International Journal on Advanced Science Engineering Information Technology

Vol.14 (2024) No. 6
ISSN: 2088-5334

Wrong-Way Driving Detection for Enhanced Road Safety using Computer Vision and Machine Learning Techniques

Ryan Mo Xian Wee [a], Tee Connie [a,*], Michael Kah Ong Goh [a]

[a] *Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, Melaka, Malaysia*
*Corresponding author: *tee.connie@mmu.edu.my*

***Abstract*—This paper describes a real-time vehicle detection and tracking system using deep learning techniques together with computer vision. A system based on the YOLOv4 model is developed using the Lukas-Kanade optical flow technique to locate the vehicles considering real-life traffic situations that involve atmospheric disturbances. The subsystem first assesses the trajectory of vehicles concerning the intended direction of traffic flow. Any deviation from this norm triggers a signal that will enable traffic managers to intervene in time to prevent the possibility of traffic congestion. The initial findings confirmed the system's evident ability to reduce the incidence of wrong-way driving, such that its enforcement was concentrated on specific highway sections, targeting high average accuracy rates and reducing the overall risk of base rate accidents. This paper tackles the issues related to the existing problems of surveillance systems, such as the quick detection of unusual traffic patterns and the ability to respond to critical situations quickly. Furthermore, the combination of cutting-edge AI technologies represents a practical and easy approach to implementing intelligent transport systems that are safe, cost-effective, efficient, and novel. Future work will extend the system's adaptability to various traffic environments, refine its performance under challenging conditions, and explore advanced deep-learning models. This research ultimately contributes to smarter traffic monitoring technologies, fostering safer travel environments, mitigating road hazards, and reducing congestion for a more efficient and secure transportation ecosystem.**

***Keywords*—YOLOv4; deep learning; wrong-way driving; traffic surveillance; optical flow; Lucas-Kanade; computer vision.**

Manuscript received 12 Nov. 2023; revised 20 Apr. 2024; accepted 14 Sep. 2024. Date of publication 31 Dec. 2024.

I. INTRODUCTION

Following recent trends, it is evident that there is an increase in the number of vehicles on the road. This reflects the need for traffic surveillance systems in the society systems in our society [1], [2], [3]. These systems assist in enhancing the safety of roads, mitigating traffic jams, and increasing transport systems' effectiveness. The traffic monitoring systems employ the use of CCTV cameras, radar detectors, and other sensors in the collection and analysis of the movement of traffic, speed of vehicles, and types of vehicles on the road [4], [5], [6], [7]Traffic congestion data assists in predicting locations with traffic jams and potential sites for accidents, among other things that can slow traffic flow.

Traffic management systems need to work in real-time to be effective. Rather than just observing, the systems play an important role in spotting and dealing with dangerous or careless driving to allow police to take necessary actions [8]. These systems are also crucial for monitoring highways where accidents, vehicle breakdowns, and debris on the road can be quickly detected.

The information from the traffic management systems is crucial for alerting emergency services and road maintenance teams. This helps the traffic authorities to respond quickly to prevent more problems and get traffic back to normal quickly. These measures are essential for keeping the roads safe and efficient [9], [10]. However, many systems still have trouble detecting car accidents in real-time [11]. This issue often goes unnoticed because these systems cannot spot such incidents as they happen, raising the risk of accidents and deaths.

This paper explores the development of a robust road surveillance method using state-of-the-art computer vision techniques. The goal is to improve traffic monitoring and safety. The paper also highlights the problems faced by the current systems in detecting and responding to car accidents. Even with new technologies, these systems struggle to identify and manage emergencies quickly. This leads to slow emergency responses and more traffic jams. Additionally, these systems often fail to detect driving on the wrong way swiftly, which is a significant problem for police and road maintenance workers. Accidents not handled quickly can lead to serious injuries, deaths, and long traffic delays.

The core of this study involves developing a system utilizing algorithms based on deep learning, such as YOLOv4, and computer vision methods like optical flow. The YOLOv4 algorithm [12], [13], [14], [15], [16] Initially, it identifies vehicles on the road. After that, optical flow is used to analyze the direction of the vehicles' movement. This paper's major contribution is the creation of a robust system that leverages CCTV footage to predict potential car accidents by integrating vehicle detection with motion tracking, which enables the detection of wrong-way drivers.

However, the system currently detects wrong-way driving in only a specific segment of the road or highway. The expected traffic flow direction is predetermined, and any vehicle moving contrary to this direction is flagged as a wrong-way driver. When a vehicle's optical flow markedly deviates from the expected direction, the anomaly is detected and a real-time alert is triggered to prompt immediate action from relevant authorities or traffic monitoring systems.

A. Conventional Methods for Traffic Anomaly Detection

Ha et al. [17] came out with a system that uses an improved optical flow estimation to detect Vehicle driving in the wrong way in the system uses a new algorithm to solve incompatibility issues such as the inconsistency flow of motion when an object is moving when it is only meant to move in one motion of direction. There are a total of three steps for the system to work. First, a background subtraction process is used to obtain the binary foreground. The following step is by taking the binary foreground as a mask to separate moving vehicles from the actual frame picture to generate a color foreground, which can act as an input for the optical flow to process and get the traffic flow. The last step in the improved algorithm is to identify all the object movements and the main motion direction of each object. During the background subtraction process, every pixel is represented by a weighted Gaussian distribution; different colors represent different Gaussians. In foreground detection, the value of every pixel will be compared with every background component. The system also uses a pre-processing step so that it can differentiate the original frames to separate the color foreground into separate pictures.

Furthermore, to improve the traffic flow, the system requires four algorithm steps. The first step is to use a 16-bin motion direction histogram to act as a cardinal direction class for every bin. The second step is to identify the object's movement by using contour detection for the flow of the images. The third step is calculating the histogram and identifying the bin representing the major direction. Lastly, the system will calculate the means of RGB value, and the value will represent the primary motion direction, which will subsequently be applied to all pixels in the blob. Lastly, to detect a vehicle driving the wrong way a lanes' motion orientation will be applied. The color traffic flow is produced in each new frame then pixel by pixel compared with the lane image. The pixels with a greater than a predetermined disparity between the flow direction in the current frame and the corresponding direction in the lane motion orientation are indicated to be in the wrong direction.

Tan et al. [18] proposed a fast anomaly detection algorithm using a method called sparse optical flow. Sparse optical flow is different from any other optical flow method. It has an improved computation for the foreground mask and forward and backward filtering. To aid in detecting stationary or slow-moving objects, the feature vector includes a foreground channel. A Histogram of Optical Flow (HOF) has been implemented into sparse optical flow. This feature is used to aggregate in the spatial block. This feature can obtain better results in detecting stationary vehicles and low-speed vehicles. This algorithm has been evaluated by using a dataset from UCSD Ped1 Dataset. The algorithm used by [19] has a better performance compared to the state-of-the-art methods, with a false positive rate of less than 24%. However, the resilience and effectiveness of the sparse optical flow calculation are enhanced compared to the conventional optical flow method, which can run much faster and more efficiently.

Later on, Anandkumar et al. [20] deployed a machine learning method called Principal Component Analysis (PCA) to analyze road traffic accident data and heavy traffic monitoring data in cities. The reason for using the PCA method is to predict traffic data and examine data trends because it is the easiest method to break down high-dimensional data into primary components such as principal components; from this principal component, we can clearly identify the patterns of the high-dimensional data. To perform PCA analysis, we must pre-process the data, scale the data, and calculate the covariance or correlation matrix. After performing all this activity, we can proceed to PCA analysis by choosing the principal components. After that, a new computing dataset will be generated. At the beginning of the analysis, the datasets are separated into categories: 80% will be categorized as a training data set, and 20% will be categorized as a testing dataset. In the article, the PCA analysis car counts forecast error rate for the provided data is about 32%. Moreover, multiple techniques are used to analyze different datasets, including the dense urban traffic surveillance and vehicle count datasets. We can now clearly see the necessity for controlling and forecasting traffic situations thanks to the integrated approach to heavy urban traffic analysis and principal component analysis on vehicle traffic data.

Besides, Arun et al. [20] proposed a way to detect vehicles changing lanes on the road illegally on urban highways by estimating the region of the road, and each vehicle have been tracked down so that the system can detect lane crossings vehicles based on the distance between the lines of the lanes and the vehicle center. In this article, the system is being tested by using the selected video sequences of the GRAM-RTM dataset. First and foremost, to detect illegal lane-crossing vehicles captured by a camera on the road the first step is to detect the background by using Gaussian Mixture Model. This model can identify the lane region of the road. The following step is to track down every vehicle by using a model-based tracking technique. When a vehicle's tracking window center is within lanes outside the limit, it is considered to be lane-crossing or illegally overtaking another vehicle. Vehicles that are identified crossing lanes illegally will be tracked to identify their location after being identified as unlawful. The accuracy of using the proposed method is up to 96%. Using the dataset on selected video sequences of GRAM-RTM, 28 out of 30 total illegal lane-crossing vehicles have been detected.

In addition, Leo et al. [21] proposed a way to detect traffic anomalies by observing the flow of traffic by using the Speed Transition Matrix based on Center of Mass (CoM) computation. The proposed method uses STM computation to decide the spatiotemporal patterns of road traffic. Furthermore, the proposed method has three steps: data pre-processing, a combination of grid-based map segmentation and computation of STM, and the detection of traffic anomalies based on the estimation of CoM. To extract the various traffic flow patterns from different places, grid map segmentation has been applied to divide the city into smaller cells. Furthermore, the Transitions will be filtered into road segments and have a speed limit of 50km/h. This is to lower the possibility of false anomaly detection. The next step is to apply STM as a traffic data model. Anomaly traffic can be represented by huge variances in origin and destination speeds, which strongly depend on the represented pattern's position. The vehicle's quick braking or high-speed acceleration, along with the matching positions in the STM, lower left, and upper right corner, can be used to identify these dangerous situations in traffic. Anomaly origins can be identified by examining the geographical and temporal elements among the most anomalous traffic patterns. The precision score for the anomaly detection model is 92.88%. A summary of the conventional methods is presented in Table I.

TABLE I
A SUMMARY OF STATE-OF-THE-ART CONVENTIONAL METHODS

Authors	Method	Dataset	Class	Recognition Rate
Ha, Synh & Nguyen, Huy Hung & Ha, Tran & Ho, Phong et al. [17]	Improved Optical Flow	Self-collected dataset	36ms runtime for 800x480 resolution, 77ms runtime for HD resolution, 585ms runtime for 4K resolution	36ms runtime for 800x480 resolution, 77ms runtime for HD resolution, 585ms runtime for 4K resolution
H. Tan, Y. Zhai, Y. Liu and M. Zhang et al. [18]	Sparse Optical Flow	UCSD Ped1 Dataset	Anomaly detection	False positive rate less than 24%
Anandkumar et al. [20]	Principal Component Analysis (PCA)	Kaggle	Traffic congestion	32% prediction error rate
Arun Kumar H D, Prabhakar C J et al. [20]	Gaussian Mixture Model (GMM)	GRAM-RTM dataset	Lane Crossing Vehicles	96% accuracy
L. Tišljarić, S. Fernandes, T. Carić, and J. Gama et al. [21]	Speed Transition Matrix (STM)	Mireo Inc	Anomaly detection	92.88% accuracy

B. Deep Learning Methods

Murugan et al. [22] has proposed a way to extract, detect, and recognize moving vehicles by using the Region Convolutional Neural Network (RCNN) method. The first step is to perform frame extraction by converting the videos into frames. All the coloreds will be converted into gray color, this is to reduce the computational cost. Secondly, to detect objects in the recognition system background subtraction must be performed. In this procedure, frame smoothing, estimation, and background removal will be done. In this step the image will be filtered by moving the pixels of the rectangular group also known as the kernel. The background estimate is an important aspect, and smoothing procedures increase background estimation accuracy by averaging the fast color changes brought on by moving vehicles. Furthermore, a region convolutional neural network is used to identify multiple vehicle classifications. CNN is a widely known deep learning model for recognition. Moreover, RCNN is used to speed up the process by reducing the number of computations in the RCNN architecture. Lastly, a Convolutional Neural Network (CNN) is used to extract the feature; all of these features are used for vehicle recognition. In the convolutional layers, a kernel traverses the whole picture to create a convolved image to identify data from the input frames. Convolutional layers will capture all the generic features; therefore, the number of layers will be increased. RCNN achieved 91.3% recognition accuracy for different kinds of vehicles.

As per Tian et al. [23], there has been an improvement in the detection of car crash monitoring systems by emphasizing the advancement in Cooperative Vehicle Infrastructure Systems. Due to this finding, it now becomes apparent that the detection systems of car crashes have been made more accurate and efficient. The author uses the CAD-CVIS dataset in this context to carry this method forward. With the database acquired, an elaborate and comprehensive recognition model named YOLO-CA was able to assist in identifying vehicular accidents. This deep learning model can detect vehicular accidents even more accurately and quickly than other versions of the YOLO model. Furthermore, a method called multi-scale fusion and loss function with dynamics weights is being used to detect small objects on the road. The development of the YOLO-CA model integrates deep learning algorithms with the MSFF approach, as well as dependability and real-time performance under various traffic situations. The training results for the YOLO-CA, which show the evolution of many performance indices, are shown first during the experiment. Then, we provide the outcomes of certain comparison studies between YOLO-CA and other detecting algorithms. The visual outcomes are then presented using different sorts and scales of vehicle accident objects. The dataset is divided into three sections; the first section is 80% of the dataset is used for the training set, followed by 5% for the validation set, and the last section is 15 % for the Test set. Based on the experiment carried out by the author, the proposed YOLO-CA method has achieved an average precision of 90.02% faster than YOLOv3, which has only 86.20%.

Apart from that, Ghahremannezhad et al. [24] proposed a method to detect real-time traffic accidents using a deep learning method called YOLOv4. A few steps are used to detect traffic accidents in the detection framework. First, a pre-trained model using deep convolutional neural networks is applied. Secondly, a Kalman filter was applied to track the road user's movements. This was to monitor the trajectories and identify the road users' motion behavior. Furthermore, using the Kalman filter helps to identify dangerous anomalies that might cause minor or major accidents. In the article, the author uses the YOLOv4 model to pre-train the datasets from MS COCO. This is to detect objects. Secondly, for road user tracking, the Kalman filter is being used to estimate the model to predict the future location of every object that has been detected based on its current location for improving association. These trajectories are smoother and predict lost track. Lastly to detect an accident, the author emphasizes more on the pattern of motion of the road users for time detection and the location of the trajectory that has occurred. To determine whether objects are closer than a threshold to one another, all object pairs' Euclidean distances must first be determined. These object pairings are selected for further study because they have the potential to collide with each other. Furthermore, the angle of impact is used to identify accidents and near accidents. Based on the proposed method by the author it has a 93.10% accuracy of detection rate and a 6.89% false alarm rate.

Besides, Wang et al. [25] proposed a method to detect traffic congestion in freeways by using Convolutional Neural Network method. The dataset for this experiment is obtained from Shaanxi Province Traffic Management Bureau. In the dataset, there are four levels of different situations no vehicle, low traffic density, large density, and extremely high density of traffic. To detect the congestion level of traffic TrafficNet structure, AlexNet and VGGNet, and TrafficNet are used to classify the image based on the pre-trained network. Furthermore, the final three fully linked layers of AlexNet and VGGNeT will be removed, and additional new layers will be added to categorize the level of congestion traffic and non-congestion traffic. Since the pre-trained network has extract features, a classifier may use those features as input. This application uses a support vector machine (SVM) for a layer within a pre-trained network. After finishing the setup of the TrafficNet Structure a tuning step is performed to train the network. To differentiate between images with congestion and images without congestion under various lighting, weather, and other disturbances, the highest accuracy approaches 90%. To automatically detect and report traffic congestion, the present monitoring system will include this deep learning algorithm.

In addition, Nguyen et al. [26] proposed a way to detect traffic anomalies in traffic surveillance video by using Generative Adversarial Networks (GAN). This method is used to train a U-net generator to anticipate a subsequent frame using the present frame as a starting point and an encoded motion description. Based on the proposed method by the author, information retrieved from the input frames is combined with data on the intensity, gradient, and flow between the previous frame following the ground truth frame in the sequence to train the GAN-based U-Net model. The last frame in the frame sequence is used as the main indicator of the detail in the following frame in this input approach, which provides the motion direction by layering succeeding frames. The proposed method by the author has achieved 92.61% accuracy in detecting traffic anomaly.

Zhang et al. [27] proposed a method for detecting fast vehicles by using a method called connect-and-merge convolutional neural network CMNet. CMNet has two deeply fused streams and is a single deep neural network. Connect and-merge residual network (CMRN) is used for vehicle feature extraction in highly complicated scenes. (CMRN) develops by connecting and combining blocks, which seeks to improve information flow between remaining blocks to provide more advantageous features for vehicle recognition. In the proposed method the whole picture is used as input and concurrently produces all the vehicle positions and their related categories. First, CMRN is used to extract features of vehicles in this way the features of the vehicle can identify more significantly. By using this way, it is easier to train and produce excellent performance. The proposed method has obtained a result of 89.61% accuracy on the KITTI datasets. Furthermore, for the UA-DETRAC datasets the proposed method had obtained 91.71%accuracy. A summary of the deep learning methods is provided in Table II.

TABLE II
A SUMMARY OF DEEP LEARNING METHODS

Authors	Method	Dataset	Class	Recognition Rate
Murugan. V, Vijaykumar V.R, and Nidhila. A et al. [22]	Region-based Convolutional Neural Network(RCNN)	Self-collected dataset	Vehicle Recognition	91.3%
Daxin Tian, Chuang Zhang, Xuting Duan, And Xixian Wang et al. [23]	YOLO-CA	CAD-CVIS	CAD-CVIS	90.02%
Hadi Ghahremannezhad, Hang Shi, Chengjun Liu et al. [24]	YOLOv4	MS COCO dataset	Road Accident	93.10% 6.89% false alarm rate
Ping Wang, Li Li, Yinli Jin, Guiping Wang et al. [25]	Convolutional Neural Networks (CNN)	Self-collected	Traffic congestion	90%
Khac-Tuan Nguyen, Minh N. Do, Dat-Thanh Dinh, Minh-Triet Tran et al. [26]	Generative Adversarial Network (GAN)	Self-collected	Traffic anomaly detection	92.61%
Fukai Zhang , Feng Yang, Ce Li , And Guan Yuan et al. [27]	Connect-and-merge convolutional neural network CMNet	UA-DETRAC datasets KITTI datasets	Speeding Detection	91.71% 89.61%

II. Materials and Method

A. YOLO for Vehicle Detection

The YOLO model paired with the COCO dataset is highly effective for computer vision tasks. This implementation allows the YOLO model to accomplish vision tasks without requiring further extensive retraining. For this reason, the YOLO model was selected as it was plausible for the current study as it is automatically realizable, fast, and accurate in recognizing objects in real-time. The images in the dataset will help improve the accuracy of the YOLOv4 models. With over 330,000 images in the dataset collected, there are over 80 classes of objects, such as cars, buses, and motorcycles in the dataset. Such a diverse data set helps in improving the accuracy of the model. As such, in various conditions, i.e., during the day, at night, or even rain, there is a better chance that the YOLOv4 model will be able to detect various vehicles if not all. On the other hand, vehicles can be detected by the other elements and the background chaos.

The design of YOLOv4 includes parts called the backbone, neck, and head, which work together to detect objects quickly. It uses new techniques like Self-adversarial-training and Cross-Stage-Partial connections to improve its accuracy. The backbone of YOLOv4, trained on ImageNet, helps predict the type of objects and where they are in an image. The general architecture of the YOLOv4 model is depicted in Fig. 1.

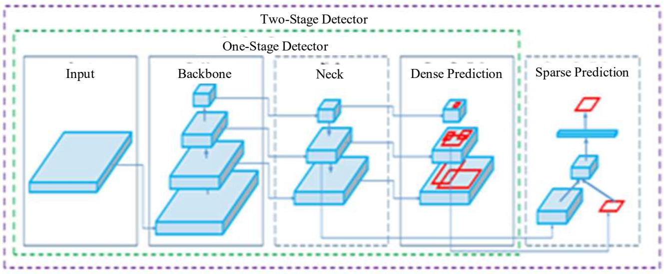

Fig. 1 Architecture of YOLOv4

The ability of the YOLOv4 model to detect objects in real-time is strengthened by working using the OpenCV's Deep Neural Network [28], [29] module. This module supports different training engines, such as TensorFlow and Darknet. YOLO v4 works by scanning images and cutting them into grids, with each grid sending out bounding boxes predicting the location of the object (Fig. 2). Such a mechanism enables YOLO v4 to quickly ascertain the presence of objects and what they are. The tools of OpenCV make it possible to perform these functions making the implementation of YOLO v4 a standalone task without dependence on other software programs.

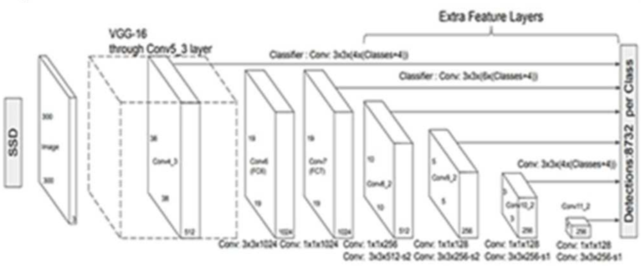

Fig. 2 OpenCV DNN Architecture

In the present analysis, the input image, which contains moving vehicles, is first processed in order to facilitate the analysis. Then it goes through CNN to produce feature maps to search out the significant features of the image. Then, the YOLO algorithm offers a collection of bounding boxes with the associated probability of vehicle classification for those features. To further increase accuracy, bounding boxes with a value less than the threshold are thrown away. The new bounding boxes that are originally relative to the vehicle dimensions, are now altered to the international coordinate system. These coordinates indicate the respective position of the vehicles as seen in other frames of the original image. The vehicle detection processes are summarized and presented in Algorithm 1.

Algorithm 1 Pseudocode for Vehicle Detection

```
processed_image =
Preprocess_Image(image)
feature_maps =
CNN_Forward_Pass(processed_image)
boxes, class_probs =
Generate_Bounding_Boxes_And_Class_Prob
s(feature_maps)
filtered_boxes =
Filter_Low_Confidence(final_boxes)
absolute_boxes =
Convert_Relative_To_Absolute(filtered_
boxes)
return absolute_boxes
```

B. Optical Flow for Vehicle Detection

Optical flow is used for tracking vehicle movement, providing detailed information about the direction in which vehicles move, such as up, down, left, and right. This is crucial for traffic monitoring to understand vehicle flow patterns on roads. Optical flow works by measuring how much individual pixels move between two video frames, assuming that the pixel brightness doesn't change much in that time [30]. This helps calculate the speed of the pixels, and thus the vehicle, from one frame to the next.

Lucas-Kanade [31] and Horn-Schunck [32] are well-known methods to calculate optical flow. The Lucas-Kanade method assumes that the motion of pixels is uniform in a small area, while the Horn-Schunck method assumes the motion is smooth over the whole image. We use the Lucas-Kanade method in this research because it is good at tracking small and consistent movements, which helps monitor traffic on the roads.

Optical flow is combined with YOLO to track vehicle movements. It starts by identifying vehicles in a video frame and then tracking their movement over time using optical flow. This method improves tracking accuracy and can handle multiple moving objects, making it suitable for busy urban settings or highways. Optical flow helps spot unusual movements like sudden stops, which could indicate accidents or risky driving behaviors. The formula for optical flow is shown in Equation 1.

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \quad (1)$$

where $u = dx/dt$ and $v = dy/dt$. The image gradients in the horizontal, vertical, and temporal axes are represented by dI/dx, dI/dy, and dI/dt, respectively. To determine movement over time, we must solve the optical flow equations *u*(*dx*/*dt*) and *v*(*dy*/*dt*). The optical flow equation for *u* and *v* cannot be solved directly as it only has a single equation for two unidentified variables. To address this issue, the Lucas-Kanade method is adopted. The procedure for performing vehicle tracking is given in Algorithm 2.

Algorithm 2 Pseudocode for Vehicle Tracking

```
1. old_points=np.array[]
2. tracking_objects= {}
3. stationary_treshold = 10
4. stationary_history = {}
5. new_points = []
6. if dy > -1
     return 'down'
   elseif dy <6
     return 'up'
```

The function requires two arguments: new and old, which indicate a point's new and old positions, respectively. These coordinates are usually the centroids of detected vehicles in every subsequent input video frame. In the line `new.ravel() - old.ravel()` the `ravel()` function converts the arrays to one-dimensional arrays, and the difference creates the displacement vector, which has two components (*dx*, *dy*) for horizontal and vertical movement. In this situation, the function just looks at the vertical component (*dy*).

The vertical displacement (*dy*) is evaluated against two thresholds if *dy* > -1, the function reads it as the vehicle is traveling downward in the frame. This threshold is quite near to zero, implying that even a minor downhill movement is sufficient to define the direction as 'down'. If *dy* is less than 6, the vehicle is expected to go upward. This threshold permits small movements that are not completely vertical.

C. Combination of YOLOv4 and Optical Flow

Combining YOLO vehicle detection and optical flow motion tracking is useful for spotting traffic violations like wrong-way driving. This approach integrates both techniques to assess traffic situations. First, YOLO detects vehicles within video frames. It is effective at spotting vehicles in different lighting and weather conditions. It enables the proposed method to track the movement of these vehicles reliably in various traffic conditions. After YOLOv4 detects the vehicles, optical flow starts to track how the vehicles move from one frame to another. The tracking includes both the speed and direction of the vehicles. YOLO keeps detecting vehicles as they appear, while optical flow follows their movement. The system can handle vehicles as they come into the frame, move through it, or leave it. The system can also spot unusual patterns like sudden stops or vehicles going the wrong way that could indicate a potential occurrence of an accident.

D. Wrong-Way Driving Detection

The pseudocode for the proposed wrong way driving detection method is presented in Algorithm 3. First, the system detects a vehicle and tracks its movement through coordinates provided by YOLOv4. Then, optical flow continues to track the vehicle's smaller movements frame by frame. The system requires prior knowledge of the direction permissible in each road segment, e.g. upward or downward. Based on this prior knowledge, the system will look for vehicles moving in the opposite direction. For instance, if traffic is supposed to go upward, any vehicle moving downward is driving in the wrong direction. If a vehicle moves differently from the expected direction, the system notices this through optical flow and sends out real-time alerts.

Algorithm 3 Pseudocode for Wrong Way Driving

```
1. old_points=np.array[]
2. tracking_objects= {}
3. stationary_treshold = 10
4. stationary_history = {}
5. new_points = []
6. if dy > -1
     return 'down'
   elseif dy <6
     return 'up'
7. if direction 'up'
8. cv2.puttext "wrong way"
```

III. Result and Discussion

A. Dataset

For this research, we propose to train the YOLOv4 model for vehicle detection based on the COCO (Common Objects in Context) dataset. The abundance and auxiliary annotations of the COCO dataset help ease the tasks of seeking out vehicles on roads. The datasets contain cars, buses, trucks, motorcycles, and bicycles that belong to 80 categories. First, the site acquires the COCO dataset, and images and vehicle-related annotations are queried. Next, we will obtain the pre-trained weights of YOLOv4 butterflies. These weights are suited for a more excellent range of object classification because they were trained on datasets with objects.

B. Vehicle Direction Detection

We use the YOLOv4 model to detect vehicles in a video. Then, we set up the Lucas-Kanade Optical Flow method to track these vehicles. This setup involves choosing specific settings for tracking points and the timing of frames. We also define thresholds to help determine if a vehicle is moving up or down. The formulas *u*(*dx/dt*) and *v*(*dy*/*dt*) are used to calculate the movement vectors that describe how vehicles move over time. The results for vehicle detection are depicted in the subsequent sections. The proposed method is tested on different videos containing various traffic conditions downloaded from online sources.

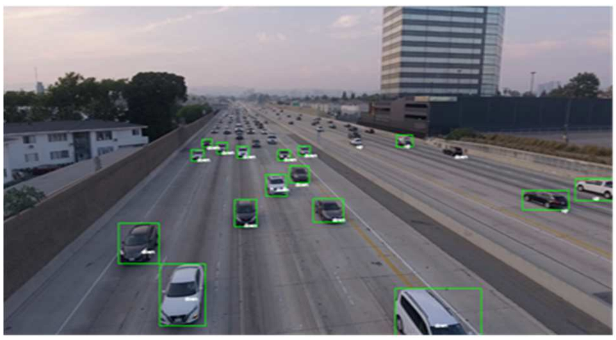

Fig. 3 Sample of direction detection result

Fig. 3 shows how the system detects the direction of a vehicle from a CCTV video taken on a US highway. The local rules say that vehicles on the left side of the road should move downward, while those on the right should go upward. If a vehicle appears close enough to the CCTV camera (with reasonable visibility), the system can accurately determine which way the vehicle is moving.

Fig. 4 Wrong direction detection in low resolution video

Fig. 4 shows the results of detecting objects in a video that has low resolution. Low resolution causes pixelation, which blurs important details of vehicles and makes it hard for the system to see and follow these details using optical flow methods. Optical flow, like the Lucas-Kanade method, needs clear features to track, but these can be blurry in low-resolution video. This blurriness leads to inaccurate tracking and makes it difficult to figure out the direction of movement. Also, in low-resolution videos, the outlines of vehicles aren't clear, and there's not much contrast between the vehicle and its background, making it even harder for the system to tell which way the vehicle is going.

C. Analysis for Wrong-Way Driving

The experiment to detect wrong way driving vehicles will be structured into five different parts. Initially, the YOLOv4 model will be employed to identify vehicles in the input video. Following vehicle detection, we configure the parameters for the Lucas-Kanade Optical Flow to facilitate vehicle tracking, including setting variables for tracking points and frames. Once the optical flow is set up, it tracks the direction of the moving vehicle within the frames and displays this information within the bounding box of the detected vehicle. The expected direction of traffic flow, which helps determine wrong-way driving, is predefined based on the general traffic directions.

Fig. 5 depicts the first scenario for wrong-way detection. Each vehicle on the road has been identified and enclosed in a bounding box. The direction of each vehicle's movement is indicated by a text adjacent to its bounding box. For example, the word "down" next to a bounding box confirms that the system has accurately determined the vehicle is moving downward. A sedan is marked as traveling the wrong way with the direction label "up" on its bounding box. This indicates that the sedan is moving against the flow of traffic, and the system flags it as a traffic violation and potential safety hazard.

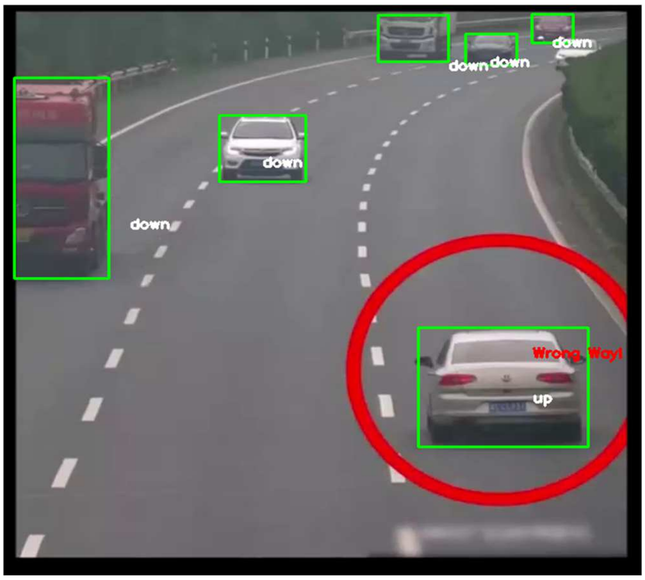

Fig. 5 Wrong-way detection scene 1

Fig. 6 shows a mistake in how the system identified the direction a sedan was travelling in. The sedan was moving up, but the system wrongly marked it as moving down. This error might be due to threshold settings for the sensitivity level that are not able to catch small changes in direction. Note that a standard threshold is used across the same traffic setting. Another possible reason is due to the video's low resolution. The vehicles look blurry in the low-resolution image and complicate the system's ability to correctly spot and follow their movements.

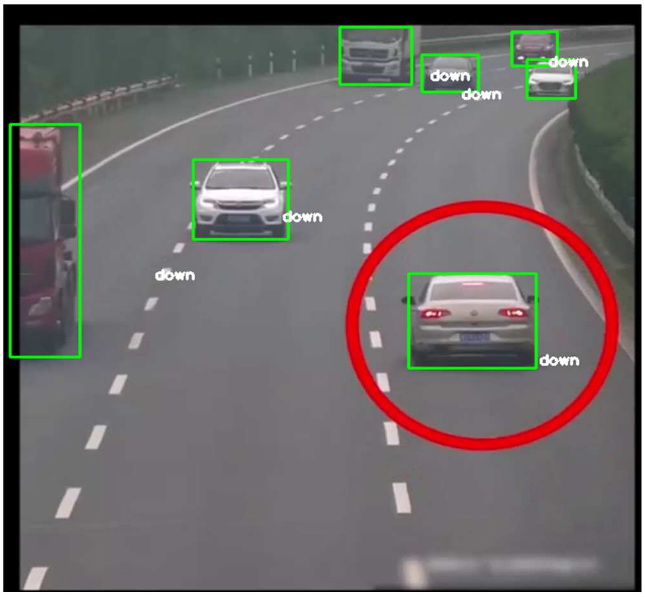

Fig. 6 False direction detection

Fig. 7 explains another case for detecting wrong way driving on a highway. The image shows the presence of many cars that are moving downward and the directional indicator on each car is noted. Most of the indicators on the cars which are enclosed in green boxes marked “down” show that indeed they are traveling in the correct traffic direction. In the scene, one car is in a red box has been labeled “Wrong Way!” and is showing an “up” direction. This shows the system has detected this car travelling in the opposite direction and this indeed is a cause for concern since it appears to flag a possible breach of a traffic rule.

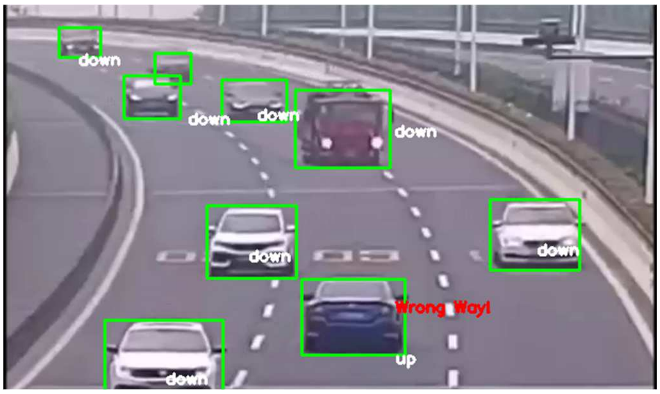

Fig. 7 Wrong-way detection scene 2

IV. Conclusion

This study has shown that the combination of YOLOv4 model and the Lucas-Kanade optical flow provides a cost-effective way for traffic monitoring. These techniques enhance the safety of roadways by ensuring that the vehicles are tracked adequately in real time. In the future, we will further improve and test the proposed methods under different traffic conditions. We will also explore more advanced machine learning models and other optical flow methods to make these systems better. By improving the technological aspect of traffic surveillance, we hope for more efficient and safer traffic ensuring minimum loss of lives and more efficient ways of tackling traffic problems across the globe.

Acknowledgment

This research is supported by TM R&D Fund (MMUE/220023).

References

[1] G. Wang, Z. Pang, F. Wang, Y. Chen, H. Dai, and B. Wang, "Urban Fiber Based Laser Interferometry for Traffic Monitoring and Analysis," *Journal of Lightwave Technology*, vol. 41, no. 1, pp. 347–354, Jan. 2023, doi: 10.1109/jlt.2022.3209499.

[2] D. A. Guastella and E. Pournaras, "Cooperative Multi-Agent Traffic Monitoring Can Reduce Camera Surveillance," *IEEE Access*, vol. 11, pp. 142125–142145, 2023, doi: 10.1109/access.2023.3343620.

[3] C.-J. Lin and J.-Y. Jhang, "Intelligent Traffic-Monitoring System Based on YOLO and Convolutional Fuzzy Neural Networks," *IEEE Access*, vol. 10, pp. 14120–14133, 2022, doi:10.1109/access.2022.3147866.

[4] S. Liang et al., "Fiber-Optic Auditory Nerve of Ground in the Suburb: For Traffic Flow Monitoring," *IEEE Access*, vol. 7, pp. 166704–166710, 2019, doi: 10.1109/access.2019.2952999.

[5] Z. Pu, Z. Cui, J. Tang, S. Wang, and Y. Wang, "Multimodal Traffic Speed Monitoring: A Real-Time System Based on Passive Wi-Fi and Bluetooth Sensing Technology," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12413–12424, Jul. 2022, doi:10.1109/jiot.2021.3136031.

[6] G. Liu et al., "Smart Traffic Monitoring System Using Computer Vision and Edge Computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12027–12038, Aug. 2022, doi: 10.1109/tits.2021.3109481.

[7] C. Chen, K. Ota, M. Dong, C. Yu, and H. Jin, "WITM: Intelligent Traffic Monitoring Using Fine-Grained Wireless Signal," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 206–215, Jun. 2020, doi: 10.1109/tetci.2019.2926505.

[8] B. E B, A. M, S. G, D. Jose, and H. J. Magadum, "Intelligent Traffic Monitoring and Management System," *2023 International Conference on Computer, Electronics & Electrical Engineering & their Applications (IC2E3)*, pp. 1–5, Jun. 2023, doi:10.1109/ic2e357697.2023.10262689.

[9] A. Alruban, H. A. Mengash, M. M. Eltahir, N. S. Almalki, A. Mahmud, and M. Assiri, "Artificial Hummingbird Optimization Algorithm With Hierarchical Deep Learning for Traffic Management in Intelligent Transportation Systems," *IEEE Access*, vol. 12, pp. 17596–17603, 2024, doi: 10.1109/access.2023.3349032.

[10] M. S. I. Mohd Zubil, Z. Che Embi, and K. I. Ghauth, "Assessing the Efficiency of Deep Learning Methods for Automated Vehicle Registration Recognition for University Entrance," *Journal of Informatics and Web Engineering*, vol. 3, no. 2, pp. 57–69, Jun. 2024, doi: 10.33093/jiwe.2024.3.2.4.

[11] L. Hu, H. Li, P. Yi, J. Huang, M. Lin, and H. Wang, "Investigation on AEB Key Parameters for Improving Car to Two-Wheeler Collision Safety Using In-Depth Traffic Accident Data," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 113–124, Jan. 2023, doi:10.1109/tvt.2022.3199969.

[12] X. Chen and Y. Zhai, "A multi-objective traffic flow detection system based on an improved yolov4 algorithm," *2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, pp. 726–730, Feb. 2023, doi:10.1109/eebda56825.2023.10090516.

[13] C. Dewi, R.-C. Chen, Y.-T. Liu, X. Jiang, and K. D. Hartomo, "Yolo V4 for Advanced Traffic Sign Recognition With Synthetic Training Data Generated by Various GAN," *IEEE Access*, vol. 9, pp. 97228–97242, 2021, doi: 10.1109/access.2021.3094201.

[14] Q. Liu, X. Fan, Z. Xi, Z. Yin, and Z. Yang, "Object detection based on Yolov4-Tiny and Improved Bidirectional feature pyramid network," *Journal of Physics: Conference Series*, vol. 2209, no. 1, p. 012023, Feb. 2022, doi: 10.1088/1742-6596/2209/1/012023.

[15] U. Ali, M. A. Ismail, R. A. Ariyaluran Habeeb, and S. R. Ali Shah, "Performance Evaluation of YOLO Models in Plant Disease Detection," *Journal of Informatics and Web Engineering*, vol. 3, no. 2, pp. 199–211, Jun. 2024, doi: 10.33093/jiwe.2024.3.2.15.

[16] J. J. Ng, K. O. M. Goh, and C. Tee, "Traffic Impact Assessment System using Yolov5 and ByteTrack," *Journal of Informatics and Web Engineering*, vol. 2, no. 2, pp. 168–188, Sep. 2023, doi:10.33093/jiwe.2023.2.2.13.

[17] S. V.-U. Ha, H.-H. Nguyen, H. M. Tran and P. Ho-Thanh, "Improved optical flow estimation in wrong way vehicle detection", *Journal of Information Assurance and Security*, vol. 9, no. 5, pp. 165-169, 2014.

[18] H. Tan, Y. Zhai, Y. Liu, and M. Zhang, "Fast anomaly detection in traffic surveillance video based on robust sparse optical flow," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1976–1980, Mar. 2016, doi:10.1109/icassp.2016.7472022.

[19] H. Niu, Y. Lu, A. Savvaris, and A. Tsourdos, "Efficient Path Planning Algorithms for Unmanned Surface Vehicle," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 121–126, 2016, doi: 10.1016/j.ifacol.2016.10.331.

[20] A. Kumar H D and P. C J, "Detection and Tracking of Lane Crossing Vehicles in Traffic Video for Abnormality Analysis," *International Journal of Engineering and Advanced Technology*, vol. 10, no. 4, pp. 1–9, Apr. 2021, doi: 10.35940/ijeat.c2141.0410421.

[21] L. Tišljarić, S. Fernandes, T. Carić, and J. Gama, "Spatiotemporal Traffic Anomaly Detection on Urban Road Network Using Tensor Decomposition Method," *Discovery Science*, pp. 674–688, 2020, doi:10.1007/978-3-030-61527-7_44.

[22] M. V, V. V.R, and N. A, "A Deep Learning RCNN Approach for Vehicle Recognition in Traffic Surveillance System," *2019 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0157–0160, Apr. 2019, doi:10.1109/iccsp.2019.8698018.

[23] D. Tian, C. Zhang, X. Duan, and X. Wang, "An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems," *IEEE Access*, vol. 7, pp. 127453–127463, 2019, doi: 10.1109/access.2019.2939532.

[24] H. Ghahremannezhad, H. Shi, and C. Liu, "Real-Time Accident Detection in Traffic Surveillance Using Deep Learning," *2022 IEEE International Conference on Imaging Systems and Techniques (IST)*, Jun. 2022, doi: 10.1109/ist55454.2022.9827736.

[25] P. Wang, L. Li, Y. Jin, and G. Wang, "Detection of unwanted traffic congestion based on existing surveillance system using in freeway via a CNN-architecture trafficnet," *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1134–1139, May 2018, doi: 10.1109/iciea.2018.8397881.

[26] K.-T. Nguyen, D.-T. Dinh, M. N. Do, and M.-T. Tran, “Anomaly Detection in Traffic Surveillance Videos with GAN-based Future Frame Prediction,” *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pp. 457–463, Jun. 2020, doi:10.1145/3372278.3390701.

[27] F. Zhang, F. Yang, C. Li, and G. Yuan, “CMNet: A Connect-and-Merge Convolutional Neural Network for Fast Vehicle Detection in Urban Traffic Surveillance,” *IEEE Access*, vol. 7, pp. 72660–72671, 2019, doi: 10.1109/access.2019.2919103.

[28] F. E. Fernandes and G. G. Yen, “Automatic Searching and Pruning of Deep Neural Networks for Medical Imaging Diagnostic,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5664–5674, Dec. 2021, doi: 10.1109/tnnls.2020.3027308.

[29] J.-R. Lee, K.-W. Ng, and Y.-J. Yoong, “Face and Facial Expressions Recognition System for Blind People Using ResNet50 Architecture and CNN,” *Journal of Informatics and Web Engineering*, vol. 2, no. 2, pp. 284–298, Sep. 2023, doi: 10.33093/jiwe.2023.2.2.20.

[30] J. Lv, H. Ji, Y. Jiang, and B. Wang, “A New Flow Pattern Identification Method for Gas–Liquid Two-Phase Flow in Small Channel Based on an Improved Optical Flow Algorithm,” *IEEE Sensors Journal*, vol. 23, no. 22, pp. 27634–27644, Nov. 2023, doi:10.1109/jsen.2023.3321632.

[31] K. L. Leonida, K. V. Sevilla, and C. O. Manlises, “A Motion-Based Tracking System Using the Lucas-Kanade Optical Flow Method,” *2022 14th International Conference on Computer and Automation Engineering (ICCAE)*, pp. 86–90, Mar. 2022, doi:10.1109/iccae55086.2022.9762423.

[32] R. K. Bhogal and V. Devendran, “Motion Estimating Optical Flow for Action Recognition : (Farneback, Horn Schunck, Lucas Kanade and Lucas-Kanade Derivative Of Gaussian),” *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, pp. 675–682, Jan. 2023, doi:10.1109/idciot56793.2023.10053515.