Traffic & Transportation

An Improved Object Detection and Trajectory Prediction Method for Traffic Conflicts Analysis

Lu YANG[1], Ahmad Sufril Azlan MOHAMED[2], Majid Khan Majahar ALI[3]

Original Scientific Paper
Submitted: 31 Jan. 2023
Accepted: 12 June 2023

[1] yanglu19881109@gmail.com, School of Computer Sciences, Universiti Sains Malaysia
[2] Corresponding author, sufril@usm.my, School of Computer Sciences, Universiti Sains Malaysia
[3] majidkhanmajaharali@usm.my, School of Mathematical Sciences, Universiti Sains Malaysia

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT

Although computer vision-based methods have seen broad utilisation in evaluating traffic situations, there is a lack of research on the assessment and prediction of near misses in traffic. In addition, most object detection algorithms are not very good at detecting small targets. This study proposes a combination of object detection and tracking algorithms, Inverse Perspective Mapping (IPM), and trajectory prediction mechanisms to assess near-miss events. First, an instance segmentation head was proposed to improve the accuracy of the object frame box detection phase. Secondly, IPM was applied to all detection results. The relationship between them is then explored based on their distance to determine whether there is a near-miss event. In this process, the moving speed of the target was considered as a parameter. Finally, the Kalman filter is used to predict the object's trajectory to determine whether there will be a near-miss in the next few seconds. Experiments on Closed-Circuit Television (CCTV) datasets showed results of 0.94 mAP compared to other state-of-the-art methods. In addition to improved detection accuracy, the advantages of instance segmentation fused object detection for small target detection are validated. Therefore, the results will be used to analyse near misses more accurately.

KEYWORDS

near-miss; object detection; object tracking; trajectory prediction.

1. INTRODUCTION

Asian cities are growing in size with accelerated economic development and urbanisation. Traffic congestion is particularly severe in large cities such as Hong Kong, Singapore, Kuala Lumpur and Bangkok [1]. With the continuous progress of urbanisation, urban traffic is facing severe pressures such as traffic congestion and frequent accidents. Therefore, the effective monitoring of traffic flow and traffic conditions is an efficient way to reduce traffic congestion.

However, commuters tend to use their own cars to get to work to avoid long waiting times, uncertain frequencies, and overcrowded buses and subways. Unfortunately, traffic jams cost commuters an average of one hour per day and, in some cases, up to three hours. Motorcycles, bicycles and electric bikes can help them eliminate these annoying things. However, the number of fatalities among motorcyclists is high [2]. In 1997, a road safety initiative "Vision Zero" is presented in Sweden. It aims to reduce the number of traffic fatalities and severe injuries to zero [3]. This work (near-miss judgments) can be used to identify potential hazards and unsafe conditions that could lead to accidents, which is essential to achieving the goals of "Vision Zero". By evaluating near-miss incidents, transportation planners and policymakers can take proactive steps to prevent accidents from occurring in the future and create safer roads for all users. Overall, the situational link between near-miss judgments and "Vision Zero" is that they both promote a proactive and data-driven approach to road safety. By identifying potential hazards and taking action to prevent accidents, near-miss judgments can play an essential role in achieving the goals of "Vision Zero" and creating safer roads for all users.

The analysis and research of road accidents involving motorcycles and bicycles are becoming increasingly urgent. Therefore, the study of road traffic cannot be limited to 4-wheeled vehicles, but should also include 2 and 3-wheeled vehicles (which are small targets on the road). The purpose of this research is to study va-

rious vehicles on the road using closed-circuit television (CCTV) cameras in Penang, Malaysia. In contrast to crashes, which have clear signs of contact and measurable damage, near-misses are events without contact between road users and significant damage [4]. Research on road safety is not limited to accidents, but also extends to near-misses and unsafe situations. This research aims to use image recognition methods based on deep learning to effectively detect near-misses and their frequencies.

1.1 Near-miss events

Over the years, safety-related incidents have been described by various authors as near-misses, near-crashes, traffic conflicts, safety-critical incidents, traffic interactions and traffic encounters [5]. Near-misses are events in which there is no contact between road users. However, a near miss is any event that has the potential to cause injury or damage to property or the environment, but it does not. Heinrich's law [6] states that for every accident resulting in one serious injury, there are 29 accidents resulting in minor injuries and 300 accidents resulting in no injuries. This suggests that, in practice, there is a positive correlation between actual casualties and near misses. If the probability of danger is reduced, the probability of injury can also be reduced. Therefore, people must learn from accidents and near-misses [7]. Previous research has demonstrated the applicability of Heinrich's law to the study of near misses in traffic scenarios [8]. The author has used the media's description of the process of road accidents and analysis of accident causes as a starting point for risk management of motor vehicle accidents and has made reasonable compromises to construct the Heinrich accident causation model. The five events in the model are clear and coherent, and form an accident chain; when five events occur simultaneously, it is clear that an accident has occurred, in line with Heinrich's chain theory.

The near-miss event already contains the basic elements of the accident. This clearly shows that the detection and prediction of near misses are of great importance for road and production safety. Only through continuous research and analysis of hidden dangers and loopholes can effective preventive measures be implemented to ensure road safety.

1.2 Measure of near-miss

Near-misses are usually identified by physical proximity or evasive actions. In terms of avoidance actions, braking and swerving are the two most common actions used to avoid collisions. The analysis of near-misses includes the following aspects: different types of vehicles (road users) and different types of roads. However, regardless of whether it is spatial distance or evasive behaviour, the speed of the road user is also a parameter to be considered.

1.3 Application of near-miss

The analysis of near-miss events on roads has a wide range of applications, the most common of which are the following:

- Analysis of driving behaviour and habits
- Offline data safety analysis
- Real-time monitoring systems
- Autonomous driving

Near-miss events are mostly caused by drivers, including their driving speed, cornering angle, single-trip distance and time. Analysing these driver behaviours and habits can lead to traffic accidents or traffic accident patterns. Conversely, traffic data can be analysed to remind drivers of how they react to current road conditions.

Today, many studies are based on text reports. Although this type of data is not real-time, it can be analysed for key information on road safety. It can be used to predict the severity of an accident.

The use of real-time monitoring systems on roads is crucial. This type of analysis typically uses video image data to create models by training on historical data. The real-time video captured by the CCTV cameras is then fed into the model to detect and output near-miss events.

Autonomous driving technology means that the vehicle senses its environment through various sensors (radar, camera, LIDAR, etc.) and makes decisions to control the vehicle, driving itself "without the driver" [9–11]. The trial evaluation and prediction mentioned in this paper can be applied to the decision-making phase of autonomous driving technology. For example, the relationship between all vehicles is calculated from the lane position information and the vehicle position and trajectory identified by the camera, and the steering wheel is controlled by an actuator (Electronic Steering System EPS) based on the near-miss threshold.

1.4 Related Works

The definition and data collection of near-miss events is also a complex process. In [12], a near-miss incident database (NIDB) is presented, which contains a large number of near-miss scenes obtained. This research then developed a near-miss recognition method that combines semantic segmentation and optical flow to improve the trajectory-pooled deep-convolutional descriptors (TDD) model. Of course, more research now tends to recalculate after identifying the object to determine whether a near-miss has occurred. Instead of using the calibrated data set directly. The authors [13] choose a deep learning model: Residual Networks (ResNets) to experiment and improve it so that each traffic near miss has a better prediction accuracy. It is a pedestrian near-miss dataset on vehicle-mounted recorders and joint learning of pedestrian detection and hazard prediction. As we all know, when the network depth reaches a certain level, the error increases, the effect becomes worse and the gradient disappears more obviously. In backward propagation, the gradient cannot be fed back to the previous network layer and the previous network parameters cannot be updated, resulting in poor training. The system cannot obtain evidence to predict low risk due to overfitting. Therefore, improving and optimising deep learning algorithms is also the main goal of this research. Currently, the mainstream near-miss research is focused on cars. Given the national conditions in many Asian countries, a significant number of people choose motorcycles and electric motorcycles for their daily commute. The paper [14] proposes a hybrid method (genetic algorithms and simulated annealing) to identify and predict the significant factors (conditions) in the severity of motorcyclist traffic accidents in Europe. By reproducing a typical critical situation, method [15] has discussed the effectiveness of near-miss data for understanding accident causation. Incidents at junctions between vehicles turning right and motorcycles going straight were specifically analysed, and several factors contributing to the failure to detect were identified. No information is currently available as official accident statistics only include crashes, not near-misses.

Vehicle detection plays an active and important role in road safety and is of great interest to academia and industry. Depth study has achieved a breakthrough in vehicle detection applications. The Single Shot Detector (SSD) algorithm is one of the object detection algorithms and its main challenge is high computational complexity and low accuracy. When the scales and aspect ratios of the standard bounding boxes are set in the vehicle detection algorithm, it is faster than normal SSD. It adds an inception block to the extra layer in the SSD before prediction and improves the performance of normal vision without increasing its computational complexity [16]. The validity of this improved algorithm is verified on the KITTI and UVD datasets. During training, these default bounding boxes are first matched to the ground truth boxes. Using k-means clustering to obtain the aspect ratios of the vehicle samples, the aspect ratios of the default bounding boxes are brought closer to the ground truth box. This method can reduce the number of bounding boxes and speed up detection. On real roads, the objects to be detected are moving vehicles, motorcycles and pedestrians. Therefore, object tracking is also an indispensable technical means. As for the object tracking strategy, the model of the vehicle detection method can work with a spatial constraint, and filter template matching [17]. Then it is used in conjunction with YOLO, object attribute information and IOU. And a squeeze-and-excitation channel attention mechanism can be adopted to improve feature learning.

The object detection framework based on deep learning has made brilliant achievements. However, due to the small size and complicated background of road users and traffic signs extracted in the real world, the normal deep learning algorithm is insufficient to effectively achieve the detection accuracy and detection speed. In [18], an efficient algorithm based on the YOLOv3 model for real-world traffic sign detection is proposed. First, the author constructs a deep neural network based on YOLOv3 for traffic sign detection. In addition, network pruning is used to minimise network redundancy and model size. Then, a fourth scale prediction performance branch is added to expand the detection range and enrich the feature maps for multi-scale prediction. Finally, the cost of classification prediction is increased, and the confidence loss and classification loss between each scale prediction are multiplied by the corresponding weight. Another optimisation approach for the small object is to use the k-means clustering algorithm to cluster the bounding boxes of the traffic signs to identify the anchor size for the YOLO algorithm [19].

In summary, our study addresses the following questions:

- Improving the detection accuracy of small targets.
- Tracking and localisation of dynamic objects.
- The effect of object motion speed on object trajectory prediction.
- Image and video-based analysis of traffic conflict events.

1.5 The main contribution

The main solution introduced in this research is a new method for detecting and predicting near misses in urban intelligent traffic management, based on the well-known concept of object detection and tracking. This research aims to improve the detection and tracking process to improve detection accuracy and reduce the false positive rate of near misses. The contributions of this research can be summarised as follows.

- An enhanced object detection algorithm. This is an improved object detection algorithm obtained by improving the YOLOv7 algorithm. According to the data characteristics of this research, image segmentation and clustering optimisation are used together. In order to obtain the best weight and bias value, and thus higher classification performance.
- According to the road users' characteristics, the improved trajectory prediction algorithm is used to predict the probability of near-miss events in the next few seconds and anticipate the risk in advance.
- To confirm whether there is a near-miss between the target objects, the relationship between the target objects is examined using the location and speed information. And there is currently a lack of research in this area in the video/imaging field.

The remainder of this paper is divided into the following sections. Section 2 describes the proposed near-miss detection framework and the object detection, tracking and trajectory methodology. Section 3 provides details of the model and describes the experimental procedure. The modelling results and discussion are presented in Section 4. Finally, Section 5 concludes the paper.

2. DATA AND METHODOLOGICAL ISSUES

Defining and recognising near-miss events pose different problems owing to the different types of data used. Today, the two main types of data are text- and vision-based. Text-based methods are primarily based on traffic police, hospital reports and questionnaires. Problems include the under-reporting of near-miss data, a small sample size, over-dispersion, outdated data and omitted variable bias. The data used in the vision-based approach were obtained mainly from CCTV and video recordings from vehicle recorders. Problems include large amounts of data, complex data (e.g. weather and light changes) and many road users and road types (e.g. different types of vehicles, roads and even different countries). However, the advantages of vision-based systems are obvious, such as real-time analysis and the at-a-glance presentation of results.

With the development of modern management, the CCTV monitoring system [20] has become a modern management tool commonly used today. It can inform management, maintenance and security personnel of any situation on-site in the form of images and text. It enables people to react quickly and effectively, supports the entire incident process, records important data and provides a practical basis for accident management. Owing to the characteristics of CCTV technology, it is also widely used in urban traffic management [21]. CCTV surveillance networks have been established in major cities worldwide and CCTV video surveillance systems have become an important way to obtain target information. Therefore, this research uses CCTV surveillance video to track all types of vehicles and pedestrians in a traffic network and to detect all types of near-miss events.

2.1 Inconsistency in identification

Near misses occur without significant collisions; therefore, there is no universal criterion to guide identification, leading to inconsistent criteria. In particular, different studies have different judgement thresholds, making the cross-validation and generalisation of research results difficult. In this study, the probability of a near collision is determined by the real-time distance between road users.

2.2 Multi-dimension

Near-misses are not a single type of event. Each dimension has different indicators, scoring methods and metrics. One approach combines different indicators to create a new severity index. Another approach is to assess each dimension individually using different indicators as safety ratings. The latter approach was selected for this study.

2.3 Unobserved heterogeneity

Typically, near-miss events are extracted from text or video data. In addition to the conventional road and vehicle behaviours observed in these data, many other factors can be studied (e.g. vehicle year, quality,

braking performance and the driver's driving habits and mental state). These unobserved factors are known as unobserved heterogeneities. Neglecting these elements can lead to biased models, and therefore, incorrect conclusions. However, the driver's mental state cannot be observed using low-resolution, long-distance data. The driver's driving habits cannot be learned from the CCTV data. This problem should be considered in future studies.

3. MODELLING METHODS

The main modelling approaches include modelling according to specific near-miss types and modelling by identifying road users and then defining the relationship between them. The designed vehicle detection and identification system mainly consists of CCTV video acquisition, road user localisation and tracking, and near--miss discrimination between road users. This study is based on the POL37 dataset of 1841 road traffic CCTV cameras in Penang, Malaysia, as these data have not undergone any cleaning and pre-processing. Therefore, this study adopts the approach of first identifying the targets and then defining the relationship between them for modelling. The overall research flow is illustrated in *Figure 1*.

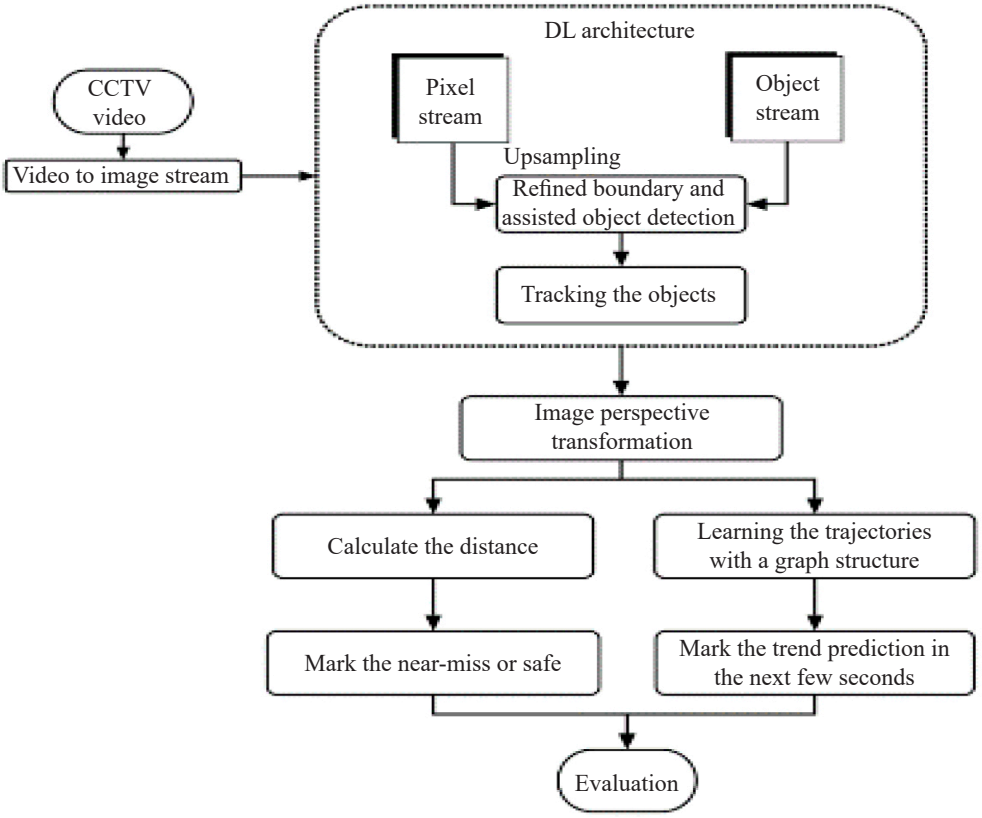


Figure 1 – Overview of the framework

The research steps:

- Extraction of data captured by CCTV cameras for annotation and pre-processing.
- Modelling to identify and track target objects: vehicles, motorcycles and pedestrians. A dual stream is being developed to improve recognition accuracy. Although it is a dual-stream design, the same backbone network was used to improve the detection speed.
- Because there is a certain angle between the CCTV camera and the ground, a graphical transformation is performed to obtain a more accurate target location.
- The probability of a near miss is determined by the distance between targets.
- Predict the probability of a near-miss event in the next few seconds by learning the trajectory of the target's movement.
- The results of each step were analysed.

3.1 Problem statement

Traffic accidents and near misses have similarities; therefore, the former detection methods and approaches can be applied to the detection of near misses. In CCTV surveillance videos, the main problem identified is the detection and tracking of small objects (two or three-wheeled vehicles). Missing instance masks, inconsistent bounding boxes and instance masks owing to inaccurate bounding box positioning during detection significantly degrade the performance of the traffic incident detection system. In *Figure 2*, the red box represents the detection box. The yellow circled part represents the blank area or object outside the detection box. The yellow-circled areas in *Figure 2* show the signs of inaccurate detection.

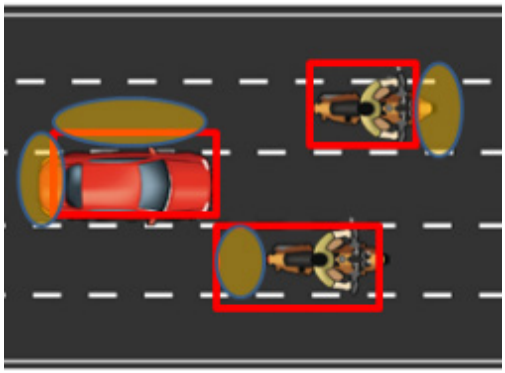


Figure 2 – The problems of the object detection process

This work proposes a new idea aimed at training and optimising the detection and prediction process to improve the near-miss event judgement and prediction performance. The objective of the proposed method is to optimise the bounding box and biases of the detection neural network.

In the object-detection stage, a pixel stream is introduced to improve the confirmation step of the bounding box. The acceleration parameter component was introduced in the IPM [22] to improve the accuracy of the near-miss judgement. The trajectory prediction task was based on an optimal detection and tracking method. Issues that still need to be addressed include dataset cleaning and pre-processing.

3.2 Data augmentation

The experimental dataset was relatively simple and was obtained from only one city. Therefore, increasing the number of extractable features can improve the accuracy of the Convolutional Neural Networks (CNNs). At the same time, the real-time detector and near-miss judgement would become more accurate. The purpose of data augmentation is to increase image variability so that the object detection model can be more robust to different environments. Therefore, effective data augmentation methods can enhance certain attributes in the model, such as expanding the perceptual field, incorporating attention mechanisms and improving feature integration capabilities.

The number of samples is an important parameter for improving the accuracy of the algorithm. Whether it is a traffic accident or near miss, the sample size is usually determined by the time and number of observations. Depending on the characteristics of the video data, some data enhancement and augmentation methods can solve the image blur and data sparsity problems. The training process used in this experiment was a mosaic image-enhancement method. The use of mosaic data enhancement allows the dataset to be enriched and improves the robustness of the model. First, the use of four random images, randomly scaled and then randomly stitched together, adds many small targets and significantly increases the diversity of the data. Second, mixing four images with different semantic information allows the model to detect targets outside the regular context. Therefore, in this study, the number of variables (video frames) added was set to 1000, and data augmentation was performed in the object detection phase.

3.3 Object detection

Object detection [23] is a type of image segmentation based on the geometric and statistical features of the target. It uses bounding boxes to label the target locations. In recent years, many deep-learning-based object detection algorithms have emerged in the field of computer vision, among which one-stage target detection

algorithms, such as Single Shot MultiBox Detector (SSD) and You Only Look Once (YOLO), have become the focus of research owing to their high speed and accuracy. This study was based on the YOLOv7 multitasking model, which performs both object detection and instance segmentation at the cost of a small increase in computation and memory.

The ultimate goal of this research was to determine and predict the probability of near-misses on roads. Therefore, the accuracy of the target bounding box and overlap with the object have a significant impact on the assessment results. Existing object detection algorithms have a situation in which the bounding box is either larger or smaller than the target. Therefore, the object detection stage performs boundary detection by jointly optimising the instance segmentation.

First, the bounding box is used to promote instance segmentation, that is, the outer bounding box of the instance mask is used to constrain the instance segmentation, and the location centre of the bounding box is used as the clustering centre in the segmentation stage. This improves the boundary box accuracy of the instance mask and aggregation speed. Second, instance segmentation was used to facilitate the boundary box, and the boundary box probability and mask boundaries were combined to infer the boundary boxes to suppress non--edge similarity and obtain accurate location frames [24].

Combination strategy refers to the process of combining image segmentation and object detection to form a hybrid algorithm. It is hybridised from pixel and object streams to achieve improved performance in terms of detection capabilities and better accuracy. The specific process and description are presented in the following *Figure 3*.

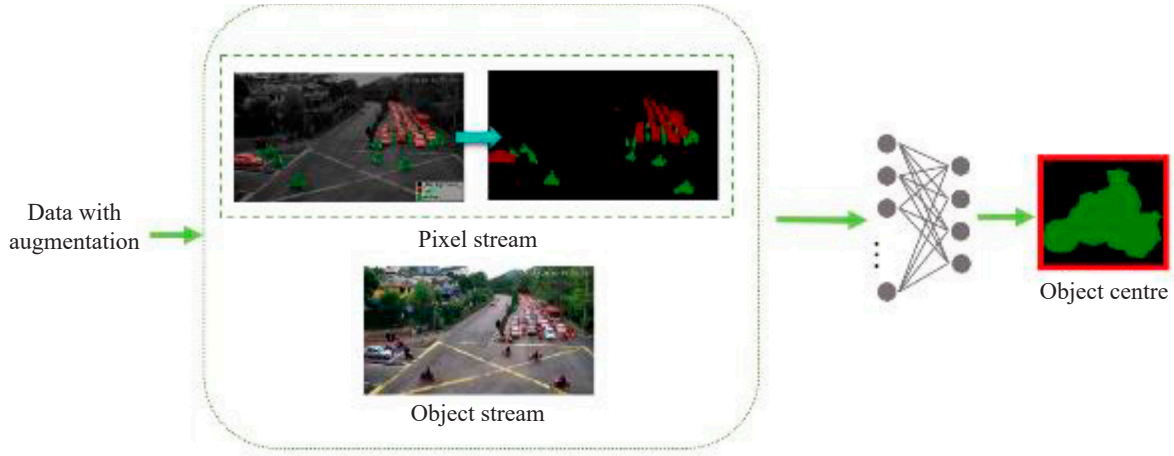


Figure 3 – The framework of the detection process

To achieve a low computational cost, the same backbone was shared during the detection process. The object stream focuses on completing the target detection task and the pixel stream focuses on completing the segmentation task. First, the object stream was used to obtain the target bounding box. Its centroid is used to define the centre of each cluster during image segmentation for end-to-end training, while suppressing the noise far from the target centre. Second, the target object mask is obtained in the pixel stream stage. It can be used to determine the probability that the target belongs to the bounding box and to obtain a more accurate bounding box. In the object detection task, this approach does not significantly increase the computational complexity and further improves detection accuracy.

Poly-YOLO [25] uses the same idea, building on the original idea of YOLOv3 and eliminating two of its weaknesses: a large number of rewritten labels and inefficient anchor point assignment. Poly-YOLO reduces the number of rewritten labels using stepped upsampling to aggregate features in the SE-Darknet-53 backbone via a super-column technique, thus reducing the number of rewritten labels and producing a high-resolution single-scalar output. Compared to YOLOv3, Poly-YOLO has only 60% trainable parameters but a 40% improvement in mAP. In addition, Region-Based Convolutional Neural Network with Mask (Mask-RCNN) [26] combines instance segmentation and target detection. However, the backbone network can also be replaced to increase the detection speed and accuracy of network analysis [27]. This project is based on object detection in

traffic scenes, which has high accuracy and speed requirements; therefore, YOLOv7 [28] is used in the target detection phase of this project. The YOLOv7 network is shown in *Figure 4*.

The main optimisation directions of the current target detection are faster and stronger network architecture, more efficient feature integration methods, more accurate detection methods, more accurate loss functions, more efficient label assignment methods and more efficient training methods. The instance segmentation part of YOLOv7 is based on Facebook Detectron2 (DET2) [29], which allows object detection and instance segmentation to be performed simultaneously, resulting in a more appropriate object detection framework.

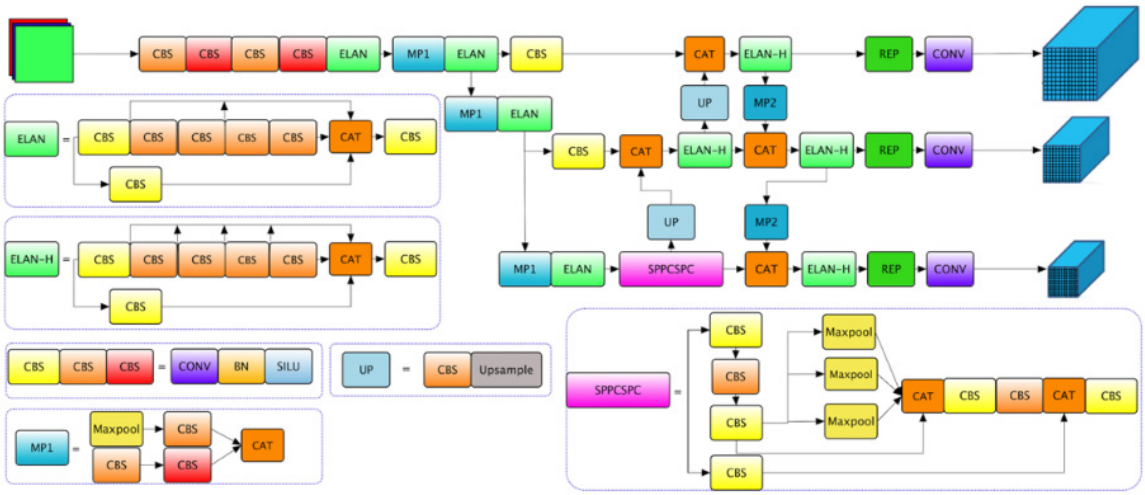


Figure 4 – The network of the YOLOv7

YOLOv7 is a highly accurate real-time detector with fewer parameters and less computation than the current optimal model, with faster inference and higher detection accuracy. The two necessary conditions for traffic monitoring are high speed and accuracy, so this paper proposes an improved version of the YOLO object detection algorithm and applies it to the monitoring of near-miss traffic events. YOLOv7 consists of three parts: input, backbone and head. First, the image is pre-processed through a series of data enhancement operations in the input section and then aligned to a 640×640 RGB image, which is fed into the backbone network. Based on the output of the three layers in the backbone network, three feature maps of different sizes are output in the head layer, and the image is predicted by the RepVGG block and Conv, and the final result is output. In *Figure 4*, the backbone network part of the network model is mainly composed of the convolutional layer CBS (Conv+BN+SiLU), the ELAN module and the MP module. The specific functions are:

1) Backbone

The CBS module consists of a Conv layer, a BN layer and a SiLU layer. The SiLU activation function is a variant of the Swish activation function, and the *Equations 1 and 2* are as follows:

$$silu(x)=x\cdot sigmoid(x) \tag{1}$$

$$swish(x)=x\cdot sigmoid(\beta x) \tag{2}$$

There are three colours in the CBS module, and the three colours represent that they have different convolution kernels (k) and stride sizes (s). The lightest colour, which is a 1×1 convolution with a stride size of 1 and is mainly used to change the number of channels. The slightly lighter colour, which is a 3×3 convolution with a stride of 1 and is mainly used to extract features. The darkest colour, which is a 3×3 convolution with a stride of 2, is mainly used for downsampling.

The ELAN module is an efficient network structure that allows the network to learn more features and be more robust by controlling the shortest and longest gradient paths. It has two branches, the first of which goes through a 1×1 convolution that performs a change in the number of channels. The second branch first goes through a 1×1 convolution that does the channel count change, and then four 3×3 convolution modules that do the feature extraction. Finally, the four features are superimposed to produce the final feature extraction result. Depending on the number of outputs selected in the second branch, there are ELAN and ELAN-H modules. The ELAN module is a total of four branches for fusion, with twice as many output channels as before, and

is used for the feature extraction stage in the backbone. The ELAN-H module is a fusion of six branches with twice as many channels as before, as four of the branches have only 1/4 of the number of channels of the previous input, and is used in the FPN and PAN structures in the head structure.

The MP module has two branches that are used to perform the downsampling. The first branch goes through max-pooling to achieve the downsampling and then a 1×1 convolution to perform the channel count change. The second branch goes through a 1×1 convolution to do the channel count change and then a 3×3 convolution kernel with a stride size of 2 to do the downsampling. Finally, the results of the first branch and the second branch are added together again to obtain the super downsampling result. The MP1 module is used in the feature extraction stage to achieve downsampling while keeping the dimensionality constant. In the PAN stage, the MP2 module is used to downsample and increase the dimensionality by a factor of 2, allowing direct summation with the FPN results.

The upsample operation completes the upsampling using nearest neighbour interpolation and is used in the FPN to implement a scaling operation for higher level feature maps.

2) Neck & Head

SPPCSP obtains different perceptual fields by max-pooling to accommodate images of different resolutions. As can be seen in the branch structure, the first branch undergoes four max-pooling operations with convolution kernels of sizes 1×1, 5×5, 9×9, and 13×13, respectively. Four different perceptual fields are used to distinguish between large and small targets.

The REP (Repvgg_block) module is divided into two parts, one for training and one for deployment.

The training module has three branches. The top branch is a 3×3 convolution for feature extraction. The middle branch is a 1×1 convolution, which is used for feature smoothing. The last branch is identity. The inference module contains a 3×3 convolution with a stride of 1, which is converted from the training model re-parameterisation. In the model re-parameterisation process, the 1×1 convolution is converted to a 3×3 convolution, and then a matrix addition, i.e. a matrix fusion, is performed. Finally, the weights are summed to obtain a 3×3 convolution. The idea of structural re-parameterisation allows the multi-branch structure, which gives the model high performance in the training phase, to be converted into a one-way model for the inference network, which can effectively increase speed and save memory, giving the network an efficient inference speed.

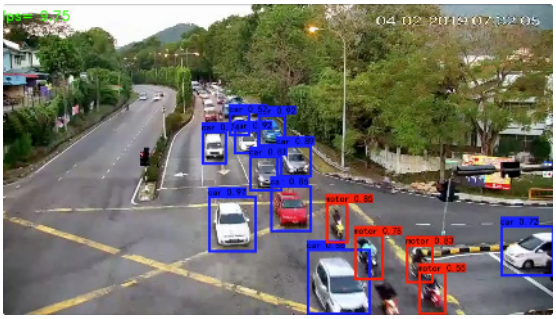
In the model training phase, transfer learning was applied to the backbone algorithm. The backbone of the trained large-scale network is then extracted. These backbones have been trained on a large number of images and exhibit good robustness and feature extraction effects. Other researchers can maintain the weights of the backbones and then apply them to their models.

The model was trained on both the POL37 instance segmentation and object detection datasets. In the object detection phase, image information is acquired by YOLO to detect the presence of vehicles in the image or video and locate them for recognition. The structure is shown in *Figure 5*.

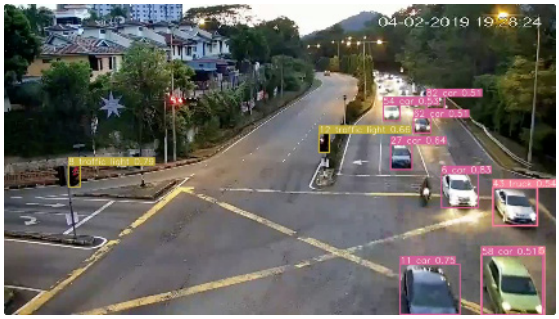
By validating the same POL37 dataset, the detection results of different YOLO versions are shown in *Figure 5*. *Figure 5a* shows the detection structure of YOLOv4, which shows a large gap between the detection frame and the object. *Figure 5c* shows the detection results of Poly-YOLO, which indicates that the area of the mask does not cover the object well and even exceeds the area of the detection frame. *Figure 5d* shows the results of the YOLOv5 semantic segmentation. Although the detection accuracy improved more than in the last two versions, the speed was still slower than that in YOLOv7 (*Figure 5b*). From the verification, the semantic segmentation technique has a limited effect on improving object detection accuracy. Therefore, in *Figure 5e*, YOLOv7-mask chooses the instance segmentation technique to improve the accuracy of the object detection frame, and YOLOv7 has the shortest computation time. As seen in the output, both cars and motorcycles are small objects for the CCTV view, and it is easy to see that YOLOv4-YOLOv7 has good performance in detecting small targets.

3.4 Object tracking

Object tracking [30] must solve several difficult problems such as appearance deformation, lighting changes, background similarity interference, fast motion and motion blur. In the CCTV surveillance video, each frame contains multiple targets of the same type, but with different IDs. Therefore, a multitarget tracking algorithm is required at this stage.



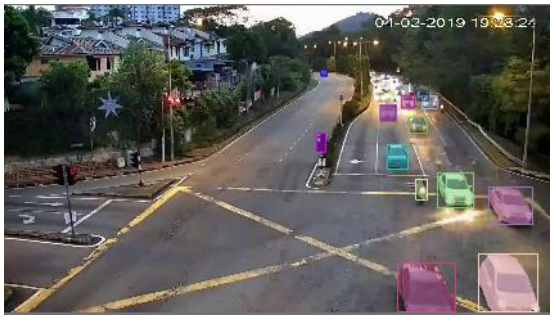
a) YOLOv4



b) YOLOv7

c) Poly-YOLO

d) YOLOv5-Semantic



e) YOLOv7-mask

Figure 5 – The results of the road user detection

The main task of Multiple Object Tracking or Multiple Target Tracking (MOT or MTT) is to simultaneously locate multiple objects, maintain their IDs and record their trajectories. Multitarget tracking relies on attitude estimation, motion detection and behavioural analysis of the target [31]. Target objects in CCTV surveillance videos face two problems: target deformation (partial occlusion or overlap) and fast motion. These situations lead to the target loss and boundary effects, resulting in poor classifier discrimination. The MOT must also deal with more complex key problems (frequent occlusion, trajectory initialisation and termination, similarity in appearance and multi-object interaction) [32]. To increase the proportion of real samples detected, larger detection patches and smaller filters are used, with the aim of achieving a better tracking effect and higher FPS.

Deep Simple Online Realtime Tracking (SORT) [33] is an algorithm commonly used in MOT that introduces epistemic features based on SORT and can handle the occlusion problem better. For the appearance branch, the StrongSORT algorithm [34] uses a stronger appearance feature extractor: Batch Normalisation Neck (BN-Neck) [35] to extract more discriminative features.

In this study, the StrongSort algorithm was used to perform a generalised intersection of unsuccessful matching trajectories and YOLOv7 detection results for association matching, which can improve tracking accuracy as much as possible. In essence, the StrongSort is able to correlate short trajectories with full trajectories to compensate for the missing detections.

a) Road 1 *b) Road 2*

Figure 6 – The results of the road user tracking

In *Figure 6*, the number in the upper-left corner of the detection box is the tracking ID, which uniquely identifies an object in the test video. The purpose of tracking is to record the trajectory of the object, which is also used for training and predicting the position of the object in the next 10 frames, to determine in advance whether a near-miss event will occur.

3.5 Near-miss judgement

Near-miss traffic events typically include two dimensions: spatial distance and avoidance behaviour. Owing to the limitations of the dataset, the discussion in this study focuses on near-miss events at spatial distances. A perspective transformation of the position of the target object in the image is required before analysing the relationship between the target objects. Perspective transformation is based on the law of object image projection.

The image is distorted because of the oblique angle between the CCTV viewing angle and the ground. Therefore, images are usually corrected using inverse perspective mapping (IPM) [36]. IPM uses the frontal view as input to produce a top-down view of the scene by mapping the pixels to another 2D coordinate frame, also known as a bird's eye view. In IPM, the transformation relationship between the before- and after-perspective images can be represented by a 3×3 transformation matrix, which can be obtained from the coordinates of the four corresponding points in the two images. Therefore, perspective transformation is also known as a "four-point transformation" [37].

However, in most CCTV surveillance systems, the camera is fixed and uses a fixed focal length. Under these conditions, the distortion of the target is relatively small unless wide-angle shooting is used. This provides better conditions for measuring the target distances and geometric parameters. Geometric methods can therefore be used to determine the size of the target in the image and the distance between targets, so that surveillance images can be scaled and analysed for near-misses. In addition, the length and width of the target in the image can be assumed to vary proportionally with the distance between the target and the camera, as long as the CCTV camera is not a wide-angle lens with a large deformation. The target speed and angle can then be added as parameters in the IPM to obtain a more accurate target position. The distance between objects was chosen as the basis for judging near-miss events. It is defined as follows:

$$a_{sim,t}^{ij} = 1 / \left\| v_t^i - v_t^j \right\|_2 \tag{3}$$

In *Equation 3*, $a_{sim,t}^{ij}$ denotes the similarity, v_t^i and v_t^j denote the linear distance between the vectors of the two vehicles. This similarity is the distance relationship between the target objects, as described in the previous paragraph. The closer the similarity, the higher the similarity. When the similarity is higher and a certain threshold is reached, the possibility of a near miss occurs.

Normally, the CCTV camera is 6 metres above the ground at an angle of 45 degree. Therefore, the parameters of the IPM algorithm are set according to these two data and the calculation result of *Equation 5*, and transformation inversion is performed on the images of the target dataset. The results are shown in *Figure 7*. *Figure 7a* shows the original image and *Figure 7b* shows the result of the inverse transformation. The transformation process is shown in *Equation 4,* which is generally the relationship between pixels and image planes, the projection relationship and the camera-world relationship. The *u* and *v* denote the number of columns and rows of

pixels in the array type, that is, the image coordinates in the pixels. The x and y denote the image coordinates in millimetres.

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{d_x} & 0 & u_0 \\ 0 & \frac{1}{d_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4}$$

$$a = \frac{S_2 - S_1}{t^2} \tag{5}$$

a) Original image *b) The result of the IPM*

Figure 7 – The result of the IPM

The position and acceleration of the object must also be determined. After the graphical transformation, the displacement of the object is calculated frame-by-frame. The acceleration of the current target object is calculated using the *Equation 5* and the result is used as one of the parameters to determine whether a near-miss event will occur: The design of the near-miss algorithm is shown in *Algorithm 1*. The experimental results are shown in *Figure 8.*

Algorithm 1 – Near-miss Judgement

Step 1: Initialisation
Load YOLO-generated information
Select the bounding box with the target
Select the content inside the bounding box as the potential target
For the content inside the bounding box **do**
Run DeepSort
    Initialise the label for each area
Calculate the speed for each target
    If targets in moving
        Calculate the acceleration for each target as a parameter
    Else if the target in stop
        Skip

Step 2: Calculation
Function IPM
Get central point for each target
For each target
    Judgement near-miss events
    Case 1 (Distance & The acceleration)
        1: arr[i]← target_center point[i]
        2: arr[j] ←the target_center points[j] around i
        3: if arr[j] ・weight(The acceleration[j]) - arr[i] ・(The acceleration[i]) ≤ (width[i] + width[j])/2 + 50cm
        4: set near-miss
        5: else set safe

6: end if

Case 2 (PICUD & The speed)

1: v[i]← velocity of the leading car[i]
2: v[j] ←velocity of the following car[j]
3: S ← distance between car[i] and car[j]
4: Calculate the PICUD with driver's reaction time & deceleration rate to stop
5: if PICUD ≤ (width[i] + width[j])/2 + 50cm
6: set near-miss
7: else set safe
8: end if

Case 3 (PSD & The speed)

1: RD← remaining distance to the potential point of collision
2: MSD ←Minimum acceptable stopping distance
3: Calculate the PSD by RD divide MSD
4: if PSD ≤ the threshold
5: set near-miss
6: else set safe
7: end if

a) Road 1 *b) Road 2*

Figure 8 –The near-miss judgement of the road users

3.6 Trajectory prediction

In the previous section, the distance between two real objects was detected and a distance less than a certain distance was counted as a near miss. This section focuses on the problem of predicting trajectories of moving objects. Predicting the motion of dynamic objects can avoid a number of risky behaviours for both humans and self-driving cars. The core problem of trajectory prediction is to predict the trajectory of dynamic objects, such as a person, car or motorcycle. In constantly changing environments, prediction is essential for smooth and safe path planning. The most common dynamic objects on roads are cars and motorcycles. Therefore, predicting the trajectories of cars and their motorcycles is essential for navigation, planning and human-computer interaction tasks. However, predicting human and vehicle motions is difficult because the trajectories of vehicles change as people voluntarily apply causal forces and continuously adjust their paths as they navigate around obstacles to achieve their goals. Part of this complex planning process is internal, making it difficult to predict the trajectory of a vehicle from observations. Therefore, in addition to past motion history, many other aspects must be considered, such as potential intended targets, other moving objects in the scene and social behaviour patterns. In this study, a combined Kalman filter was used to learn the trajectory of a car or motorcycle over a period of time and predict its trajectory over the next 10 frames. Predicting the trajectory of motorcycles is crucial for identifying near misses.

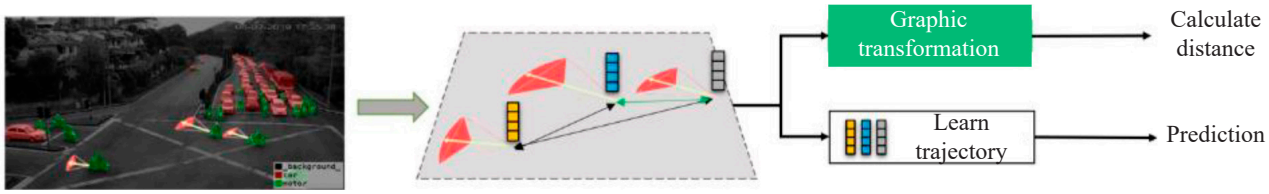


Figure 9 –The framework of detection and trajectory prediction for near-miss event

The Kalman Filter (KF) algorithm is a time-domain discrete autoregressive optimisation-based algorithm that is widely used in dynamic target position, velocity and other applications. The KF algorithm is an autoregressive optimisation algorithm based on time domain dispersion. Let the target state vector be y_t, and the observed variables are z_t according to the Kalman filter algorithm, there is the following relation as shown in *Equation 6 and 7*:

$$y_t = Ay_{t-1} + w_t \tag{6}$$

$$z_t = Hy_t + v_t \tag{7}$$

where A is the target state transfer matrix representing the transfer relationship between the states of the system from moment t-1 to moment t. H is the target observation matrix, w_t and v_t are the process noise and observation noise respectively satisfying the normal distributions $w_t \sim N(0, Q_t)$, $w_t \sim N(0, R_t)$.

The Kalman filter algorithm solves for the target state vector by iteratively updating through a feedback loop consisting of two update steps: a prediction step and an update step. In the prediction step, the Kalman filter algorithm uses the state vector of the target at the previous moment to predict the state of the target at the next moment (*Equation 8 and 9*).

$$\hat{y}_t^{'} = A\hat{y}_t - 1 \tag{8}$$

$$\hat{P}_t^{'} = A\hat{P}_{t-1}A^T + Q_t \tag{9}$$

In the update step, the Kalman gain K_t must first be computed and then the target state estimate $\hat{y}_t$ at time t and the covariance matrix $\hat{P}_t$ must be corrected by the observed variable z_t:

$$K_t = \hat{P}_t^{'} H^T (H\hat{P}_t^{'} H^T + R_t)^{-1} \tag{10}$$

$$\hat{y}_t = \hat{y}_t^{'} + K_t(\hat{z}_t - H\hat{y}_t^{'}) \tag{11}$$

The results of the experiments are shown below.

a) Road 1 *b) Road 2*

Figure 10 –The trajectory prediction of the road users

As shown in *Figure 10*, each blue dot represents the predicted position of the trajectory for the next frame. The tendency of the moving vehicle is accurately determined. This is because vehicles travel along a fixed road direction. The longer the trajectory learning time, the more accurate is the trend prediction. Therefore, the distance between the predicted trajectories can be calculated and the probability of a near-miss event occurring in the future can be obtained.

4. RESULTS AND DISCUSSION

In this study, the results were obtained from a series of test experiments, which are explained in the following sections. All experiments were performed on the same platform to ensure a fair comparison. The platform is a desktop computer with a Core i5 2.40 GHz processor and an NVIDIA Quadro P5000 graphics card with 32 GB of memory. The algorithm implementations were compiled using Python 3.8 on Windows 10 Home Premium SP1.

This section presents an evaluation of the performance measures and evaluates three target tasks. The first was an improved object detection task, the second was a near-miss judgement task and the third was a trajectory prediction task.

4.1 The proposed data

POL 37 video data were used in this study. POL 37 data were obtained from Majlis Bandaraya Pulau Pinang (MBPP), which is located on the island of Penang. There are no records or data on near misses in POL 37. Before the experiment, we downloaded the pre-trained model using the COCO dataset. If we load the pre-trained model with the COCO dataset, we can directly obtain the object detection parameters from this model as the initial parameters. This improved the correct object detection rate and reduced the training time. In addition, this experiment mainly selected two sections of POL37 video data within the experiment: Jalan Lim Chwee Leong Road (Road 1) and Jalan Ke Jawa-Lebuhraya Tun Dr Lim Road (Road 2) with 1280×720 quality.

We installed and used the FFmpeg Builds software to extract images from videos. To ensure the quality of the final sample, LabelImg and Labelme were used to label and manually check the entire sample to eliminate errors in the primary processing. LabelImg was used to label the target box and the type of each target object. Labelme was used to label the specific shape and type of each target. The distribution of the final label is shown below, as there are fewer pedestrians on the road, accounting for only 2.07%. Vehicles (including cars, trucks and buses) and motorcycles accounted for 70.11% and 27.82%, respectively, and motorcycles were studied as small targets for recognition; therefore, when testing the strengths and weaknesses of the algorithm, a period with a high number of motorcycles was mainly selected for testing.

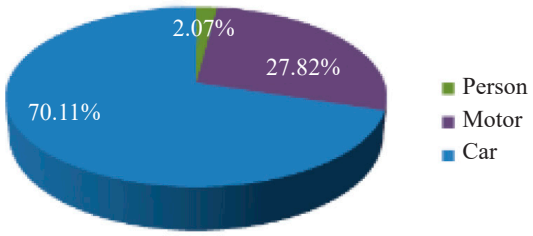


Figure 11 – The POL37 dataset statistics

4.2 Evaluation of Detection Results

The purpose of object detection is to construct a classification function and positioning model, assign the data object to a given category by the classifier, and then delineate the coordinate position of the detected object. The main evaluation indicators were the average precision (AP) and mean average precision (mAP). *Table 1* shows the metric results obtained with the object detection algorithm for different object classes with high detection accuracy.

Table 1 – The experimental results with the POL37 dataset based on the proposed algorithm

Class	Frames	Labels	Precision	Recall	AP@0.5
Car	400	26032	0.868	0.920	0.920
Person	400	26	0.849	0.955	0.941
Motor	400	6848	0.889	0.941	0.981
All	400	32936	0.869	0.939	0.947

The detection rates for different road users are shown in *Table 1*. When the confidence level is set to 0.5, the AP value for car is 92%, for person 98.1% and for motorcycle 98.1%. The recognition of people and motorcycles on a 640?640 image is a small target recognition and the recognition accuracy is higher than 90%, which fully proves the effectiveness of the algorithm proposed in this paper.

Table 2 shows the metric results for the POL37 dataset based on different object detection algorithms. The detection results are compared with those of the model proposed in this paper to verify the superiority of the

detection performance of the algorithm in this paper. The mAP value is used as the evaluation metric for the model.

Table 2 – The metrical results with the POL37 dataset based on different object detection algorithms

Model	mAP@0.5^{val}	mAP@0.65^{val}	mAP@0.8^{val}	mAP@0.95^{val}
YOLOv3	0.841	0.687	0.477	0.213
Poly-YOLO	0.894	0.699	0.495	0.302
YOLOv4	0.925	0.734	0.611	0.407
YOLO5m	0.932	0.783	0.654	0.497
Ours	0.947	0.824	0.701	0.598

The above *Table 2* shows that the mAP value of the proposed algorithm is the highest, which is 10.6%, 5.3%, 2.2% and 1.5% higher than that of mAP@0.5 for YOLOv3, Poly-YOLO, YOLOv4 and YOLO5m network models respectively. The results of the comparison experiments show that the algorithm in this paper has the highest accuracy in detecting road users in the POL37 dataset.

Figure 12 shows that the YOLOv7 mask has the shortest computation time for the same test video in the different YOLO versions and that the advantage of the YOLOv7 mask becomes more apparent as the video length increases.

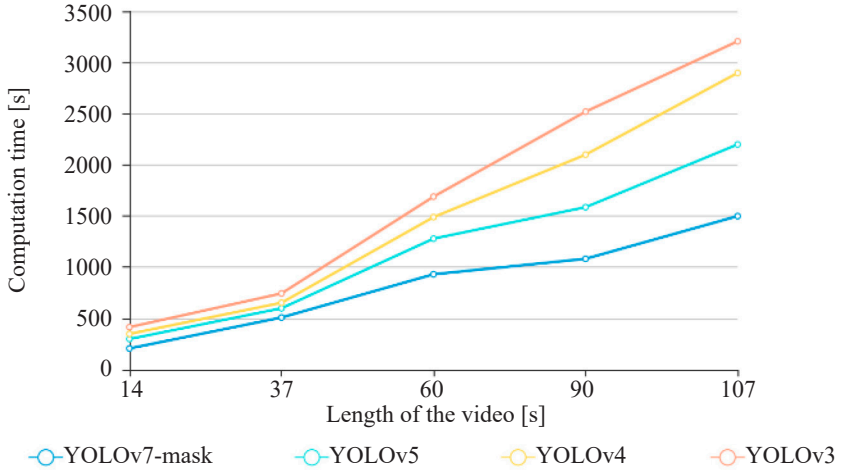


Figure 12 – Comparison of computation time in different visions of YOLO with the same video

In the experiment, videos with a duration of 37 to 107 seconds and 30 frames per second are taken from the CCTV video to perform the experiments. If the length of the video is increased, the running time will be increased. Thus, it is obvious that the computation time of the improved algorithm in this paper is shorter for the computation process when performing the image process.

4.3 Evaluation of trajectory prediction

To assess the quality of forecasts, they should be compared using the same metrics. In the literature [38] it is pointed out that most of the current trajectory forecasts are evaluated based on accuracy. The accuracy-based evaluation metrics are the Average Displacement Error (ADE) and Final Displacement Error (FDE). As shown in the *Figure 13*, ADE and FDE were used to evaluate the similarity between the predicted trajectory (red) and true trajectory (blue). ADE is defined as the average Euclidean distance over all estimated positions in both the predicted and ground-truth trajectories. The FDE is the Euclidean distance between the predicted and ground truth positions at the final destination.

$$ADE = \frac{1}{T_F} \sum_{T=T_p+1}^{T^p+T_F+1} \sqrt{(x_i - x_i^{GT})^2 + (y_i - y_i^{GT})^2} \tag{12}$$

$$FDE = \sqrt{(x_{T_p+T_F+1} - x^{GT}_{T_p+T_F+1})^2 + (y_T - y^{GT}_{T_p+T_F+1})^2} \tag{13}$$

$$ADE_{all} = \frac{\sum_{n=1}^{N} ADE_n}{N} \tag{14}$$

$$FDE_{all} = \frac{\sum_{n=1}^{N} FDE_n}{N} \tag{15}$$

where (x_i, y_j) and x_i^{GT}, y_i^{GT} are the ground truth value and estimated vehicle position at future time T, respectively. T_p denotes the past trajectory sequence length and T_F denotes the future trajectory sequence length.

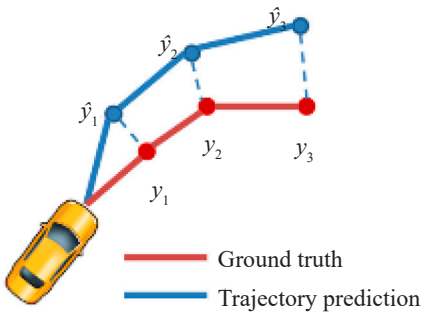


Figure 13 – The description of trajectory ground truth and prediction

In this experiment, 20 trajectory sequences under unit time are taken as a group, the first 10 moments are taken as historical trajectory data, the model outputs the trajectories of the last 10 moments, and ADE and FDE are calculated according to the output predicted and true values. The detailed experimental results are shown in the *Figure 14* and *Table 3*.

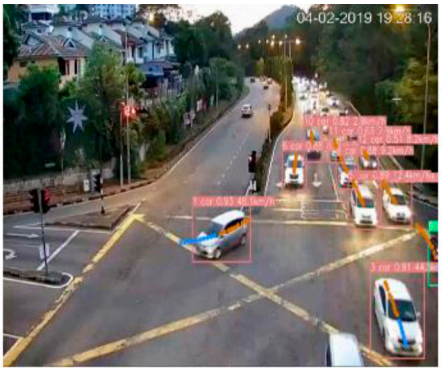
Figure 14a, 14c and 14e show the ground-truth trajectory (orange) and predicted trajectory (blue) for all targets in the detected frames at f=17, 175, and 166, respectively. *Figures 14b, 14d, and 14f* show the ground-truth trajectory (orange) and the predicted trajectory (blue) for a car with tracking ID 1, a motorbike with tracking ID=5, and a person with tracking ID=21, respectively. The values of the horizontal and vertical coordinates are based on the position of the target object at the pixel point. As can be seen in *Figures 14a and 14b*, the speed of the vehicle was accurately calculated and the predicted trajectory was more accurate.

Next, we evaluated the effectiveness of tracking by analysing the prediction results of each segment of the experimental data and calculating the ADE and FDE values. The results are shown in *Table 3*.

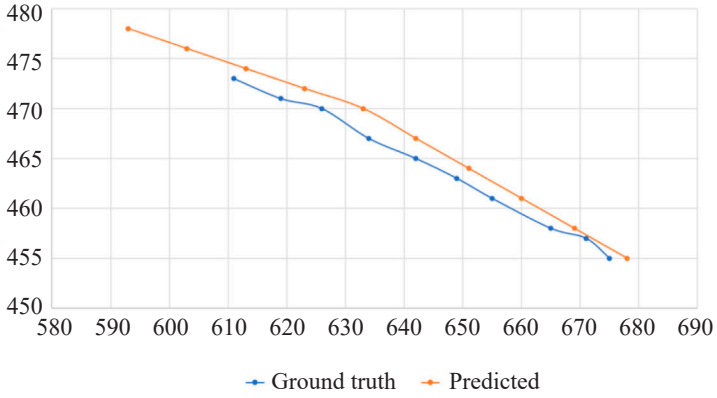
Table 3 – The trajectory prediction analysis of proposed model

Road		ADE	FDE
Road 1	Car	8.5	13.2
	Motorcycle	14.6	18.3
	Person	/	/
	AVG	11.6	15.8
Road 2	Car	13.3	22.3
	Motorcycle	18.2	32.4
	Person	11.1	21.1
	AVG	14.2	25.3

This experiment used historical observed trajectory data to predict the future trajectory of a target object. The experiment was divided into ten time periods. The observed trajectories of the first 10 moments were

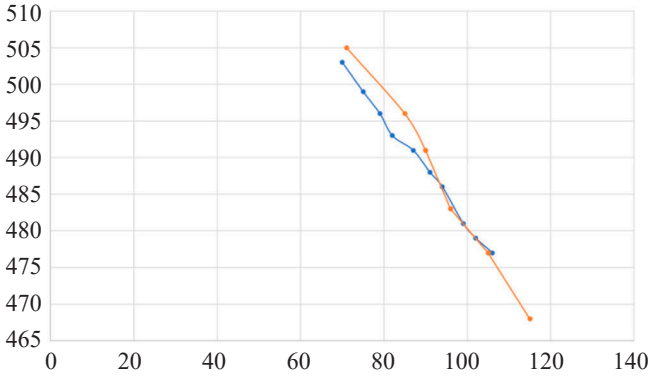


a) The ground truth and predicted trajectory of the road 1 (object is the car)

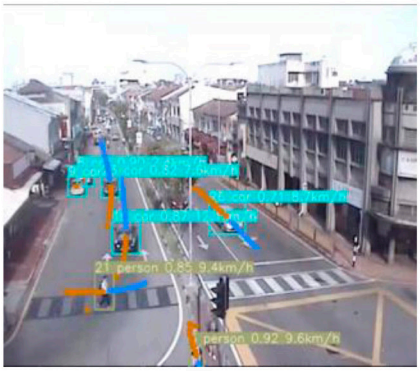


b) The ground truth and predicted trajectory (Time/Frame = 17, Tracking ID = 1, ADE = 8.5, FDE = 8.3)

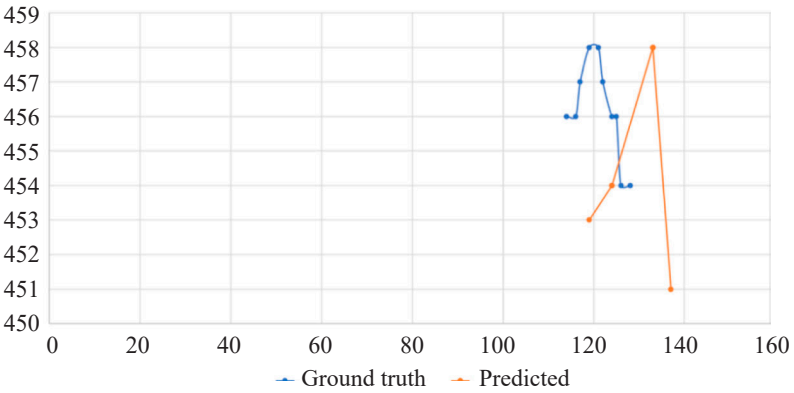
c) The ground truth and predicted trajectory of the road 2 (object is the motorcycle)



d) The ground truth and predicted trajectory (Time/Frame = 175, Tracking ID = 5, ADE = 10.3, FDE = 12.7)



e) The ground truth and predicted trajectory of the road 2 (object is the person)



f) The ground truth and predicted trajectory (Time/Frame = 166, Tracking ID = 21, ADE = 7.9, FDE = 9.5)

Figure 14 – Example of predicted and ground truth trajectories

input to predict the future trajectories of the next 10 moments of the ROAD USER. Twenty samples with ADE and FDE as evaluation metrics were selected for model evaluation. In *Table 3*, the test video pixels for Road 1 were 1280×720 pixels, and the test video pixels for Road 2 were 704×640 pixels. The results shown are all the pixel values. The prediction accuracy from the FDE metric comparison decreased significantly at the 10th time point. By mapping the actual width of the road to the pixel values, the results show that both models have high detection accuracy.

4.4 Evaluation of near-miss judgement

If the objects are connected by red lines, there is the possibility of a near miss between the two targets. *Table 4* lists the number of lines in the videos at different times. Three different periods (each event period of 20

Table 4 – Comparison of different times on different roads

Near-miss events \ Times			7:31:45 - 7:32:05	17:43:53 - 17:44:13	19:28:15 - 19:28:35
Road 1	Car-Car	Near-miss detections	23	1	8
		Error detections	1	0	0
		Accuracy percentage	95.7%	100%	100%
	Car-Motor	Near-miss detections	78	2	20
		Error detections	2	0	0
		Accuracy percentage	97.4%	100%	100%
	Car-Person	Near-miss detections	0	0	0
		Error detections	0	0	0
		Accuracy percentage	100%	100%	100%
	Motor-Motor	Near-miss detections	89	2	13
		Error detections	7	0	1
		Accuracy percentage	92.1%	100%	92.3%
	Motor-Person	Near-miss detections	0	0	0
		Error detections	0	0	0
		Accuracy percentage	100%	100%	100%
			7:29:58 - 7:30:19	16:43:55 - 16:44:23	18:25:01 - 18:25:24
Road 2	Car-Car	Near-miss detections	2	8	13
		Error detections	0	0	1
		Accuracy percentage	100%	100%	92.3%
	Car-Motor	Near-miss detections	3	15	32
		Error detections	1	0	0
		Accuracy percentage	66.7%	100%	100%
	Car-Person	Near-miss detections	0	1	0
		Error detections	0	0	0
		Accuracy percentage	100%	100%	100%
	Motor-Motor	Near-miss detections	3	27	23
		Error detections	0	1	2
		Accuracy percentage	100%	98.9%	91.3%
	Motor-Person	Near-miss detections	0	3	2
		Error detections	0	1	1
		Accuracy percentage	100%	66.7%	50%

seconds duration) of the same road section were selected to obtain the number of near misses between car-car, car-motor, car-person, motor-motor and motor-car separately.

The metric for these near-misses is the distance between the target objects. A near miss was considered to have occurred if the distance between the closest edge of the bounding box for the objects was less than 50 cm. However, as there are cases where the target detection frame may extend beyond the boundary of the target object or where the detection frames of multiple objects overlap, we select their centre point to aid detection.

This means that the distance between the centre points was judged to be more than 2.5 m for vehicles and 1.2 m for motorcycles. In addition, if both targets are stationary, no near-misses are considered. Instead, we calculate whether the target is stationary from its moving speed, and if the current measured speed is 0 km/h, it is judged to be stationary.

Before calculating the target distance, the detection results were corrected using the IPM to ensure that the true centre of the target was found.

To demonstrate that distance can be the first factor determining a near miss, two other indicators were chosen for the experiment. These are the Potential Index of Collision Urgent Deceleration (PICUD) and Proportional Stopping Distance (PSD) [39]. The PICUD [40] indicates the final distance between the involved road user and the road user in front when the latter brakes with maximum braking force. The PSD [41] represents the ratio between the distance left between two road users to collide and the minimum stopping distance for other road users when the road user stops suddenly. The results of the experiments are presented in *Table 5*. The formula is defined as follows:

$$PICUD = S_x^{0j} + \frac{v_{x_j}^2}{2MADR_x^j} - \frac{v_{x_0}^2}{2MADR_x^0} - t_h v_{x_0} \tag{16}$$

$$PSD = \frac{S_x^{0j}}{v_{x_0}^2 / 2MADR_x^0} \tag{17}$$

where s_x^{0j} is the longitudinal distance between two targets. v denotes the speed of the target. $MADR_x^0$ represents the braking capacity of the road users. However, the value of the MADR was set according to different types of road users. The setting criterion is the size of the road user to determine the impact of the vehicle size in the risk analysis. It was set at 3.4 m/s^2 [42]. where t_h is the driver reaction time of the object road user and is set to 1.5 s based on brake research.

Table 5 – Analysis of near-miss events occurring in different time periods

Type \ Time		7:31:45 - 7:32:05	17:43:53 - 17:44:13	19:28:15 - 19:28:35
Road 1	Distance	190	5	31
	PICUD	146	19	105
	PSD	298	115	174
		7:29:58 - 7:30:19	16:43:55 - 16:44:23	18:25:01 - 18:25:24
Road 2	Distance	8	54	70
	PICUD	23	101	122
	PSD	121	154	304

As can be seen from *Table 5*, the trend in the number of near-miss events estimated using PICUD and PSD as indicators of near-miss events was consistent with that estimated using distance as an indicator.

In summary, the main contributions of the algorithm proposed in this paper are as follows:

- A data enhancement algorithm is added in the pre-processing stage to increase the training samples while complicating the feature distribution, which can effectively improve the feature learning ability of the model.
- From *Figure 5*, it can be seen that the target detection frame of the proposed algorithm is more accurate and can be more effective for traffic near-miss analysis.
- From *Figure 5* and *Table 1*, the experimental results prove that this algorithm has an excellent detection effect on small objects.
- From *Figure 12*, the experimental results show that this algorithm has a faster detection speed.
- From *Tables 4 and 5*, the experiments were conducted at different time periods, i.e. under different lighting conditions with high accuracy.

5. SUMMARY AND FUTURE WORK

This study focuses on traffic near-miss events using a dual-stream deep-learning method. This involves the detection and prediction of near-missing events. The primary techniques used were object detection, object tracking and IPM. In the object detection phase, the YOLOv7 mask algorithm was used to maximise the restricted object detection frame to calculate the centre of the object more accurately. However, before calculating the centre of an object, the detected image must be corrected using the IPM technique. After this process, the acceleration of the object movement must be added as a parameter as one of the conditions to determine the near-miss event. Object tracking uses the StrongSort technique to target a specific object, learn its trend to predict the future trajectory of the object and determine whether it will have a near-miss event in the next 5–10 seconds. The study achieved 94.7% mAP values in the object detection phase and 12.9 ADE and 20.55 FDE values in the trajectory prediction phase. All the data structures indicate the high accuracy and feasibility of this test study method. In the future, evasive actions (movement changes to avoid a collision), intersections, turning, rear-end, car-pedestrian, bicycle, etc., meetings, and overtaking/lane change can be studied in addition to this one. We also need to consider the intensity of evasive action. For example, if a vehicle is in the left-turning lane and makes a hard left-turn, it poses a threat to the car coming straight across. However, if the vehicle suddenly brakes, it poses a threat to the vehicle behind it. In the future, we will discuss how to predict the consequences of potential collisions.

ACKNOWLEDGMENTS

The authors express their sincere gratitude and appreciation to the Fujian Provincial Education and Research Foundation for Youths in 2020 (No. JAT201039) for supporting this study.

REFERENCES

[1] Loo BPY, Huang Z. Delineating traffic congestion zones in cities: An effective approach based on GIS. *J Transp Geogr*. 2021;94(6):103108. DOI:10.1016/j.jtrangeo.2021.103-108.
[2] Kitamura Y, Hayashi M, Yagi E. Traffic problems in Southeast Asia featuring the case of Cambodia's traffic accidents involving motorcycles. *IATSS Res*. 2018;42(4):163-170. DOI: 10.1016/j.iatssr.2018.11.001.
[3] Zwetsloot G, Leka S, Kines P. Vision zero: From accident prevention to the promotion of health, safety and well-being at work. *Policy and Practice in Health and Safety*. 2017;15:88-110. DOI: 10.1080/14773996.2017.1308701.
[4] Zheng L, Sayed T, Mannering F. Modeling traffic conflicts for use in road safety analysis: A review of analytic methods and future directions. *Anal Methods Accid Res*. 2021;29:100-142. DOI: 10.1016/j.amar.2020.100142.
[5] Ge J, et al. Accident causation models developed in China between 1978 and 2018: Review and comparison. *Saf Sci*. 2022;148(12). DOI: 10.1016/j.ssci.2021.105653.
[6] Heinrich HW, Stone RW. Industrial accident prevention. *Soc Serv Rev*. 1931;5(2):323-324. DOI: 10.1086/630904.
[7] Terum JA, Svartdal F. Lessons learned from accident and near-accident experiences in traffic. *Saf Sci*. 2019;120(6):672-678. DOI: 10.1016/j.ssci.2019.07.040.
[8] Wu LS, Dong Y, Wu. A case study of road traffic accidents based on Heinrich's accident causation theory. *Chinese Journal of Ergonomics.* 2018;24(2):60-64. DOI: 10.13837/j.issn.1006-8309.2018.02.0012.
[9] Hajjami LE, Mellouli EM, Berrada M. Neural network based sliding mode lateral control for autonomous vehicle. *2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET).* 2020. p. 1-6. DOI: 10.1109/IRASET48871.2020.9092055.
[10] Lhoussain EH, Mellouli EM, Berrada M. Robust adaptive non-singular fast terminal sliding-mode lateral control for an uncertain ego vehicle at the lane-change maneuver subjected to abrupt change. *Int. J. Dynam. Control*. 2021;9:1765-1782. DOI: 10.1007/s40435-021-00771-x.
[11] Lhoussain EH, et al. A robust intelligent controller for autonomous ground vehicle longitudinal dynamics. *Applied Sciences*. 2023;13(1):501. DOI: 10.3390/app13010501.
[12] Kataoka H, et al. Drive video analysis for the detection of traffic near-miss incidents. *Proc. - IEEE Int. Conf. Robot. Autom.* 2018. p. 3421-3428. DOI: 10.1109/ICRA.2018.8460812.
[13] Suzuki T, Aoki Y, Kataoka H. Pedestrian near-miss analysis on vehicle-mounted driving recorders. *Proc. 15th IAPR Int. Conf*. 2017. p. 416-419. DOI: 10.23919/MVA.2017.7986889.
[14] Ospina MH, et al. Extraction of decision rules using genetic algorithms and simulated annealing for prediction of severity of traffic accidents by motorcyclists. *J. Ambient Intell. Humaniz. Comput.* 2021;(12):10051-10072. DOI: 10.1007/s12652-020-02759-5.
[15] Uchida N, Kawakoshi M, Tagawa T, Mochida T. An investigation of factors contributing to major crash types in Japan based on naturalistic driving dana. *IATSS Res*. 2010;(34):22-30. DOI: 10.1016/j.iatssr.2010.07.002.
[16] Chen W, Qiao Y, Li Y. Inception-SSD: An improved single shot detector for vehicle detection. *J. Ambient Intell. Humaniz. Comput.* 2020. DOI: 10.1007/s12652-020-02085-w.

[17] Yang B, et al. A vehicle tracking algorithm combining detector and tracker. *Eurasip J. Image Video Process.* 2020;(1). DOI: 10.1186/s13640-020-00505-7.
[18] Wan J. An efficient small traffic sign detection method based on YOLOv3. *J. Signal Process. Syst.* 2020. DOI: 10.1007/s11265-020-01614-2.
[19] Tran AC, et al. A model for real-time traffic signs recognition based on the YOLO algorithm – A case study using vietnamese traffic signs. *Futur. Data Secur. Eng.* 2019;11814:104-116. DOI: 10.1007/978-3-030-35653-8_8.
[20] Kurniawan J, Dewa CK, Afiahayati. Traffic congestion detection: Learning from CCTV monitoring images using convolutional neural network. *Procedia Comput Sci.* 2018;144:291-297. DOI:10.1016/j.procs.2018.10.530.
[21] Abdel-Aty M, Wu Y, Zheng O, Yuan J. Using closed-circuit television cameras to analyze traffic safety at intersections based on vehicle key points detection. *Accid Anal Prev*. 2022;176:106794. DOI: 10.1016/j.aap.2022.106794.
[22] Bertozzi M, Broggi A, Fascioli A. Stereo inverse perspective mapping: Theory and applications. *Image Vis Comput*. 1998;16(8):585-590. DOI: 10.1016/s0262-8856(97)00093-0.
[23] Zhao ZQ, Zheng P, Xu ST, Wu X. Object detection with deep learning: A Review. *IEEE Trans Neural Networks Learn Syst*. 2019;30(11):3212-3232. DOI: 10.1109/TNNLS.2018.2876865.
[24] Galoogahi HK, Sim T, Lucey S. Correlation filters with limited boundaries. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*. 2015;7(6):4630-4638. DOI: 10.1109/CVPR.2015.7299094.
[25] Hurtik P, et al. Poly-YOLO: Higher speed, more precise detection and instance segmentation for YOLOv3. *Neural Comput Appl*. 2022;34(10):8275-8290. DOI: 10.1007/s00521-021-05978-9.
[26] Zhang Q, Chang X, Bian SB. Vehicle-damage-detection segmentation algorithm based on improved Mask RCNN. *IEEE Access*. 2020;8:6997-7004. DOI: 10.1109/ACCESS.2020.2964055.
[27] Khan MA, Akram T, Zhang YD, Sharif M. Attributes based skin lesion detection and recognition: A mask RCNN and transfer learning-based deep learning framework. *Pattern Recognit Lett*. 2021;143:58-66. DOI: 10.1016/j.patrec.2020.12.015.
[28] Wang CY, Bochkovskiy A, Liao H-YM. *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. 2022. http://arxiv.org/abs/2207.02696.
[29] Pham V, Pham C, Dang T. Road damage detection and classification with Detectron2 and Faster R-CNN. *Proc - 2020 IEEE Int Conf Big Data, Big Data 2020*. 2020. p. 5592-5601. DOI: 10.1109/BigData50022.2020.9378027.
[30] Luo W, et al. Multiple object tracking: A literature review. *Artif Intell*. 2021;293:1-49. DOI: 10.1016/j.artint.2020.103448.
[31] Mangalam K, et al. It is not the journey but the destination: Endpoint conditioned trajectory prediction. *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)*. 2020;12347:759-776. DOI: 10.1007/978-3-030-58536-5_45.
[32] Galoogahi HK, Fagg A, Lucey S. Learning background-aware correlation filters for visual tracking. *Proc IEEE Int Conf Comput Vis*. 2017;2017(10):1144-1152. DOI: 10.1109/ICCV.2017.129.
[33] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. *2017 IEEE International Conference on Image Processing (ICIP)*. 2017. p. 3645-3649. DOI: 10.1109/ICIP.2017.8296962.
[34] Du Y, Song Y, Yang B, Zhao Y. *StrongSORT: Make DeepSORT great again*. 2022. http://arxiv.org/abs/2202.13514.
[35] Luo H, et al. A strong baseline and batch normalization neck for deep person re-identification. *IEEE Trans Multimed*. 2020;22(10):2597-2609. DOI: 10.1109/TMM.2019.2958756.
[36] Bruls T, Porav H, Kunze L, Newman P. The right (angled) perspective: Improving the understanding of road scenes using boosted inverse perspective mapping. *IEEE Intell Veh Symp Proc*. 2019;2019(6):302-309. DOI: 10.1109/IVS.2019.8814056.
[37] Tanveer MH, Sgorbissa A. An inverse perspective mapping approach using monocular camera of pepper humanoid robot to determine the position of other moving robot in plane. *ICINCO 2018 - Proc 15th Int Conf Informatics Control Autom Robot*. 2018;2(Icinco):219-225. DOI: 10.5220/0006930002190225.
[38] Ivanovic B, Pavone M. Rethinking trajectory forecasting evaluation. *ArXiv*. abs/2107.10297. DOI: 10.48550/arXiv.2107.10297.
[39] Mahmud SMS, Luis FL, Hoque MS, Tavassoli A. Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs. *IATSS Research.* 2017;41(4):153-163. DOI: 10.1016/j.iatssr.2017.02.001.
[40] Uno N, Iida Y, Itsubo S, Yasuhara S. A microscopic analysis of traffic conflict caused by lane-changing vehicle at weaving section. *Proceedings of the 13th Mini-EURO Conference-Handling Uncertainty in the Analysis of Traffic and Transportation Systems*. 2002. p. 10-13.
[41] Allen BL, Shin TB, Cooper PJ. Analysis of traffic conflicts and collisions. No. HS-025 846, 1978.
[42] Chen QH, et al. Modeling accident risks in different lane-changing behavioral patterns. *Analytic Methods in Accident Research*. 2021;(30). DOI: 10.1016/j.amar.2021.100159.

杨璐，Ahmad Sufril Azlan Mohamed, Majid Khan Majahar Ali

基于改进的物体检测和轨迹预测的交通冲突分析方法

摘要
尽管基于计算机视觉的方法在评估交通状况方面得到了广泛的利用，但评估和预测

交通中的未遂事件还缺乏研究。此外，大多数目标检测算法在检测小目标物体时效果欠佳。本研究提出了一种结合目标检测、跟踪算法、逆向透视映射（IPM）和轨迹预测机制的方法来评估和预测未遂事件。首先，使用一个实例分割头来提高目标检测阶段的准确性。其次，IPM被应用于所有检测结果，然后根据目标之间的距离探索它们之间的关系，以确定是否存在未遂事件。在这个过程中，目标的移动速度被视为一个参数。最后，卡尔曼滤波器被用来预测物体的轨迹，以确定在接下来的几秒钟内是否会发生未遂事件。在CCTV数据集上的实验显示，检测结果达到了0.94 mAP。除了检测精度的提高，实例分割融合目标检测对小目标检测的优势也得到了验证。因此，该目标检测结果可被用于分析未遂事件的发生概率。

关键字
未遂事件；物体检测；物体跟踪；轨迹预测