Moving Vehicle Tracking and Scene Understanding: A Hybrid Approach

Xiaoxu Liu, Wei Qi Yan, Nikola Kasabov

Auckland University of Technology, 1010 Auckland New Zealand

Abstract In this paper, we present a novel deep learning method for detecting and tracking vehicles within the context of autonomous driving, particularly focusing on scenarios related to vehicle failures. Ensuring the precise identification and monitoring of vehicles is paramount for enhancing road safety in autonomous driving systems. Our contribution involves the introduction of a hybrid Siamese network that merges the capabilities of YOLO models with Transformers. This integration aims to address the limitations of Convolutional Neural Networks (CNNs) in grasping high-level semantic nuances, thereby facilitating accurate detection and tracking of multiple vehicles within a given scene. Beyond this, we also curated the traffic scene dataset, which serves as a resource for training a multi-vehicle tracking model specifically tailored to the unique characteristics of traffic environment.

Keywords: Deep learning · Siamese network · Transformer · Attention module · Vehicle detection and tracking · Scene understanding.

1. Introduction

The comprehension of traffic scenes has emerged as a prominent research area in computer vision and a focal point in artificial intelligence, particularly in light of the progress made in autonomous driving advancements [38, 47]. The existing scholarly literature delves into traffic scene understanding from diverse perspectives, reflects the significance and widespread interest in this subject. Among them, vehicle tracking task stands out as a crucial component within the realm of comprehending traffic scenes.

The evolution of visual object recognition and tracking, from R-CNN to Faster R-CNN, has traditionally relies on a two-stage training process. While this approach improves accuracy, it slows down visual object detection due to the increased factors involved. On the other hand, YOLO models adopt a single-shot grid segmentation approach, where each grid is responsible for recognizing the center, bounding box, and class label of the target simultaneously. This end-to-end framework significantly enhances real-time capabilities, makes YOLO models much efficient, saves over 90% of the space in the YOLO series [1, 26]. At the same time, vehicle tracking using a Siamese network employes this specific type of neural network architecture to identify vehicles within a sequence of images or video frames. The Siamese network is particularly useful for the tasks that require comparing and matching pairs of inputs. In the context of vehicle tracking, Siamese network aims to determine

Corresponding author: Wei Qi Yan

whether a detected vehicle in one frame corresponds to the same vehicle in another frame, effectively to establish the trajectory and movement of a vehicle over time.

However, CNNs excel at capturing localized patterns and features within data, rendering them exceptionally effective for tasks such as image classification and object detection. The utilization of fixed-size convolutional kernels with predetermined receptive fields encounters difficulties in grasping extended dependencies and encompassing global context. This limitation can impede the performance which is confronted with tasks demanding an understanding of relationships among distant elements.

Furthermore, CNNs lack an inherent understanding of positional relationships among a diversity of elements among input data. The reliance on spatial hierarchy of data may prove inadequate for the tasks including sequential data or scenarios where precise positional information holds paramount importance. Additionally, CNNs construct feature hierarchies through the sequences of convolutions and pooling layers. While this process benefits the extraction of hierarchical features, it may not be optimal for the tasks that necessitate intricate modeling of multifaceted interactions and dependencies across various segments of the input.

Therefore, we turn our attention to Transformer. Transformer architecture, particularly Swin Transformer, has demonstrated strong capabilities in traffic scene understanding by improving global feature extraction ability. Swin Transformer combines powerful modeling ability of Transformers with important visual signal priors, including hierarchy, locality, and translation invariance. The design of shifted non-overlapping windows in Swin Transformer reduces computational complexity, which leads to faster speeds compared to traditional method of sliding windows [20].

Moreover, Transformers can tackle the entire input sequence with parallelization, enable them to better understand the global context of the sequential data. This is particularly advantageous for the tasks that require a holistic understanding of the input. Pertaining to vehicle tracking, Transformers explicitly handle with positional information through the addition of positional encodings. This ensures that the model deals with the order and relative positions in a sequence, makes them well-suited for vehicle tracking with the sequential data.

To further enhance visual object detection and target tracking in traffic scene understanding, attention mechanisms are incorporated into deep learning networks. Techniques such as Squeeze-and-Excitation (SE) enable the model to focus on essential channel information by learning adaptive channel weights [10]. The Convolutional Block Attention Module (CBAM) combines convolution and attention mechanisms together to process images from both spatial and channel perspectives [36]. Coordinate Attention (CA) takes into account of both channel and spatial dimensions, allows the model to emphasize on crucial channel information through learned

adaptive weights [9]. The mechanisms contribute to improve local feature extraction abilities, which result in more accurate and efficient performance of the models in traffic scenes [27].

In this paper, we propose a new method for scene analysis to achieve higher real-time multitarget vehicle detection and tracking as well as scene understanding. Our method makes use of visual features from YOLO model and Siamese network, it also combined Transformer and attention module in order to understand high level semantic of the scene as we improve the extraction ability of both local feature and globle feature.

In the remaining parts of this paper, our prior knowledge is reviewed in Section 2; the proposed method is presented in Section 3; our experimental and test results are showcased in Section 4 and conclusion is drawn in Section 5.

2. Literature Review

Over the past decade, deep learning methods have demonstrated strong capabilities in visual object tracking. Conventional object tracking algorithms rely on particle filtering, which necessitates a large number of particles for classifier training, resulting in complex convolutional layers for feature extraction [7]. To enhance accuracy and speed, a number of algorithms have combined deep learning methods with relevant filtering techniques. HCFT utilizes VGG-16 to extract features from Conv3-4, Conv4-4, and Conv5-4, training corresponding correlation filters to locate the target accurately. Similarly, HDT utilizes a combination of multi-layer depth features and correlations, while enhancing the depth from three to six layers and adopting adaptive weight addition [22, 28].

Another network to implement tracking task is Siamese networks, a typical model with the deep learning correlation filtering method, which have been employed to simulate the entire process of related filtering. One branch saves target template information, while the other extracts features in the search region. The merged parts generate the response map while reflecting the target state [3]. Siamese networks excel in one-shot and few-shot learning scenarios that can effectively learn to distinguish between different object instances using only a few examples through making them suitable for tracking tasks where acquiring extensive labeled data is challenging. At the same time, Siamese networks directly learn a similarity metric in the feature space. This characteristic is well-suited for tracking tasks where the focus is on finding the similarity between the target object features and those of the candidate regions.

One of the primary hurdles in target tracking tasks involves the intricate interplay between background elements and objects within complex environments. To tackle this issue, the joint Siamese attention-aware network (JSANet) integrates self-

attention and cross-attention modules, strategically designed to surmount the challenges arising from subtle features and background noise [32]. The self-attention modules introduced in this framework synergize channel and spatial attention mechanisms. The channel attention component accentuates pertinent channel coefficients to spotlight high-scoring channels, whereas the spatial attention module transforms spatial domain data, which ensures precise identification of crucial regions. Moreover, the cross-attention element orchestrates the fusion of contextual dependencies between the target template and the search image through cross-channel attention. This sophisticated approach enables the unveiling correlations between objects with temporal associations. The utilization of Siamese region proposal networks (SiamRPNs) further amplifies this methodology via enabling the prediction of a singular tracking region based on the feature streams that have been modulated by attention mechanisms.

However, most models focus on single-target tracking, while multitarget tracking research progress is relatively slower due to limited datasets and references. Single-target tracking is often employed for short-term image sequences, while multitarget tracking deals with longer videos, involving various appearance, occlusion, and separation of targets. The implementation methods also differ, with single-target tracking prioritizing target relocation, while multitarget tracking focuses on matching detected targets. Multitarget tracking algorithms can be detection-based or non-detection-based, which are further classified into online tracking and offline tracking based on frame processing and utilization of subsequent frames [21].

A multitarget tracking method [31] includes a detector to identify targets in video image space, predicts the position and motion of the targets in the next frame using Kalman filter that calculates the overlapping between detection and prediction boxes using Complete Intersection over Union (CIoU) as a distance measure via the Hungarian algorithm to perform data association between multiple targets. The Hungarian algorithm can effectively handle situations where there are multiple detections and tracks, and the number of detections might not match the number of tracks. It assigns each detection to a track and vice versa, which makes it robust against cases where the number of objects being tracked dynamically. Furthermore, the Hungarian algorithm guarantees finding the globally optimal solution to the assignment problem. This means that it provides the best possible assignment that minimizes the total cost among all possible combinations, ensures high-quality associations between detections and tracks [8, 34].

3. Methodology

Fig. 1 depicts the proposed vehicle tracking model. The implementation of the proposed tracking network involves utilizing a modified SiamRPN subnetwork. In

contrast to the original SiamRPN, the modified SiamRPN sub-network has been integrated with Hungarian algorithm [13]. This integration allows for the advancement of single-target tracking to multitarget tracking.

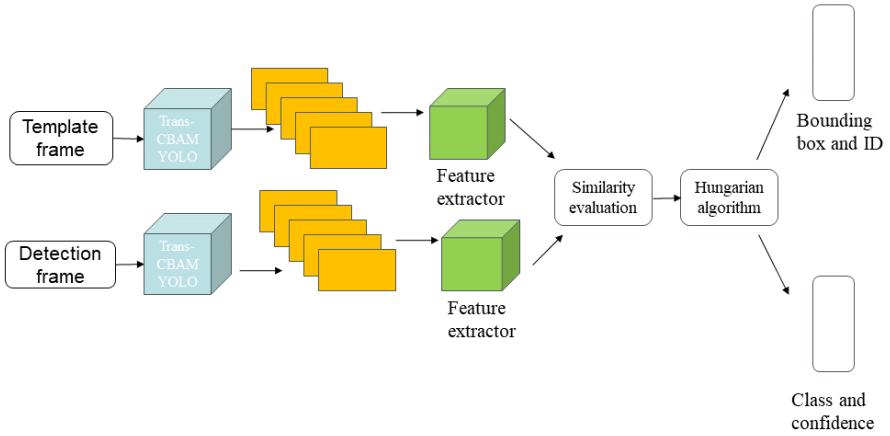

Fig. 1 The vehicle tracking model

The multiobject tracking is different from single-object tracking. The single-object tracking focuses on a single specific vehicle throughout a video sequence. The goal is to follow the movement of a particular vehicle, keep it in focus and provide its trajectory over time. Our proposed multiobject tracking is to detect and track all vehicles presented in the scene, assign unique identities to each vehicle and keep tracking along individual trajectory. The way in this paper is to implement multitarget tracking by using YOLO model combined with attention modules and Transformer for visual object detection in each frame and then use Hungarian algorithms to associate the detected objects across video frames.

For each detected vehicle in the current frame, we calculate the distance between its bounding box and the bounding boxes of all previously tracked vehicles in the previous frame. We make use of Hungarian algorithm to find the best assignment of detected vehicles to previously tracked vehicles based on the distance matrix. The Hungarian algorithm efficiently solves the assignment problem, maximizes the total similarity between detected vehicles and tracked vehicles. If a detected vehicle is assigned to an existing track, we update the tracking with the new position and other information related to this vehicle. If a detected vehicle is not assigned to any existing track, we create a new tracking for that vehicle. To handle occlusions or vehicles leaving the scene, we remove any tracks that have not been assigned a detected vehicle for a certain number of frames [2].

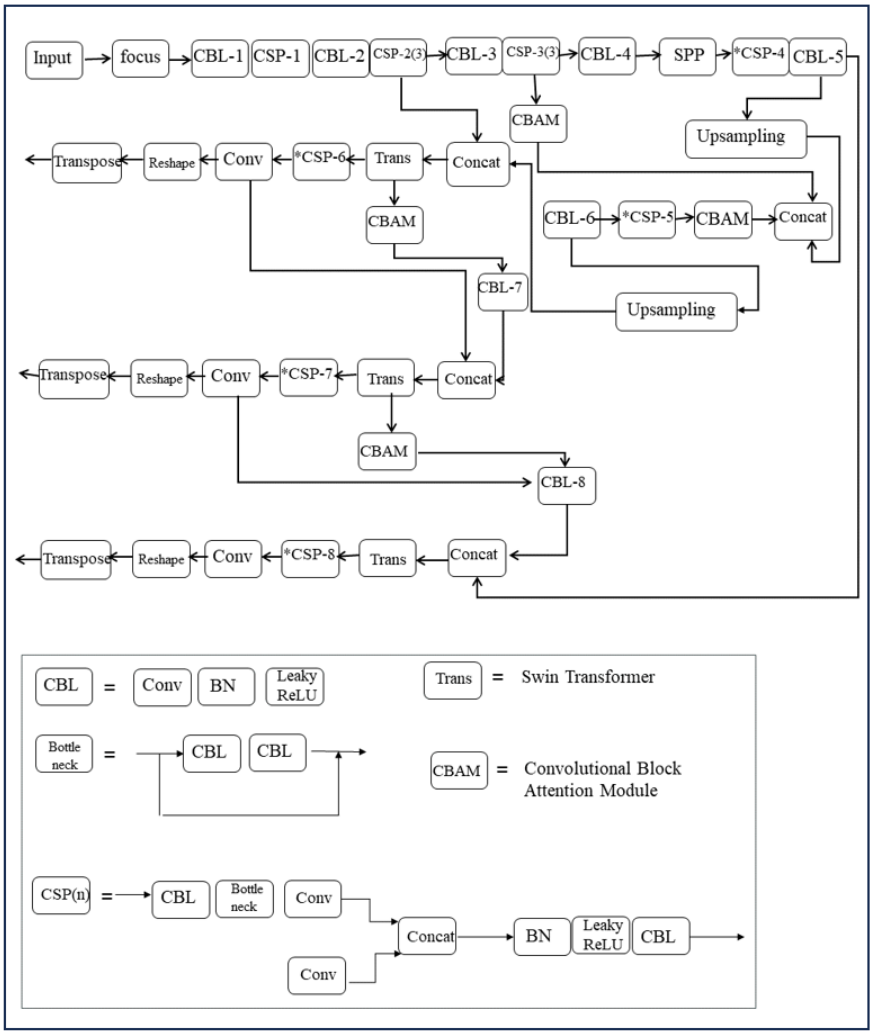

Fig. 2 The improved YOLO model in the proposed method

In order to enhance the performance of vehicle detection in traffic scenes, the target detection component of this model combines Transformer, Convolutional Block Attention Module (CBAM) and YOLO model that shown in Fig. 2. YOLO model inherits its structure from the four-part networks, consisting of input, backbone, neck, and prediction stages [4, 30].

However, YOLO model introduces further improvements, including data augmentation at the input side, as well as adaptive anchor frames and adaptive image

scaling functions. These enhancements contribute to better accuracy in anchor location and faster inferencing speed [5, 6].

YOLO models also employ Darknet as the backbone to extract features from input images [30]. Additionally, YOLO models benefit from the Cross Stage Partial Network (CSPNet), which addresses the gradient problem in network optimization for other large-scale CNN frameworks. CSPNet integrates gradient changes into the feature map from start to finish, which leads to a reduction in model parameters and floating-point operations (FLOPs) value. This approach allows for a reduction in model size while maintaining both inference speed and accuracy.

In the proposed model, through learning the four offsets of t_x, t_y, t_w and t_h, the bounding box coordinates obtained by regression are b_x, b_y, b_w, b_h, that is, the positioning and size of the bounding boxes are interconnected with the feature map. Among them, t_x and t_y represent the predicted coordinate offset values, while t_w and t_h denote the scaling factors.:

$$\mathrm{b_x} = 2\sigma(\mathrm{t_x}) - 0.5 + \mathrm{c_x} \tag{1}$$

$$\mathrm{b_y} = 2\sigma(\mathrm{t_y}) - 0.5 + \mathrm{c_y} \tag{2}$$

$$\mathrm{b_w} = \mathrm{p_w}(2\sigma(\mathrm{t_w}))^2 \tag{3}$$

$$\mathrm{b_h} = \mathrm{p_h}(2\sigma(\mathrm{t_h}))^2 \tag{4}$$

where c_x and c_y correspond to the coordinates of the upper left corner of the grid cell in the feature map, while p_w and p_h represent the width and height of the predefined anchor box mapped to the feature map.

The major difference in the loss function lies in the computation of the positive sample anchor area. The classification and confidence branches utilize Binary Cross Entropy (BCE) loss, while the bbox (Bounding box) branch employs the GIoU loss,

$$\mathrm{BCE}(\hat{\mathrm{c}}_\mathrm{i}, \mathrm{c}_\mathrm{i}) = -\hat{\mathrm{c}}_\mathrm{i} \times \log(\mathrm{c}_\mathrm{i}) - (1 - \hat{\mathrm{c}}_\mathrm{i}) \times \log(1 - \mathrm{c}_\mathrm{i}) \tag{5}$$

$$\mathrm{GIoU} = \frac{|\mathrm{A} \cap \mathrm{B}|}{|\mathrm{A} \cup \mathrm{B}|} - \frac{|\mathrm{A_c} - \mathrm{U}|}{|\mathrm{A_c}|} \tag{6}$$

where A_c represents the minimum overlap area of the two boxes. To consider the aspect ratio of the bounding boxes in the loss function, the CIoU loss is utilized as the boundary regression loss function:

$$\mathrm{L_{CIoU}} = 1 - \mathrm{IoU} + \frac{\rho^2(\mathrm{b}, \mathrm{b^{gt}})}{\mathrm{c}^2} + \alpha \mathrm{v} \tag{7}$$

$$\mathrm{v} = \frac{4}{\pi^2}(\arctan\frac{\mathrm{w^{gt}}}{\mathrm{h^{gt}}} - \arctan\frac{\mathrm{w}}{\mathrm{h}})^2 \tag{8}$$

where b and b^{gt} represent the center points of the predicted bounding box and the ground truth bounding box, respectively; ρ denotes the Euclidean distance between the two center points, while c represents the diagonal distance of the smallest enclosing area that contains both the predicted box and the ground truth box. Additionally, α is the weight applied in the calculation.

The matching strategy in YOLO models ensures that each ground truth bounding box is assigned a unique anchor. The rule dictates that, while guaranteeing the maximum Intersection over Union (IOU), a ground truth box cannot be matched to predictions across all three prediction layers simultaneously. However, this matching strategy does not take into account for cases where one ground truth bounding box corresponds to multiple anchors, nor does it consider the appropriateness of anchor settings. Consequently, if a ground truth bounding box is associated with multiple anchors, it may slow down the overall model convergence. In this paper, the approach of augmenting the number of positive sample anchors is employed to expedite convergence. This is the key reason why YOLO models achieved rapid convergence in practical applications.

In contrast to the max IOU matching rule in previous versions, YOLO models abandon this approach for any output layers. Instead, it directly utilizes the shape rule for matching, wherein the bounding box and the anchor of the current layer are employed to calculate the aspect ratio.If the aspect ratio exceeds the predefined threshold, the object feature is revealed. Pertaining to the remaining bounding boxes, YOLO models identify the nearest two grids that encompass the box based on which grid it falls into. By applying the rounding rule, these three grids are collectively deemed responsible for predicting the box. By employing this approach, the number of positive samples is roughly estimated to have increased by at least three times compared to the previous YOLO series.

To further enhance the performance of the model, we have incorporated the attention mechanism module and Transformer with YOLO model. The attention mechanism simulates the internal process of biological observation behavior, where it aligns internal experience with external senses, thereby enhancing the precision of observation in specific areas. For instance, during the processing of an image, human vision promptly scans the global image to identify a target area for concentrated focus, referred to as the focus of attention. Subsequently, a greater allocation of attentional resources is directed towards this region to acquire more comprehensive information regarding the attended target and, simultaneously, to suppress irrelevant information from other regions.

In summary, the attention mechanism assigns distinct weighting parameters to individual elements of the input, thereby intensifying focus on elements that bear similarity to the input and concurrently suppressing superfluous information. Its principal advantage lies in its capacity to simultaneously account for global and local connections in a single step, facilitating parallel computation, a crucial attribute especially pertinent to big data scenarios.

In this paper, our primary research objective is to enhance the performance of the original YOLO network by incorporating the Convolutional Block Attention Module (CBAM) in conjunction with Swin Transformer [20, 36]. The integration of

CBAM facilitates the network in determining what and where to focus when analyzing intricate traffic environments by leveraging both spatial and channel feature connections. The neck of this network is responsible for reprocessing crucial environmental features extracted from the backbone, transmitting them to the head, and subsequently producing prediction outcomes. To achieve this, we strategically insert the CBAM module after each Concatenation operation and refine the information of the channel and spatial feature fusion layer. This refinement assists the model in allocating greater attention to key information within the complex traffic environment. The experiment aims to ascertain whether the inclusion of CBAM can lead to improved performance compared to the original YOLO model.

The architecture of CBAM attention mechanism module consists of two main components: Spatial attention and channel attention. Upon receiving the feature map as input, it undergoes the channel attention process. Global Average Pooling and Global Max Pooling operations are performed based on the width and height of the feature map. Subsequently, the channel attention weight is obtained through Multilayer Perceptron (MLP), and further is normalized using the Sigmoid function. Finally, the original input feature map is recalibrated channel by channel through element-wise multiplication, through completing the channel attention-based feature recalibration

In pursuit of attention features in spatial dimension, the feature map derived from channel attention undergoes both global maximum pooling and global average pooling operations, which results in a transformation of the feature dimension from H×W to 1×1. Afterward, the dimension of feature map is reduced via convolution with a 7×7 kernel, followed by the application of the ReLU activation function. Subsequently, the feature map is restored to its original dimension through another convolutional operation in the completion of the feature map's recalibration process.

Within the spatial attention module, spatial attention features are obtained using global average pooling and maximum pooling techniques. The establishment of spatial feature correlations is achieved through two convolutional operations, which ensures that the input and output dimensions remain unchanged. The utilization of a 7×7 convolutional kernel significantly reduces the parameters and computational complexity through facilitating the establishment of high-dimensional spatial feature correlations. Followed the application of CBAM, the new feature map acquires attention weights in both the channel and spatial dimensions. This improvement significantly enhances the interconnection between each feature in the channel and space, thereby promots the extraction of effective target features.

Regarding comparison with CBAM, we explore the Squeeze-and-Excitation (SE) attention mechanism [10]. The SE attention mechanism was introduced to address the issue arising from the varying significance of different channels within a feature

map during the convolutional pooling process. In convolutional pooling, each channel of a feature map is inherently considered equally important. However, in practical scenarios, the significance of different channels varies.

The SE attention enhancement model focuses on the object in two steps: Squeeze and excitation. The squeeze is based on global average pooling of channel information for the input feature map, conditional on input x, the squeeze step for the c-th channel can be expressed as,

$$\mathrm{Z_c} = \frac{1}{\mathrm{H*W}} \sum_{\mathrm{i=1}}^{\mathrm{H}} \sum_{\mathrm{j=1}}^{\mathrm{W}} \mathrm{x_c(i,j)} \tag{9}$$

where Z_c represents output of the c-th channel. The input x originates from a convolutional layer with a predetermined kernel size, and the squeeze operation enables the model to gather global information.

The information after squeeze is multiplied onto the input feature map by two fully connection layers, the activation function and then normalized. The aim of this excitation is to completely capture the dependencies between channels:

$$\mathrm{Y} = \mathrm{X} * \sigma(\bar{\mathrm{Z}}) \tag{10}$$

$$\bar{\mathrm{Z}} = \mathrm{T_2(ReLU)(T_1(Z))} \tag{11}$$

where T_1 and T_2 are two linear transforms that capture the importance of each channel through learning.

SE is incorporated into YOLO model by using two approaches: Attention is added to the final layer of the backbone; All occurrences of C3 in the backbone are replaced. The Coordinate Attention (CA) is the second attention mechanism. CA disassembles channel attention into two one-dimensional feature encoding processes, each designed to gather features along two distinct spatial orientations. As a result, this configuration enables the capturing of long-range dependencies along one spatial direction while preserving precise positioning information along the other spatial direction. The generated feature maps are encoded as a pair of direction-aware and location-sensitive attention maps, which can be combined with the input feature maps to enhance the representation of the target object [9].

The incorporation of CA attention mechanism involves two steps: The embedding of the coordinate message and the generation of the coordinate attention. Given an input X, each channel undergoes encoding along the horizontal and vertical coordinates through pooling kernels of size (H, 1) or (1, W), respectively. Consequently, the output of channel c with height l is represented as,

$$\mathrm{Z_c^h}(h) = \frac{1}{W} \sum_{0 \le \mathrm{i} \le \mathrm{W}} \mathrm{x_c}(h, i) \tag{12}$$

The output of channel c with width W is expressed as

$$\mathrm{Z_c^w}(w) = \frac{1}{H} \sum_{0 \le \mathrm{i} \le \mathrm{H}} \mathrm{x_c}(w, i). \tag{13}$$

After undergoing the information embedding transformation, this section performs a concatenation operation on the aforementioned transformations.

Subsequently, the concatenated output is further processed using 1×1 convolutional transformation function,

$$f = \delta\left(\mathrm{F}_1([\mathrm{Z}^{\mathrm{h}}, \mathrm{Z}^{\mathrm{w}}])\right) \tag{14}$$

where $[\cdot]$ denotes the concatenate operation along the spatial dimension, $\delta(\cdot)$ represents the non-linear activation function, and f denotes the intermediate feature mapping that encodes spatial information in both horizontal and vertical directions. Furthermore, other two 1×1 convolutional transformation functions F_1, f^h and f^w are employed to transform into tensors with the same number of channels as the input X, respectively:

$$\mathrm{g}^{\mathrm{h}} = \sigma(\mathrm{F}_{\mathrm{h}}(\mathrm{f}^{\mathrm{h}})) \tag{15}$$

$$\mathrm{g}^{\mathrm{w}} = \sigma(\mathrm{F}_{\mathrm{w}}(\mathrm{f}^{\mathrm{w}})) \tag{16}$$

The output Y is,

$$\mathrm{y}_{\mathrm{c}}(\mathrm{i},\mathrm{j}) = \mathrm{x}_{\mathrm{c}}(\mathrm{i},\mathrm{j}) * \mathrm{g}_{\mathrm{c}}^{\mathrm{h}}(\mathrm{i}) * \mathrm{g}_{\mathrm{c}}^{\mathrm{w}}(\mathrm{j}) \tag{17}$$

Due to the ability of Transformer to capture global information, it exhibits superior performance in comprehending dense and occluded objects within intricate traffic environments. Consequently, we integrate Swin Transformer encoder into the head of YOLO model, effectively combine the two networks.

The primary breakthrough in the Swin Transformer lies in its adoption of localization and shifted windows. By employing non-overlapping windows for self-attention computation, localized self-attention is computed within each scale feature map's window. However, the computation between different scales leads to a deficiency in information interaction between windows. To overcome this limitation, the Swin Transformer incorporates shifted windows at varying levels, encompasses both W_MSA (Window Multi-head Self-Attention) and SW_MSA (Shifted Window Multi-head Self-Attention) [20].

We assume that the feature map passed into the Swin Transformer Block is Z^{l-1} which passes through LayerNorm and MSA and then adds with Z^{l-1} and adds to get $\hat{Z}^l$. After passing through a LayerNorm and MLP, $\hat{Z}^l$ is directly connected to $\hat{Z}^l$ and added to obtain Z^l:

$$\hat{\mathrm{Z}}^{\mathrm{l}} = \mathrm{W} - \mathrm{MSA}(\mathrm{LN}(\mathrm{Z}^{\mathrm{l}-1})) + \mathrm{Z}^{\mathrm{l}-1}, \tag{18}$$

$$\mathrm{Z}^{\mathrm{l}} = \mathrm{MLP}(\mathrm{LN}(\hat{\mathrm{Z}}^{\mathrm{l}})) + \hat{\mathrm{Z}}^{\mathrm{l}}, \tag{19}$$

$$\hat{\mathrm{Z}}^{\mathrm{l}+1} = \mathrm{SW} - \mathrm{MSA}(\mathrm{LN}(\mathrm{Z}^{\mathrm{l}})) + \mathrm{Z}^{\mathrm{l}}, \tag{20}$$

$$\mathrm{Z}^{\mathrm{l}+1} = \mathrm{MLP}(\mathrm{LN}(\hat{\mathrm{Z}}^{\mathrm{l}+1})) + \hat{\mathrm{Z}}^{\mathrm{l}+1}, \tag{21}$$

The regression modified SiamRPN, integrated with YOLO+CBAM+Transformer, assumes the role of target tracking. Within this model, the region proposal network (RPN) comprises two branches, each is responsible for foreground and background classification, as well as proposal regression, respectively.

Siamese-RPN employs a fully connected CNN without padding. The network consists of two branches: The template branch takes the target patch from the historical frame as input; the detection branch uses the target patch from the current frame as input [39]. To ensure compatibility with subsequent tasks, both branches share the same parameters in the CNN. We denote the feature maps of these two branches as $\varphi(x)$ and $\varphi(z)$, respectively [21].

Similar to the RPN network in Faster R-CNN, if there are k ($k > 0$) anchors, the network is required to produce a channel map with dimensions $2k$ for object classification and a $4k$ channel map for anchor regression. Consequently, $\varphi(z)$ is initially split into two branches, $[\varphi(z)]_{\mathrm{cls}}$ and $[\varphi(z)]_{\mathrm{reg}}$, through two separate convolution operations, corresponding to $2k$ and $4k$ channels, respectively. Similarly, $\varphi(x)$ is also divided into two branches, $[\varphi(x)]_{\mathrm{cls}}$ and $[\varphi(x)]_{\mathrm{reg}}$. The channels in $\varphi(x)$ remain unchanged, while a specialized convolution operation is applied using $[\varphi(z)]_{\mathrm{cls}}$ and $[\varphi(z)]_{\mathrm{reg}}$ as the convolution kernels. Convolution operations are performed on the feature maps $[\varphi(x)]_{\mathrm{cls}}$ and $[\varphi(x)]_{\mathrm{reg}}$, respectively. Finally, the outputs after the convolution consist of $17\times17\times2k$ and $17\times17\times4k$ channels.

$$A^{\mathrm{cls}}_{\mathrm{w\times h\times 2k}} = [\varphi(\mathrm{x})]_{\mathrm{cls}} \star [\varphi(\mathrm{z})]_{\mathrm{cls}} \tag{22}$$

$$A^{\mathrm{reg}}_{\mathrm{w\times h\times 4k}} = [\varphi(\mathrm{x})]_{\mathrm{reg}} \star [\varphi(\mathrm{z})]_{\mathrm{reg}} \tag{23}$$

where '$\star$' represents the convolution operation. The final classification branch outputs a feature map with $2k$ channels, k is the number of anchors. This feature map will be grouped into pairs and split into k groups, each group has a score map with two channels that represents the scores of the foreground and the background, respectively. In a similar way, for the regression branch, the final output of $4k$ channel feature maps, each has a group with 4 channels that represents the center position and size of the anchor, which is also divided into k groups.

Visual object tracking is a one-shot detection, where z represents the template part and x denotes the detection part, the Siamese feature extraction subnet is composed by the function $\varphi(\cdot)$, and the RPN subnet is denoted by function $\zeta(\cdot)$. The one-shot detection can be expressed as,

$$\min_{\mathrm{W}} \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathcal{L}(\zeta(\varphi(\mathrm{x_i}; \mathrm{W}); \varphi(\mathrm{z_i}; \mathrm{W})), \ell_{\mathrm{i}}) \tag{24}$$

Regarding proposal selection, SiamRPN utilizes the window and scale change penalty to reorganize the proposal scores, ultimately obtaining the optimal proposal. While following the elimination of outliers, a window is applied to mitigate large displacements, while a penalty is incorporated to suppress substantial changes in size and ratio.

$$\mathrm{Penalty} = \mathrm{e}^{\mathrm{k}*\max\left(\frac{\mathrm{r}}{\mathrm{r}'}, \frac{\mathrm{r}'}{\mathrm{r}}\right)*\max\left(\frac{\mathrm{s}}{\mathrm{s}'}, \frac{\mathrm{s}'}{\mathrm{s}}\right)} \tag{25}$$

where k denotes a hyperparameter, r represents the aspect ratio of the proposal, r' shows the height to width ratio of the last frame, s and s' respectively signify the overall scale of the proposal and the last frame [15, 29].

Data association is a critical process in multitarget tracking, primarily focusing on matching multiple targets between frames. In this paper, we adopt the classic Hungarian matching algorithm to accomplish the task of multi-target vehicle tracking.

Vehicle proposals are gathered from the current frame and the preceding frame. Subsequently, a cost matrix is computed based on the similarity measures between each vehicle proposal and the existing tracks. This cost matrix is designed such that each element (i, j) signifies the cost linked with assigning the ith vehicle proposal to the j-th track. The calculation of this cost takes into account metrics such as distance, appearance similarity, motion prediction, or a composite of these factors. The primary goal of Hungarian algorithm is to minimize the overall assignment cost while simultaneously ensuring that each proposal is allocated to a single track, and conversely, each track is associated with just one proposal. This objective is accomplished by identifying the optimal assignment that results in the lowest cumulative cost. The algorithm takes the cost matrix as input and computes the assignment pattern that generates the minimum cost. Once the Hungarian algorithm produces the optimal assignments, these assignments can be matched with the respective tracks and proposals. Each assignment pair (i, j) signifies that the i-th vehicle proposal is linked to the j-th track.

Hereinafter, we provide a step-by-step explanation of Hungarian matching algorithm:

Step 1. Determine the smallest element in each row of the matrix and subtract the corresponding smallest element from each element in that row.

Step 2. Check if the objective of this algorithm has been achieved. If not, proceed to the next step; otherwise, terminate.

Step 3. Determine the smallest element in each column of the modified matrix and subtract the corresponding smallest element from each element in that column.

Step 4. Cover all the 0s using the fewest possible vertical and horizontal lines.

Step 5. Check if the number of covered rows equals the order of the matrix. If so, the optimal matching solution is obtained. Otherwise, proceed to step 6.

Step 6. Find the minimum value in the uncovered portion. Subtract this value from each element in the uncovered rows and add it to each element in the covered columns. Return to step 4.

This algorithm is applied to the multitarget tracking problem and find the optimal matching solution for multiple targets in two frames [13].

4. Results

We collected a substantial amount of visual data from traffic scenes using a driving recorder and created a dataset focusing on vehicles as the target objects. From this dataset, we selected 3,000 high-quality images for processing and labeling, dividing them into a training set and a test set with a 3:1 ratio. Additionally, we applied data augmentation such as rotation, flipping, and translation to enhance the diversity of our dataset.

Through the experimental results of vehicle detection, we observe that the model accurately detects both larger vehicles nearby and smaller ones in the distance. Moreover, there is no noticeable shift in the position of the bounding box, even for smaller vehicles in the distance. The effectiveness of this vehicle detection task provides crucial support for subsequent vehicle tracking, as illustrated in Fig. 3. The scales of tracking box in the vehicle tracking task are highly adaptive according to the varying distances between vehicles. Our model assigns the same ID to the same target, allowing for effective detection and tracking of vehicles even in the presence of occlusions.

The proposed model employs three loss functions to evaluate its performance in multiple aspects: Bounding box attribute, object confidence, and class probability score, as depicted in Fig. 4. The bounding box loss assesses how accurately the model predicts the position of the bounding box through regression. The object confidence loss evaluates the model confidence in detecting visual objects and how accurate its predictions are for the box containing an object. The classification loss measures the model ability to distinguish between objects and backgrounds. In addition, we are use of precision and recall as metrics to assess the quality of our results. Precision indicates the proportion of true positive predictions among all positive predictions made by the model, while recall measures the proportion of true positive predictions among all actual positive instances in the dataset.

During the training process, the curve trends of bounding box loss, objective loss, and classification loss, as depicted in Fig. 4, follow a pattern. In the initial 50 epochs, all three loss curves experience a rapid decline, indicating the rapid learning and adjustment of this model. Subsequently, from 50 epochs to 150 epochs, the three loss curves gradually stabilize, suggesting that the model starts to converge and fit the data.

Simultaneously, the precision and recall curves show a tendency to plateau at around the 75th epoch and 50th epoch, respectively. To comprehensively evaluate the performance, we trained the model using various Intersection over Union (IOU) thresholds, ranging from 0.50 to 0.95. The final average precision for the detection of 2214 vehicles is 0.995, as shown in Fig. 5, which indicates the accuracy and robustness of the network in detecting vehicles.

During the evaluation of vehicle tracking performance, we vary the thresholds of location error during the training process to compute precision values for each threshold and assess the performance of models based on the area under the curve. Additionally, we calculate the ratio of frames successfully tracked in the sequence to the total number of frames at different overlap rate thresholds. In both Fig. 6 and Fig. 7, the plotted points on the curves are positioned above the diagonal lines, indicating that the performance is satisfactory. These results signify that the model is capable of accurately tracking vehicles, with the achieved performance surpassing the expected baseline.

Fig. 3 Experimental results of object tracking

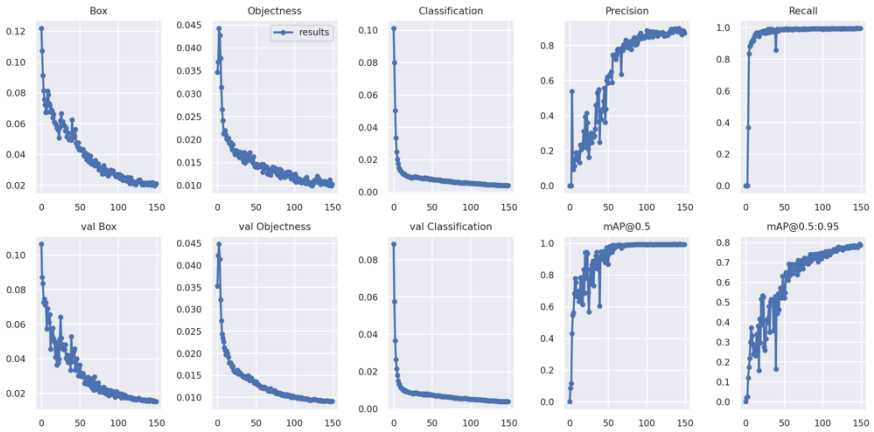

Fig. 4 Evaluations of vehicle detection with multiple methods

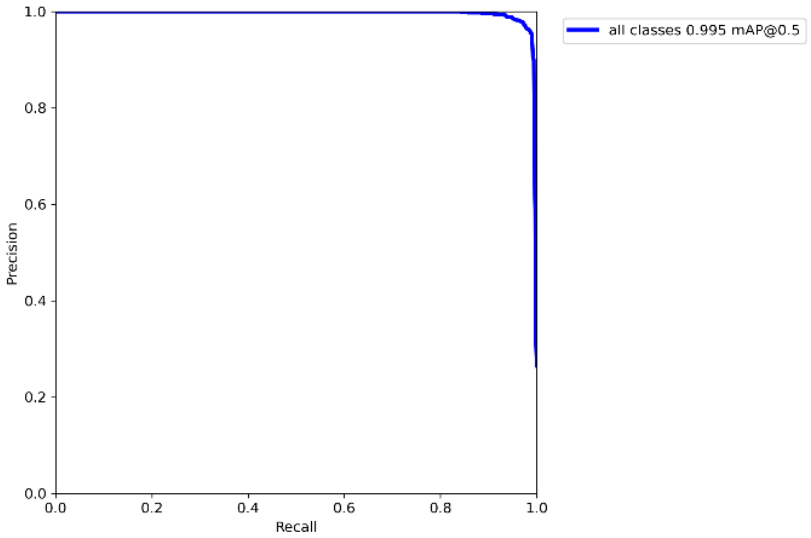

Fig. 5 The mean average precision curve for a detection task

Through comparing the state-of-the-art models, our results are shown in Table 1. In the case of a batch size 16, our proposed model is excellent in FPS and mAP. We can also know from the comparison between the combination of YOLO andvarious attention modules, the results of adding Transformer to them can effectively improve the detection average precision and mean average precision.

In order to ensure the accuracy and robustness of this model, we also compare the performance of our proposed model in Table. 2. Regarding MOTP, the performance is the best among the other three models, but it is 0.30% lower than the proposed

model. Moreover, from the results, we see that under the premise of combination of our proposed model which can receive the best detection result than other models, the effect of using modified SiamRPN is better than that of DaSiamRPN in MOTA (5.2% higher) while they are not much different on MOTP.

The proposed model takes use of a large number of training data samples from our traffic scene and provides convenience for future research on traffic signs and road conditions. Secondly, this model combines the two our modifications of advanced deep learning models for the first time to realize the understanding of the traffic scenes. The experimental results show that the model is satisfactory in detecting and tracking vehicles with various sizes in the distance within complex vehicle-related scene, the bounding box has strong adaptability to vehicles of different sizes in dynamic traffic scenes. This model applies Hungarian algorithm to achieve multi-object tracking, the model is able to efficiently detect and track multiple vehicles in a complex traffic environment.

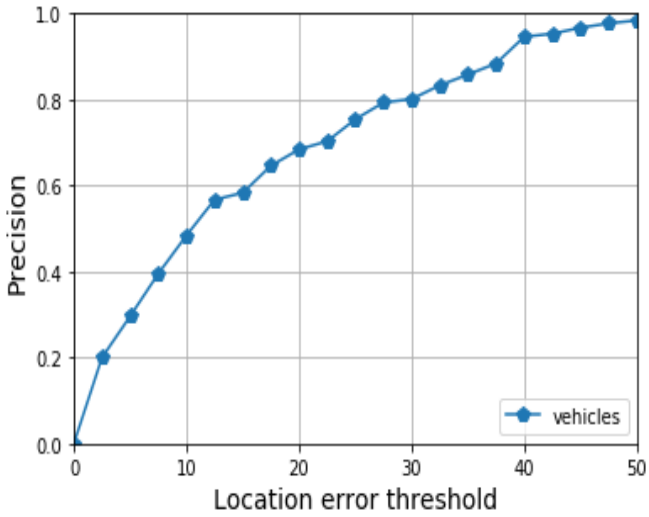

Fig. 6 The curve related to the thresholds of location errors and precisions

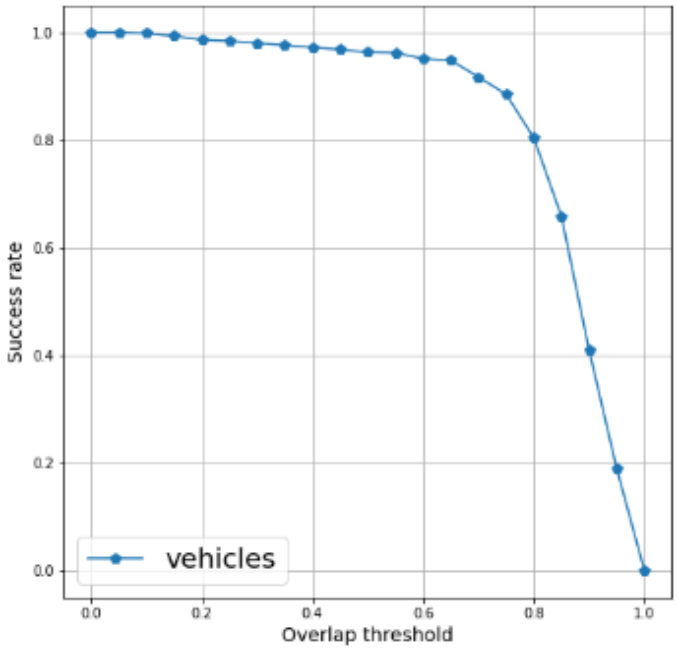

Fig. 7 The curve reflected the relationship between the overlapping thresholds and success rates

Table 1 Comparisons of different detection methods

Methods	mAPs_0.5	mAPs_0.5:0.95	FPS	Batch sizes
SSD	98.10	85.60	39	16
YOLOv4	97.60	77.90	35	16
YOLOv5	98.40	87.30	37	16
YOLOv5-CA	97.80	82.10	36	16
YOLOv5-CA-Tramsformer	97.80	82.30	35	16
YOLOv5-SE	95.90	72.20	37	16
YOLOv5-SE-Transformer	96.50	77.80	37	16
YOLOv5-CBAM	98.90	88.10	36	16
YOLOv5-CBAM-Transformer	99.50	88.70	37	16

Table 2 Comparisons of various tracking methods

Models	MOTA (%)	MOTP (%)	MT	ML	FP	FN	FM
YOLOv5-CBAM-Transformer + DaSiamRPN	33.70	74.10	21.2	39.7	307	3998	84
YOLOv5-CBAM +DaSiamRPN	37.80	75.90	19.4	36.2	311	14427	171
The proposed YOLOv5-CBAM+Modified-SiamRPN	37.30	76.40	12.6	45.8	591	12769	198
The proposed YOLOv5-CBAM-Transformer+Modified-SiamRPN	38.90	76.70	15.5	32.7	276	15962	247

In summary, though the detection speed of our proposed model is not the fastest one in FPS, the detection precisions under the same conditions is higher than that of other compared models. We found that our model can effectively improve the accuracy by 1.1%. Moreover, compared to other Siamese Networks (DaSiamRPN), our modification of SiamRPN performs more prominently in vehicle tracking tasks.

5. Conclusion and future directions

The proposed method for multiple target detection, tracing, and scene understanding improves the understanding ability of high level semantic as well as enhencing the extraction ability both of globle feature and local feature. Different from other single target tracking using SiamRPN, we achieve successful multitarget tracking. From the traffic scene, we created and labelled a new benchmark dataset that is available for a public use.

In this project, we are use of a frame-based approach, where video information is processed frame by frame. We plan to take account of spiking neural networks and dynamic vision sensors (DVS) as further research work to achieve asynchronous detection of minute changes in the scene and ensure incremental, online learning and a better interpretation of the scene [11, 12, 14, 19, 33, 48].

Declarations

This work has not any funding support, it has not any conflicts of interests or competing interests.

Data Availability Statement

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

References

1. Alexey, B., ChienYao, W., & Mark, L.: YOLOv4: Optimal speed and accuracy of object detection. Image and Video Processing, arXiv:2004.10934. (2020)
2. An, N., Yan, W.: Multitarget tracking using Siamese neural networks. ACM Transactions on Multimedia Computing, Communications and Applications, 17, pp.1-16. (2021).
3. Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., & Torr, P. H.: Fully convolutional Siamese networks for object tracking. IEEE ICCV, pp.850-865. (2016).
4. Bochkovskiy, A., Wang, C. Y, & Liao, H. Y. M.: YOLOv4: Optimal speed and accuracy of object detection. arXiv 2004.10934. (2020).
5. Chienyao, W., Alexey, B., Mark, L.: YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. IEEE CVPR (2022).
6. Chuyi, L. et, al.: YOLOv6: A single-stage object detection framework for industrial applications. IEEE CVPR (2022).
7. Elhani, D., Megherbi, A. C., Zitouni, A., Dornaika, F., Sbaa, S., & Taleb-Ahmed, A.: Optimizing convolutional neural networks architecture using a modified particle swarm optimization for image classification. Expert Systems with Applications, 229, pp.120411.(2023).
8. Gai, Y., He, W., & Zhou, Z.: Pedestrian target tracking based on DeepSORT with YOLOv5. Computer Engineering and Intelligent Control, pp.1-5. (2021).
9. Hou, Q., Zhou, D., & Feng, J.: Coordinate attention for efficient mobile network design. IEEE CVPR, pp.13713-13722. (2021).
10. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. IEEE CVPR, pp.7132-7141. (2018).
11. Kasabov, N.: Time-space, Spiking Neural Networks and Brain-inspired Artificial Intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg (2018).
12. Kasabov, N., et al.: Evolving spatio-temporal data machines based on the NeuCube neuromorphic framework: Design methodology and selected applications. Neural Networks, 1-14 (2016).
13. Kuhn, H.: The Hungarian method for the assignment problem. Naval Research Logistic Quarterly, pp.83-97.(2012).
14. Laña I., Lobo J., Capecci E., Del Ser J., & Kasabov N.: Adaptive long-term traffic state estimation with evolving spiking neural networks. Transportation Research Part C: Emerging Technologies 101, 126-144 (2019).
15. Li, B., Yan, J., Wu, W., Zhu, Z., & Hu, X.: High performance visual tracking with siamese region proposal network. IEEE CVPR, pp. 8971-8980 (2018).
16. Liu, X., Nguyen, M., Yan, W.: Vehicle-related scene understanding using deep learn. Asian Conference on Pattern Recognition, pp. 61-73 (2019).
17. Liu, X., Yan, W.: Traffic-light sign recognition using Capsule network. Springer Multimedia Tools and Applications, pp. 15161-15171 (2021).

18. Liu, X. &Yan, W.: Depth estimation of traffic scenes from image sequence using deep learning, PSIVT (2022).
19. Liu, X., Yan, W., & Kasabov, N.: Vehicle-related scene segmentation using CapsNets. IEEE IVCNZ, pp. 1-6 (2020).
20. Liu, Z, et al.: Swin transformer: Hierarchical vision transformer using shifted windows. IEEE ICCV, pp.37-49 (2021).
21. Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., & Kim, T. K.: Multiple object tracking: A literature review. Artificial Intelligence, pp.103448 (2021).
22. Ma, C., Huang, J. B., Yang, X., & Yang, M. H.: Hierarchical convolutional features for visual tracking. IEEE ICCV, pp.3074-3082 (2020).
23. Mehtab, S., Yan, W., & Narayanan, A.: 3D vehicle detection using cheap LiDAR and camera sensors. IEEE IVCNZ, pp. 1-6 (2021).
24. Mehtab, S., Yan, W.: FlexiNet: Fast and accurate vehicle detection for autonomous vehicles-2D vehicle detection using deep neural network. ACM ICCCV, pp. 43-49 (2021).
25. Mehtab, S., Yan, W.: Flexible neural network for fast and accurate road scene perception. Springer Multimedia Tools and Applications, pp. 7169-7181 (2021).
26. Müller, J., & Dietmayer, K.: Detecting traffic lights by single shot detection.Intelligent Transportation Systems, pp.266-273 (2018).
27. Peng, J., et al.: Implementation of the structural SIMilarity (SSIM) index as a quantitative evaluation tool for dose distribution error detection. Medical Physics, 47(4), pp.1907-1919. (2020).
28. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., & Yang, M. H.: Hedged deep tracking. IEEE CVPR, pp.4303-4311 (2021).
29. Rao, Y., Cheng, Y., Xue, J., Pu, J., Wang, Q., Jin, R., & Wang, Q.: FPSiamRPN: Feature pyramid Siamese network with region proposal network for target tracking. IEEE Access, 8, 176158-176169. (2020).
30. Redmon, J., & Farhadi, A.: YOLOv3: An incremental improvement. ArXiv,abs/1804.02767. (2018).
31. Sun, S., Wang, Y., & Piao, Y.: A Real-time multi-target tracking method based on deep learning. Physics, pp.12112. (2021).
32. Song, W., Jiao, L., Liu, F., Liu, X., Li, L., Yang, S., ... & Zhang, W.: A joint siamese attention-aware network for vehicle object tracking in satellite videos. IEEE Transactions on Geoscience and Remote Sensing, 60, pp.1-17. (2022).
33. Tu, E., Kasabov, N., & Yang, J.: Mapping temporal variables into the NeuCube spiking neural network architecture for improved pattern recognition and predictive modelling. IEEE Trans. on Neural networks and learning systems, 28 (6), 1305-1317 (2017).
34. Wang, K., & Liu, M.: YOLOv3-MT: A YOLOv3 using multi-target tracking for vehicle visual detection. Applied Intelligence, 52(2), pp.2070-2091. (2022).
35. Wang, Q., Gao, J., Xing, J., Zhang, M., & Hu, W.: DCFNet: Discriminant correlation filters network for visual tracking. arXiv:1704.04057 (2017).
36. Woo, S., Park, J., Lee, J. Y., & Kweon, I. S.: CBAM: Convolutional block attention module. IEEE ICCV, pp. 3-19 (2019).

37. Wu, N., & Fang, H.: A novel traffic light recognition method for traffic monitoring systems, Asia-Pacific Conference on Intelligent Robot Systems, 141-145 (2017).
38. Xing, J., Luo, Z., Nguyen, M., & Yan, W. Q.: Traffic sign recognition from digital images by using deep learning. PSIVT, pp. 37-49. (2022).
39. Xu, Y., Zhang, J., & Brownjohn, J.: An accurate and distraction-free vision-based structural displacement measurement method integrating Siamese network based tracker and correlation-based template matching. Measurement, 179, 109506. (2021).
40. Yan, W.: Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics. Springer, pp. 1-6. (2019).
41. Yan, W.: Computational Methods for Deep Learning: Theoretic, Practice and Applications. Springer (2021).
42. Yang, B., Huang, C., & Nevatia, R.: Learning affinities and dependencies for multi-target tracking using a CRF model. IEEE Conference on Computer Vision and Pattern Recognition, 1233–1240 (2011).
43. Yang, B., & Nevatia, R.: Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. IEEE CVPR, 1918–1925 (2012).
44. Yang, M., Yu, T., & Wu, Y.: Game-theoretic multiple target tracking. IEEE CVPR, 1–8 (2007).
45. Zha, Y., Wu, M., Qiu, Z., Dong, S., Yang, F., & Zhang, P.: Distractor-aware visual tracking by online Siamese network. IEEE Access, 89777-89788 (2019).
46. Zhang, L., & Maaten, L.: Structure preserving object tracking. IEEE CVPR, 1838–1845 (2013).
47. Zhu, Y., & Yan, W. Q.: Traffic sign recognition based on deep learning. Multimedia Tools and Applications, 81(13), pp.17779-17791. (2022).
48. Zuo, J., Jia, Z., Yang, J., & Kasabov, N.: Moving object detection in video sequence images based on an improved visual background extraction algorithm. Multimedia Tools and Applications, 79(39-40), 29663–29684 (2020).