depth, and L color brightness.

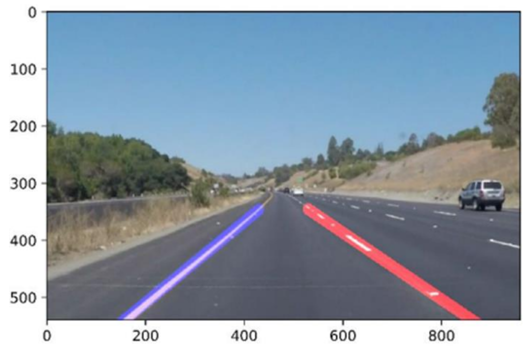


Fig. 1. Detected lane boundaries by the LaneRTD algorithm.

Furthermore, several other color intensity schemes, including such HSV (Hue, Saturation, Value) as well as HSL (Hue, Saturation, Value) Saturation, Crispness, could be stated to produce the lane of traffic whatever is easier identified using incoming films. Filters have such a three-channel memory. As shown in Figure 2, H, S, V as well as H, S, L indicate the shades, S color saturation, V color change, and L color luminosity. When employing such places, you may come into the need to define a barrier so that roadways are more protected and feasible.

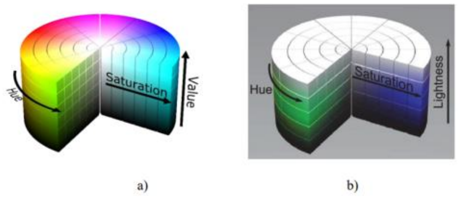


Figure 2. Color space [9]: a) HSV1, b) HSL2

B. CANNY EDGE DETECTOR

The standard Canny algorithm's basic stage is to flatten the picture. Canny calculated the whole first gradient of the Gaussian filter, which would be the closest approach to the ideal feature extraction activator. Select a suitable 1-d Gaussian filter to flatten the picture based mostly on the sequence as well as the panel, i.e., apply compression upon this digital image. Because the Fourier process obeys both distributive and algebraic laws, the Canny technique often employs a two-dimensional Gaussian filter (as illustrated in (1)) to flatten the picture and reduce outliers.

$$G(x,y) = exp[-(x2 + y2)/2\sigma 2]/2\pi\sigma 2 \qquad (1)$$

where σ is the Gauss filtration system variable that regulates the extent of picture softening. The amplitude and phase of the picture slope are calculated in the second phase. To evaluate the cost and orientation of pixel intensity, the classic Canny technique uses a restricted variety of 22 surrounding communities [9]. The accompanying formulae could be used to calculate the very first-degree fractional derivative's estimate in the X and Y-axis:

$$Ex[i,j] = (I[i+1,j] - I[i,j] + I[i+1,j+1] - I[i,j+1])/2 \qquad (2)$$

$$Ey[i,j] = (I[i,j+1] - I[i,j] + I[i+1,j+1] - I[i+1,j])/2 \qquad (3)$$

As a result, the picture gradients computation driver's template are:

$$GX = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \qquad (4)$$

$$GY = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \qquad (5)$$

Scale and direction of the inclination may be calculated. Image on a gradient scale is:

$$||M(i,j)|| = \sqrt{(〖E\_x\ [i,j]〗 ^2 + 〖E\_y\ [i,j]〗 ^2)} \quad (6)$$

The azimuth of the picture slope is:

$$(i,j) = arctan(Ey[i,j]/\ Ex[i,j]) \quad (7)$$

The Canny edge detector [8] works by calculating the slope of the image twice. Sides are therefore determined as feature points of the mean vector that are greater than certain criteria. Non-maximum suppression, or NMS, is the name given to this threshold highest qualities classification algorithm. The goal behind Canny's edge generator would have been to create an "ideal" activator within view that minimizes the likelihood of repeatedly identifying a line, the chance of refusing to accept a border, and the gap between the claimed side as well as the genuine corner. The very first two of all these requirements fix the challenges of recognition, that is, would the segmentation method locate an area if one exists? (and no other edges). The third component tackles the problem of navigation, or how correctly a side's location is conveyed. There is indeed an exchange among detection and recognition: the better the sensor, the worse the localization, and likewise.

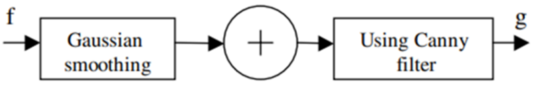


Fig. 3. Block diagram of image enhancement (sharpening and de-noising) using Canny edge detector

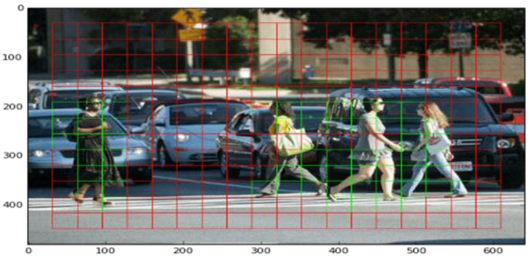
C. HOUGH TRANSFORM

The Hough transform is well acknowledged as a strong pattern analysis tool that produces reasonable outcomes often in the face of distortion and diffraction. The method's main disadvantages are its high material needs and computation cost. Illingworth and Kittler (Comput. Vision Graphics Image Process.44, 1988, 87-116) provide a very thorough assessment of known approaches up to and beyond 1988. The present project offers information on cutting-edge Hough approaches. This comprises analysis and comparison of current approaches, recent conceptual viewpoints, a large number of original methods, concurrent applications, and enhancements to mission equipment. It is important to make the distinction between work that tries to advance basic knowledge of the studies but is not always feasible and development that could be relevant in an industry situation.

D. YOLO (You Only Look Once)

1. *Residual blocks*

- Initially, the image is subdivided up into various grids. Each grid is S × S in size. How a grid is created from a source image is shown in the below diagram



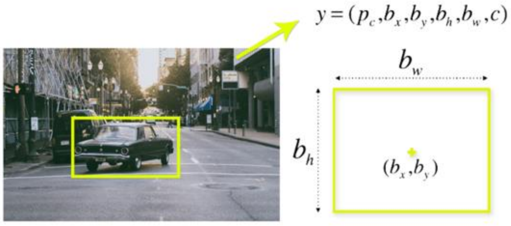
- Several grid cells of the same size are given in the image above. Each grid cell detects Objects that occur within the grid cells. If an item centre arrives within a certain grid cell, for example, that cell will be responsible for detecting it.

2. *Bounding box regression*

- An outline in a photo that draws attention to a certain object is called a bounding box.
- It has the following attributes:

1. Width (bw)
2. Height (bh)
3. Class (for example, person, car, traffic light, etc.)- This is represented by the letter c.
4. Bounding box centre (bx, by)

- A bounding box represents an object's height, width, centroid size, and class. To determine its height, width, and class, YOLO uses a single regression.



- To determine the height, breadth, and class of objects, the YOLO algorithm involves a single bounding box regression.
- The intersection over union (IOU) technique shows how object detection is done by overlapping boxes. IOU is implemented in YOLO to generate an output box that correctly surrounds the objects.

3. *Intersection Over Union (IOU)*

- In each grid cell the bounding boxes and their confidence ratings are forecasted. The IOU is 1, when the predicted and actual border boxes are identical, the IOU is 1. To remove bounding boxes that are not the same size as the main box this approach is used.

- A simple illustration of how an IOU works is shown below.

- The image below describes two bounding boxes, one in green and one in blue. The blue box represents the anticipated gift, whereas the green box represents the actual box. YOLO guarantees that the size of the two border boxes is the same.

4. *Merge of the 3 techniques*

- The below image shows a combination of three methodologies to produce the last detection result.

- To split the picture Grid cells are used. In every grid cell, B bounding boxes are projected, along with their reliability or best output score. The class probability of the cell is estimated to identify each item class.
- At least three various types of objects are identified, including a truck, a dog, and a bicycle. To construct all of the projections at the same time A single convolutional neural network is used.

  When IOU is utilized, the object's bounding box boxes are just like the object's actual boxes. This phenomenon eliminates any unnecessary bounding boxes that do not correspond to the attributes of the items (like height and width). The end detection will consist of separate bounding boxes that are tailored to the objects.

E. THE IMPLEMENTED LINE-DRAWING FUNCTION

The "Draw Lines()" vector illustration method is used to link the intersection points for every carriageway (left or right) to form a single straight edge that follows the real traffic inside the photos.

The horizontal lines are then generated using the Hough transform, as seen in Fig. 4, and thus are connected mostly by the "Draw Lines()" method to look such as the counterparts in Figs. 5 and 6.

The method does this by doing the proper procedures:

1) Orientation (Left or Right): Every Hough horizontal line is classified as belonging towards the left or right sections. This would be done depending on just the online sector's inclination. Unless the gradient is high somewhere between 0.4 and 1.0, the sector comes to that same left linear category; if this is low and between -0.4 and -1.0, the sector falls to that same right line group.

2) The durations and detects (pedestrian crossings only with x-axis) of all categorized Left as well as Right splices are computed and saved, together with respective gradients.

3) For every class (Left and Right), a line fitting approach is performed, with the gradient of every vertical line indicating whether the vertical line is excellent or poor (sound).

4) The Left and Right arcs may still be produced; although to decrease distortion, the data from the image sequences are used to create an Nth order screen (which has already been tested).

5) The FIR filter formula used in this version seems to be as follows:

$$Yk = a0 * Xk + a1 * Xk - 1 + a2 * Xk - 2 + a3 * Xk - 3 + \cdots .. an * Xk - n \qquad (8)$$

Table 1 shows the resulting FIR filter settings and factors utilized in this design.

Parameters	Value
Order	7
A_0	0.075
A_1	0.125
A_2	0.175
A_3	0.250
A_4	0.175
A_5	0.125
A_6	0.075

6) By using data of "the feature map" determined in step 5 of the process (V), the gradients and detects of the consequent endpoints are designed to characterize the left line in blue as well as the right line in red.

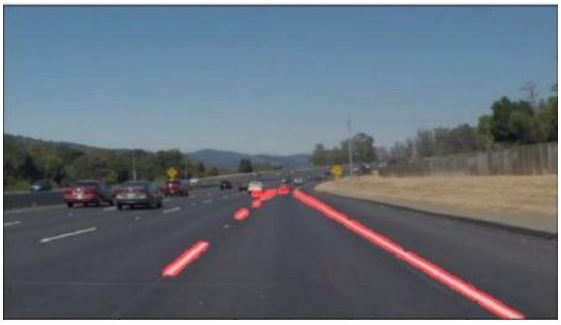


Fig. 4. The image with Hough line segments.

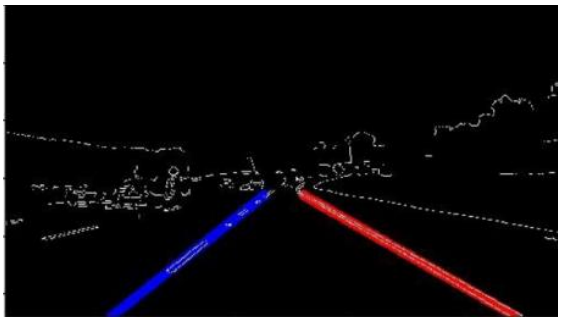
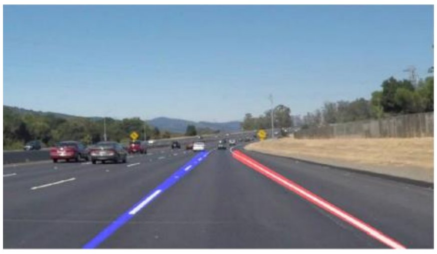


Fig. 5. The image after drawing lane lines (blue and red) by extrapolating the Hough lines segments.



The image after drawing lane lines (blue and red) by extrapolating the Hough lines segments.

III. VALIDATION & TESTING

The images of the indicated lane lines are forwarded to the upgraded yolov3 method circuit for retraining. The images employed in the tutorial mode are 416 By 416 px in dimension. Numerous essential indices variables within the method get constantly stored throughout the testing period. Figure 6 depicts the evolution of a standardized incidence functional L value. Figure 7 shows that perhaps the failure optimal value is around 1.8 somewhere at the start of retraining. The closure of such failure evaluation function drops as the number of classes grows, and once the computation time is around 20000, this same depreciation rate is 0.1 left as well as the right to obtain the intended impact.

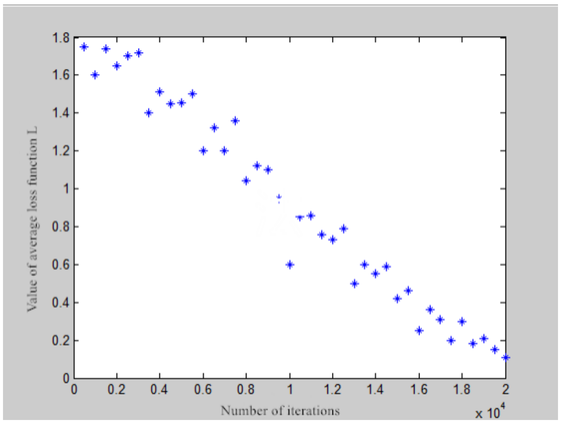


Figure 6. Variation Trend of average loss function value

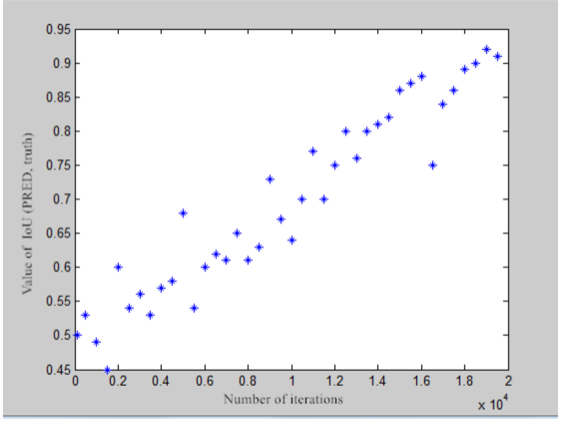
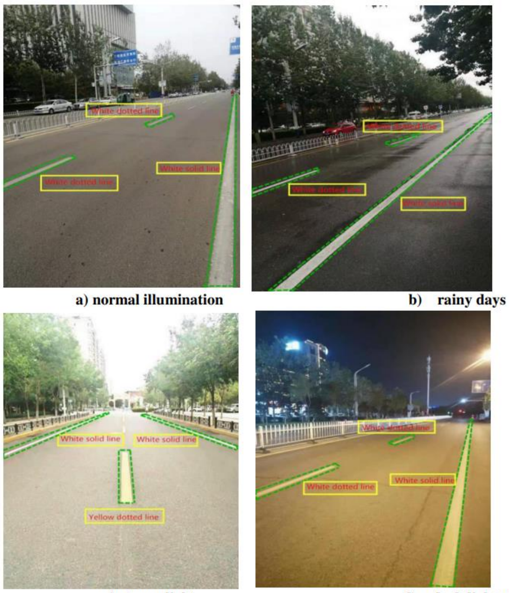


Figure 7. Change trend of matching degree value

Following the example education and experience, 100 straight boundary photographs are examined to see if the software can effectively recognize the lines on the road in the images. Figure 8 depicts the platform's lab results across various changes experienced.



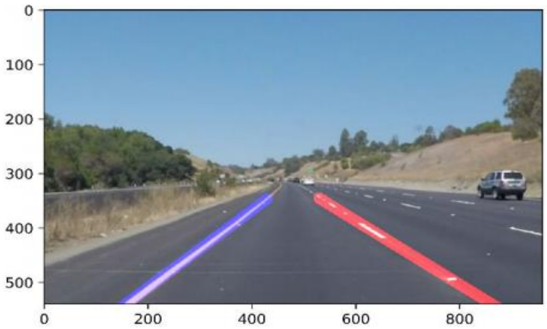
a) normal illumination b) rainy days

c) strong light d) dark light at night

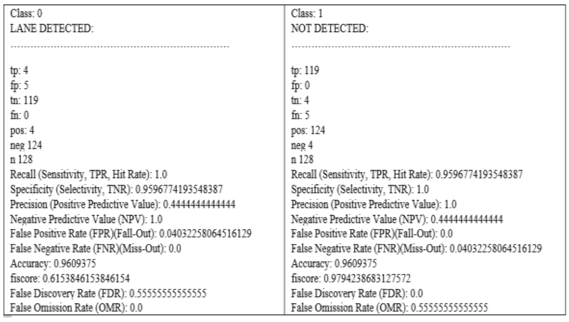
Figure8. System tests in different external conditions

IV. RESULT

Sample Name	No. of Frames	Total time Sec	Frame Per Sec
SolidYellowLeft Video	682	48.0	18.76
SolidWhiteRight Video	222	8.0	25.6
Challenge Video	251	23	10.9



The outcomes of videos labeled Clip:1 which carry linear road, Clip:2 which carry curved road, Clip:3 carry a mixture of straight and curved road, and Clip:4 which carry curved road, and so on are shown in the pictures below.



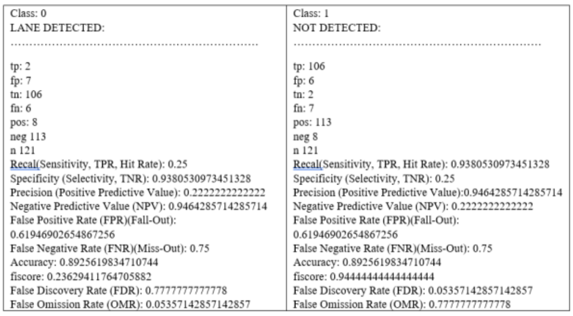
```
Class: 0
LANE DETECTED:
-----------------------------------------------------------------

tp: 4
fp: 5
tn: 119
fn: 0
pos: 4
neg 124
n 128
Recall (Sensitivity, TPR, Hit Rate): 1.0
Specificity (Selectivity, TNR): 0.9596774193548387
Precision (Positive Predictive Value): 0.4444444444444
Negative Predictive Value (NPV): 1.0
False Positive Rate (FPR)(Fall-Out): 0.04032258064516129
False Negative Rate (FNR)(Miss-Out): 0.0
Accuracy: 0.9609375
fiscore: 0.6153846153846154
False Discovery Rate (FDR): 0.5555555555555
False Omission Rate (OMR): 0.0
```

```
Class: 1
NOT DETECTED:
-----------------------------------------------------------------

tp: 119
fp: 0
tn: 4
fn: 5
pos: 124
neg 4
n 128
Recall (Sensitivity, TPR, Hit Rate): 0.9596774193548387
Specificity (Selectivity, TNR): 1.0
Precision (Positive Predictive Value): 1.0
Negative Predictive Value (NPV): 0.4444444444444
False Positive Rate (FPR)(Fall-Out): 0.0
False Negative Rate (FNR)(Miss-Out): 0.04032258064516129
Accuracy: 0.9609375
fiscore: 0.9794238683127572
False Discovery Rate (FDR): 0.0
False Omission Rate (OMR): 0.5555555555555
```

Fig.9. Result of CLIP:1



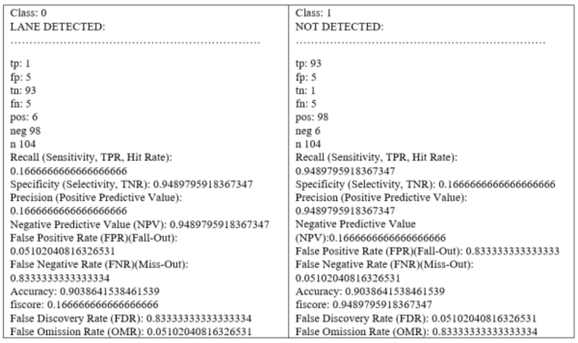
```
Class: 0
LANE DETECTED:
.................................................................

tp: 2
fp: 7
tn: 106
fn: 6
pos: 8
neg 113
n 121
Recal(Sensitivity, TPR, Hit Rate): 0.25
Specificity (Selectivity, TNR): 0.9380530973451328
Precision (Positive Predictive Value): 0.2222222222222
Negative Predictive Value (NPV): 0.9464285714285714
False Positive Rate (FPR)(Fall-Out):
0.61946902654867256
False Negative Rate (FNR)(Miss-Out): 0.75
Accuracy: 0.8925619834710744
fiscore: 0.23629411764705882
False Discovery Rate (FDR): 0.7777777777778
False Omission Rate (OMR): 0.05357142857142857
```

```
Class: 1
NOT DETECTED:
.................................................................

tp: 106
fp: 6
tn: 2
fn: 7
pos: 113
neg 8
n 121
Recal(Sensitivity, TPR, Hit Rate): 0.9380530973451328
Specificity (Selectivity, TNR): 0.25
Precision (Positive Predictive Value):0.9464285714285714
Negative Predictive Value (NPV): 0.2222222222222
False Positive Rate (FPR)(Fall-Out):
0.61946902654867256
False Negative Rate (FNR)(Miss-Out): 0.75
Accuracy: 0.8925619834710744
fiscore: 0.9444444444444444
False Discovery Rate (FDR): 0.05357142857142857
False Omission Rate (OMR): 0.7777777777778
```

Fig.10. Result of CLIP:2



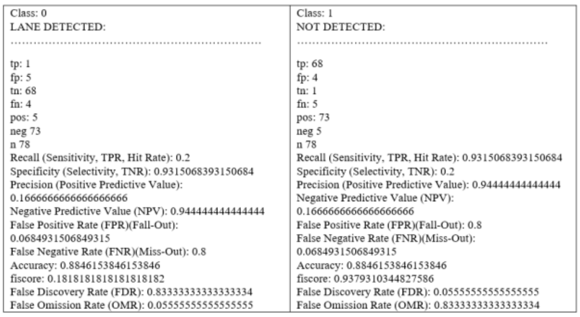
```
Class: 0
LANE DETECTED:
.................................................................

tp: 1
fp: 5
tn: 93
fn: 5
pos: 6
neg 98
n 104
Recall (Sensitivity, TPR, Hit Rate):
0.16666666666666666
Specificity (Selectivity, TNR): 0.9489795918367347
Precision (Positive Predictive Value):
0.16666666666666666
Negative Predictive Value (NPV): 0.9489795918367347
False Positive Rate (FPR)(Fall-Out):
0.05102040816326531
False Negative Rate (FNR)(Miss-Out):
0.8333333333333334
Accuracy: 0.9038641538461539
fiscore: 0.16666666666666666
False Discovery Rate (FDR): 0.8333333333333334
False Omission Rate (OMR): 0.05102040816326531
```

```
Class: 1
NOT DETECTED:
.................................................................

tp: 93
fp: 5
tn: 1
fn: 5
pos: 98
neg 6
n 104
Recall (Sensitivity, TPR, Hit Rate):
0.9489795918367347
Specificity (Selectivity, TNR): 0.16666666666666666
Precision (Positive Predictive Value):
0.9489795918367347
Negative Predictive Value
(NPV):0.16666666666666666
False Positive Rate (FPR)(Fall-Out): 0.833333333333333
False Negative Rate (FNR)(Miss-Out):
0.05102040816326531
Accuracy: 0.9038641538461539
fiscore: 0.9489795918367347
False Discovery Rate (FDR): 0.05102040816326531
False Omission Rate (OMR): 0.8333333333333334
```

Fig.11. Result of CLIP:3



```
Class: 0
LANE DETECTED:
.................................................................

tp: 1
fp: 5
tn: 68
fn: 4
pos: 5
neg 73
n 78
Recall (Sensitivity, TPR, Hit Rate): 0.2
Specificity (Selectivity, TNR): 0.9315068393150684
Precision (Positive Predictive Value):
0.16666666666666666
Negative Predictive Value (NPV): 0.944444444444444
False Positive Rate (FPR)(Fall-Out):
0.0684931506849315
False Negative Rate (FNR)(Miss-Out): 0.8
Accuracy: 0.8846153846153846
fiscore: 0.18181818181818182
False Discovery Rate (FDR): 0.8333333333333334
False Omission Rate (OMR): 0.05555555555555555
```

```
Class: 1
NOT DETECTED:
.................................................................

tp: 68
fp: 4
tn: 1
fn: 5
pos: 73
neg 5
n 78
Recall (Sensitivity, TPR, Hit Rate): 0.9315068393150684
Specificity (Selectivity, TNR): 0.2
Precision (Positive Predictive Value): 0.94444444444444
Negative Predictive Value (NPV):
0.16666666666666666
False Positive Rate (FPR)(Fall-Out): 0.8
False Negative Rate (FNR)(Miss-Out):
0.0684931506849315
Accuracy: 0.8846153846153846
fiscore: 0.9379310344827586
False Discovery Rate (FDR): 0.05555555555555555
False Omission Rate (OMR): 0.8333333333333334
```

Fig.12. Result of CLIP:4

An elimination testing was conducted throughout this research to validate the efficiency of the suggested system M-YOLO. Table 3 analyzes the AP of relinking the bounding box, substituting the communication infrastructure with MobileNet v2, and various optimization algorithms based on YOLO v3. Likewise, Table 3 analyzes the frames per second of YOLO v3 and M-YOLO.

Whenever the peg bars are updated, the overall result of traffic monitoring is considerably enhanced when compared to YOLO v3. The AP rose from 76.51 percent to 87.33 percent. Furthermore, the AP was somewhat lowered once the core system was upgraded with MobileNet v2 from DarkNet53, although the reduction was extremely tiny. Lastly, the DIoU error rate, CIoU stochastic gradient descent, and EIoU regression model are utilized to retrain the M-YOLO system as such position error function. The overall accuracy rises progressively. M-final YOLO's precision value is 94.96 percent. Furthermore, M-AP YOLO's is 18.45 percent greater than YOLO v3, yet three times faster.

Model	Anchor Boxes	MobileNet v2	loss	AP%	FPS
YOLO v3			MSE	76.51	3
	✓		MSE	87.33	/
	✓	✓	MSE	86.95	/
M-YOLO	✓	✓	DIoU	88.57	/
	✓	✓	CI0U	93.87	/
	✓	✓	EIoU	94.96	10

V. CONCLUSION

The assessment of travel demand factors is indeed a critical component of roadway construction and management, and also signalized intersections measures for current facilities. Various methods, including those of static location monitoring and "sensor automobile information" devices, could be used to gather traffic information. Furthermore, Wardrop's moving observer method (MOM) may be employed as a traffic information acquisition approach. Although as usually recognized, artificial intelligence (AI) with deep learning (DL) are widely employed in a wide range of real scenarios, notably automobile classification and tracking. In order to address the issue that standard border tracking algorithms need not take recognition rate and productivity is affected into account, this work proposes a protection system adaptive routing yolov3 approach. Below are the primary enhancements:

1. Based on the features of traffic line pictures' uneven transverse and longitudinal dispersion frequency, it really is recommended to partition the pictures across s * 2S squares to increase height recognition intensity.

2. The sensor size has been limited to four sensor scales: 13 * 13, 26 * 26, 52 * 52104 * 104, which is more suited for detecting tiny objects also including lane dividers.

3. The basic yolov3 device's fully connected layers are reduced from 53 to 49 tiers, simplifying the system and improving system efficiency.

4. Parameters including such closest cluster location and transfer functions have been modified, making them more meaningful and suited for traffic line recognition environments. The real test results reveal that perhaps the revised method has high detection accuracy while identifying level roads, however, the identification is easily influenced whenever the roadways have considerable gradients. As a result, the next report will concentrate on fixing the situation of lane line recognition in huge gradient sceneries.

REFERENCES

[1] Zhang, Xiang, Wei Yang, Xiaolin Tang, and Jie Liu. "A fast learning method for accurate and robust lane detection using two-stage feature extraction with YOLO v3." Sensors 18, no. 12 (2018): 4308.

[2] Huang, Shan, Ye He, and Xiao-an Chen. "M-YOLO: A Nighttime Vehicle Detection Method Combining Mobilenet v2 and YOLO v3." In Journal of Physics: Conference Series, vol. 1883, no. 1, p. 012094. IOP Publishing, 2021.

[3] Illingworth, John, and Josef Kittler. "A survey of the Hough transform." Computer vision, graphics, and image processing 44, no. 1 (1988): 87-116.

[4] Mukhopadhyay, Priyanka, and Bidyut B. Chaudhuri. "A survey of Hough Transform." Pattern Recognition 48, no. 3 (2015): 993-1010.

[5] Farag, Wael, and Zakaria Saleh. "Road lane-lines detection in real-time for advanced driving assistance systems." In 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), pp. 1-8. IEEE, 2018.

[6] Farag, Wael. "Real-time detection of road lane-lines for autonomous driving." Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science) 13, no. 2 (2020): 265-274.

[7] Farag, Wael. "A comprehensive real-time road-lanes tracking technique for autonomous driving." International Journal of Computing and Digital Systems 9, no. 03 (2020).

[8] Šimunović, Bruno. "Detekcija prometne trake." PhD diss., Josip Juraj Strossmayer University of Osijek. Faculty of Electrical Engineering, Computer Science and Information Technology Osijek. Department of Software Engineering. Chair of Visual Computing, 2021.

[9] Ji, Gaoqing, and Yunchang Zheng. "Lane Line Detection System Based on Improved Yolo V3 Algorithm." (2021).

[10] Gothankar, Nikhil, Chandra Kambhamettu, and Paul Moser. "Circular hough transform assisted cnn based vehicle axle detection and classification." In 2019 4th International Conference on Intelligent Transportation Engineering (ICITE), pp. 217-221. IEEE, 2019.

[11] [11] Reddy, D. Ramesh, Chandravathi Chella, K. Bala Ravi Teja, Heera Rose Baby, and Prakash Kodali. "Autonomous Vehicle Based on Deep Q-Learning and YOLOv3 with Data Augmentation." In 2021 International Conference on Communication, Control and Information Sciences (ICCISc), vol. 1, pp. 1-7. IEEE, 2021.