
ISSN: 2321-2152
IJMECE
International Journal of modern
electronics and communication engineering
E-Mail
editor.ijmece@gmail.com
editor@ijmece.com
www.ijmece.com

ISSN2321-2152
www.ijmece .com
Vol 11, Issue.1, 2023

# Next-Generation Wrong Way Detection: Drive Alert Plus

Mrs. K. Janani, Mrs. A. Baranishri, Mrs. K. Lavanya, Dr. J. Thilagavathi, Ms. R. Roopa
Professor [4] Assistant Professor [1,2,3,5]
kjanani@actechnology.in, baranishri.a@actechnology.in, klavanya@actechnology.in,
thilagavathi@actechnology.in, roopa.r@actechnology.in
Department of CS&BS, Arjun College of Technology, Thamaraikulam, Coimbatore-Pollachi Highway, Coimbatore, Tamilnadu-642 120

***Abstract*—** The detection of wrong-way driving is an important step towards alleviating traffic congestion and road accidents on a global scale. We provide a method that can automatically detect cars going the wrong way from road surveillance data using real-time traffic control systems and inexpensive cameras. Our solution is designed to work in three steps. The first stage involves using the extremely precise You Only Look Once (YOLO) algorithm to identify automobiles inside video frames. Then, within the defined area, it follows every vehicle using the centroid tracking technique. Finally, cars travelling the wrong direction are detected by the system. In our trials, we used a variety of traffic movies to show that the system can identify cars going the wrong way in a variety of lighting and weather scenarios. Our suggested method is a powerful instrument for improving traffic flow and road safety due to its simplicity and ease of deployment.

***Keywords—wrong-way driving, Centroid Tracking, yolo***

## I.INTRODUCTION

Road accidents are a prevalent issue in densely populated countries like Bangladesh, with alarming statistics of 4,702 accidents and 5,227 deaths in 2019, including children and students [1]. The inadequacy of road capacity for the growing number of vehicles exacerbates the problem, as drivers often disregard traffic rules and resort to wrong-side driving when facing red traffic signals. This behavior not only disrupts traffic flow but also escalates the risk of head-on collisions, claiming around 355 lives annually in the United States alone [3]. Fig. 1 illustrates a common sight of traffic congestion caused by wrong-way driving, a recurring scenario in many Bangladeshi cities. Therefore, curbing wrong-way driving is imperative to enhance road safety, necessitating strict law enforcement against rule-breaking drivers.

ISSN2321-2152
www.ijmece .com
Vol 11, Issue.1, 2023

Fig. 1: Heavy traffic jams due to wrong-way driving [4]

Intelligent Transportation Systems (ITS) have long employed computer vision technology to address traffic- related issues. Traffic surveillance systems analyze on-road camera footage to identify the causes of road accidents, traffic congestion, rule violations, vehicle counting, and speed monitoring. In the past, the use of this technology was hindered by expensive computational requirements and data limitations. However, recent advancements in machine learning, deep learning algorithms, the availability of powerful Graphics Processing Units (GPUs), and affordable cameras have significantly improved the efficiency of the entire system.

The system we present here operates in three stages. In the initial stage, the YOLO object detection algorithm identifies every vehicle within the video frame, generating a bounding box for each detected vehicle. Subsequently, these bounding boxes are input into the centroid-based moving object tracking algorithm, which independently tracks each vehicle within a specified Region of Interest (ROI). Finally, the vehicle's direction is determined by assessing the height of its centroid in each frame, enabling the system to detect if it is moving in the wrong direction. In cases of wrong-way driving, the system captures an image of the offending vehicle.

## II.LITERATURE REVIEW

Computer vision technology plays a crucial role in intelligent traffic monitoring systems, particularly in the development of methods for detecting wrong-way vehicles. Existing approaches fall into two main categories: sensor-based and motion pattern-based methods.

Sensor-based methods [5] rely on magnetic sensors to determine vehicle direction. These sensors detect disturbances in the Earth's magnetic field caused by passing vehicles, providing a signal that indicates vehicle direction. However, this approach can yield unsatisfactory results because the magnetic field can also be affected by factors unrelated to vehicle direction, leading to false detections.

On the other hand, motion pattern-based methods utilize optical flow measurement. For instance, a method proposed in calculates optical flow to infer vehicle direction and compares it with the expected lane direction [6]. Nevertheless, this approach is susceptible to occlusion issues, which can lead to inaccurate results. To address this, improved optical flow estimation techniques have

ISSN2321-2152
www.ijmece .com
Vol 11, Issue.1, 2023

been employed, combined with background subtraction and the Lucas-Kanade method [7]. However, this system may struggle to perform consistently in varying lighting conditions, as the background subtraction method is sensitive to vehicle shadows.

In summary, while various methods have been developed for wrong-way vehicle detection, they each come with their limitations, whether related to sensor-based inaccuracies or motion pattern-based challenges, particularly under changing lighting conditions.

## III.METHODOLOGY

This section describes the details of the whole proposed system. The details of each stage will be described consecutively in the following subsections.sections

### *A.YOLO*

In the realm of vehicle detection using bounding boxes, two prominent real-time methods stand out: Mask-RCNN [8] and YOLO [9]. While Mask-RCNN employs a technique called selective search for creating bounding boxes, YOLO offers superior speed and is considered the state-of-the-art choice for real-time applications. Our paper utilizes the most recent iteration, YOLOv3 [10].

What sets YOLO apart is its ability to process an image just once and simultaneously derive object class probabilities and bounding boxes, making it exceptionally fast compared to other object detection algorithms. The core concept involves dividing the image into an SxS grid, with each cell potentially containing bounding boxes [9]. If there are C distinct classes, the total number of bounding boxes becomes SxSxB, and each box is associated with (5+C) attributes. Four attributes pertain to box coordinates, while one signifies the confidence score that the box contains one of the C classes. To manage object redundancy, we employ a confidence score threshold of 0.5 in our system, only displaying boxes with scores surpassing this threshold and their corresponding class names. These operations are all executed in a single pass through the network. To address the issue of multiple detections of the same object, we implement a technique called non-maximum suppression. This process selects the box with the highest confidence score first and subsequently removes boxes with overlaps exceeding a threshold value with the previously selected box, ensuring only one box per object.

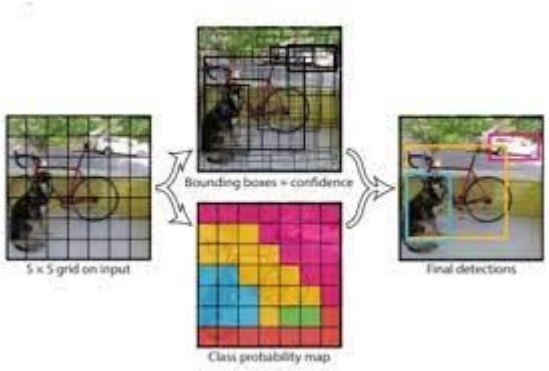



Fig. 2: YOLO divides the image into the SxS grid and predicts B bounding box for each cell [9]

In YOLOv3, the authors introduce a deeper convolutional neural network (CNN) with 53 convolutional layers to extract image features.

ISSN2321-2152
www.ijmece .com
Vol 11, Issue.1, 2023

Each convolutional layer benefits from batch normalization and Leaky ReLU activation functions. These layers efficiently extract features from the input image.

YOLOv3 is trained on the COCO dataset, encompassing 80 distinct classes. However, for our system, we focus exclusively on various vehicle classes, such as motorbikes, buses, trucks, and cars. Notably, YOLOv3 outperforms its predecessors in accurately detecting small objects, making it a valuable choice for our application.

### *B.Virtual Region of Interest*

We want to track the vehicles in a certain region of the video frame. That is why we selected a rectangular area in the frame which is shown in Fig. 3 in red color. When the vehicle is in the region, it will be tracked. When the vehicle is out of this region, it will be removed from the tracked vehicle list. As this region depends on the camera view, it must be done manually. The region should have a reasonable physical distance or there will be a problem to track the large vehicles.

### *C.Vehicle Tracking*

To effectively track each vehicle, our system employs the centroid tracking algorithm. This algorithm takes the bounding boxes generated by YOLO as input. Here's how the process works:

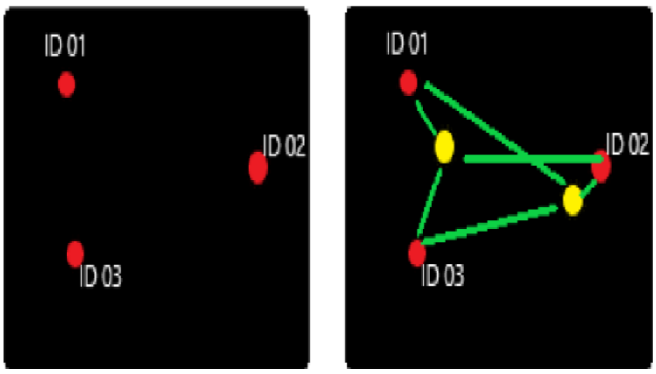
- Bounding boxes are first generated using YOLO, as illustrated in Fig. 4a.

Fig. 3: The region of interest for vehicle tracking

- When the center of each vehicle, which corresponds to the center of its bounding box, enters the predefined region of interest, it is assigned a unique identification number, as shown in Fig. 4a.
- In the subsequent frame, the centers of all objects either move to new positions or remain stationary, as seen in Fig. 4b.
- The centroid tracking algorithm operates on the assumption that objects will exhibit minimal movement between consecutive frames. To identify previously tracked objects, the algorithm computes the Euclidean distance between the new centroids (yellow) and the old centroids (red), as expressed in Equation (1). Euclidean distance, d = √((xi - xj)^2 + (yi - yj)^2)
- If the current frame contains fewer objects than the previous frame, indicating that one or more objects have left the region of interest (as shown in Fig. 4b), the system removes their identification numbers from the list of tracked vehicles.

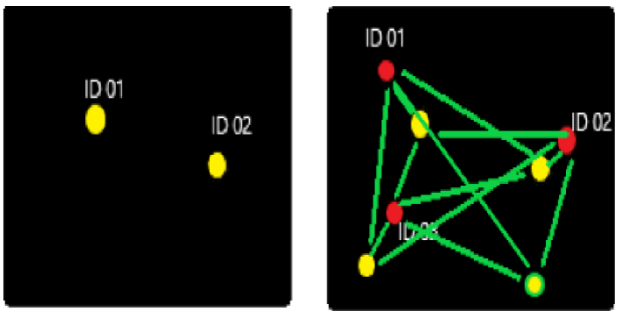
ISSN2321-2152
www.ijmece .com
Vol 11, Issue.1, 2023

- Conversely, if the current frame contains more objects than the previous frame (as shown in Fig. 5b), the algorithm updates and reassigns new centroids to existing objects while assigning new identity numbers to the newly detected objects.




(a) (b)

Fig. 4: (a) The first frame (b) second frame




(a) (b)

Fig. 5: (a) updated frame (b) when a new frame has more objects

This approach allows our system to track each vehicle independently, as demonstrated in Fig. 6. The figure showcases an example where a CNG vehicle is successfully tracked within the region of interest, receiving its unique identity number. On the right side of the image, a pedestrian detected by YOLO is not tracked because our system focuses exclusively on vehicle classes and disregards other objects.

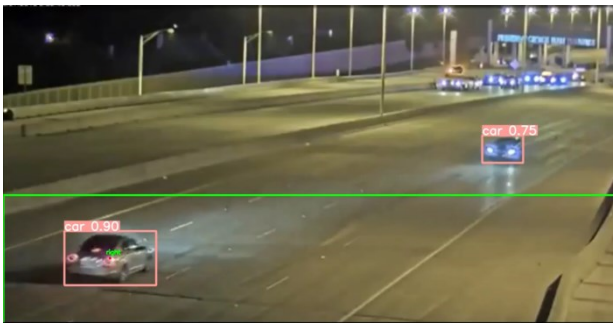



Fig. 6: A car vehicle is tracked

### *A. Wrong-way vehicle detection*

The final step of our proposed system is to detect the wrong-way vehicle. Our system already can track each vehicle that is in our specified region of interest. When the vehicle is registered first and given an identity number, the height of the centroid H1 is computed and stored in a file corresponding to the identity number.

In the next frame, the height of the centroid H2 is computed and stored in another file along with its identity number. This H2 will be updated in each consecutive frame. If the vehicles move, the H1 and H2 of a vehicle will not be equal. By comparing these two heights, our system will predict the vehicle's direction. In our system, we defined that if the vehicle moves away from the camera, it will be detected as a wrong-way

vehicle. So, if H1 < H2 then, the vehicle is coming towards the camera and is in the right way. If otherwise, our system will detect it as a wrong-way vehicle. The opposite can also be defined just by changing the condition. After the detection of such a vehicle, an image of the frame will be captured automatically for further inspection. The flowchart of the whole system is shown in Fig. 7

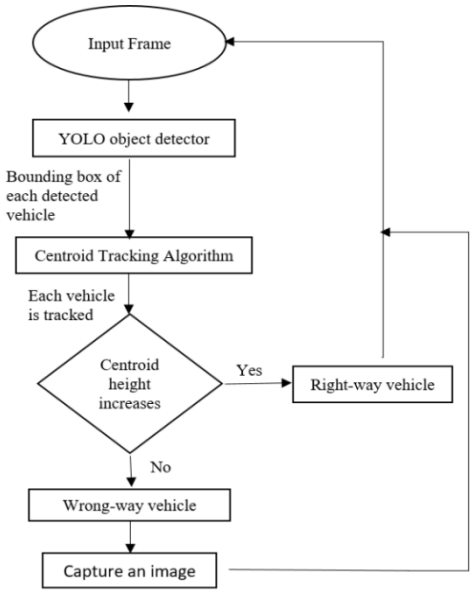



Fig. 7: The flowchart of the whole system

## IV.IMPLEMENTATION

1. *Setup Environment*: Install necessary libraries, including OpenCV, PyTorch, and YOLOv5.
2. *Load YOLOv5 Model*: Load a pre-trained YOLOv5 model (e.g., 'yolov5s').
3. *Capture Video:* Initialize a video capture object to read frames from a video source.
4. *Object Detection*: Use the YOLOv5 model to perform object detection on the frame. Extract information about detected objects (e.g., bounding boxes, class labels, confidence scores).
5. *Rendering Results*: Overlay the detection results (bounding boxes and labels) on the frame. You can use OpenCV to draw boxes and labels on the frame.
6. *Display Frame:* Show the frame with detection results in a window.
7. *Exit Condition*: Check for an exit command (e.g., pressing 'q' key). If the exit command is detected, break out of the loop.
8. *Completion:* Your object detection project is now complete and can detect objects in the specified video source.

## V. RESULTS

To evaluate our system, we captured three videos in Chittagong city, Bangladesh, each featuring a single wrong-way vehicle. These videos were recorded from the roadside and had a resolution of 1280 x 720 pixels. In Figure 8, we can observe that our system successfully detected the wrong-way vehicle in each of the three videos. Consequently, the accuracy of our system is nearly 100%. It's worth noting that other related systems [5] [6] [7] do not rely on deep learning algorithms. Thus, it's not feasible to directly compare our system with them using the same dataset and experimental setup. Furthermore, the

authors of those systems did not provide specific accuracy metrics, making a direct comparison inappropriate.

Our system was designed to detect wrong-way vehicles on one side of the road, with the camera focused accordingly. In our future work, we plan to extend the system to operate on both sides of the road using a single camera. In this setup, the camera will be positioned in the middle of the road to monitor both directions.

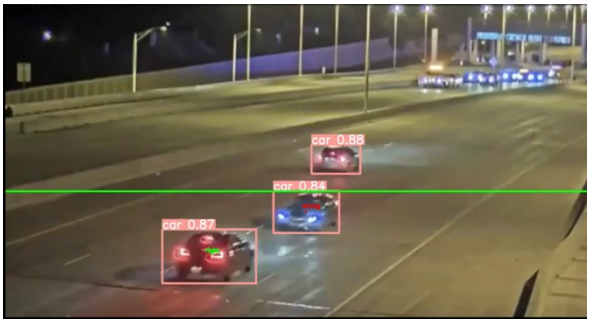



Fig. 8: A car is driving on the wrong way

The system was implemented in the Python language using the OpenCV deep learning library. Due to the computational intensity of the YOLO algorithm, the system achieves a frame rate of 2.5 frames per second on an AMD Ryzen 7 CPU running on the Windows platform. To optimize processing speed, we sampled one frame and skipped the next four frames. It's worth noting that utilizing an Nvidia GPU on a Linux platform can significantly enhance processing speed, potentially achieving several times faster performance.

## VI.REFERENCES

[1] "Bangladesh road accidents in 2019 statistics by Nirapad Sarak Chai," The Daily Star,05-Jan-2020. [Online]. Available: https://www.thedailystar.net/frontpage/bangladesh-road-accidents-in-2019-stats-by-nirapad-sarak-chai-1849588 [Accessed Feb. 20, 2020].

[2] "Traffic Jam in Dhaka: 'Driving on the wrong side a disease', says Obaidul."The Daily Star,29-May-2018. [Online]. Available: https://www.thedailystar.net/country/dhaka-traffic-jam-driving-wrong-side-disease-obaidul-quader-1583323. [Accessed Feb. 21, 2020].

[3] M. Pour-rouholamin and J. Shaw, "Current Practices of Safety Counter Measures for Wrong-Way Driving," TRB 94th Annu. Meet. Compend. Pap., no. January, pp. 1–14, 2015.

[4] "ways to avoid traffic congestion Archives - GoMetro Transport App." [Online]. Available: http://www.getgometro.com/tag/ways-to-avoid-traffic-congestion/. [Accessed: 25-Feb-2020].

[5] M. J. Caruso and L. S. Withanawasam, "Vehicle Detection and Compass Applications using AMR Magnetic Sensors," Sensors Expo, vol. 477, p. 39, 1999.

[6] "WRONG WAY DRIVERS DETECTION BASED ON OPTICAL FLOW Gonc Institute for System and Robotics Dep . of Electrical Engineering and Computers University of

Coimbra - Portugal," no. 1, pp. 141–144, 2007.

[7] S. V.-U. Ha, H.-H. Nguyen, H. M. Tran, and P. Ho-Thanh, "Improved optical flow estimation in wrong-way vehicle detection," J. Inf. Assur. Secur., vol. 9, no. 5, pp. 165–169, 2014.

[8] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-CNN," in 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.

[10] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767,2018.