

Vehicle Detection Counting Algorithm Based on Background Subtraction Algorithm and SORT

Jun Guo*, Heyan Gao*, Zeyu Yan*, Jiahui Cao**, Zhenbo Fu**

*Software College, Northeastern University, Shenyang Liaoning Province China

**School of Computer Science and Engineering, Northeastern University, Shenyang Liaoning Province China

guojun@mail.neu.edu.cn, 1520642034@qq.com, yanzeyu2019@hotmail.com, 1095164870@qq.com, 1640240763@qq.com.

Abstract—At present, the deep neural network model is commonly used to detect the vehicle in the video, and the detection accuracy is relatively high. However, the neural network model requires high computing performance and high demand for network transmission bandwidth. In many cases, the edge computing device used is of small computing power, so the neural network model is not applicable. However, background subtraction algorithm is easy to realize because of its low requirement on hardware calculation force and fast and accurate detection speed. Using SORT algorithm to track with accurate detection results can improve the speed again and reduce the consumption of computing resources. Therefore, this paper proposes an algorithm that uses the background subtraction algorithm to detect the vehicles in the video, and then uses the SORT algorithm to track the detected vehicles. The vehicle counter will automatically count when the vehicle in the video passes the traffic flow counting line. The accuracy of traffic flow counting results in this paper is 88%, which proves the feasibility and effectiveness of vehicle detection counting method based on background subtraction algorithm and SORT.

Keywords— Background subtraction algorithm; SORT algorithm; Vehicle detection count; Edge Computing; Image processing.

I. INTRODUCTION

Roads are the vital mode of transportation for people and goods around the globe and its use has grown dramatically over the years. There is one death every four minutes due to road accidents in the developing nations^[1]. The ever-increasing need for mobility has overflowed major cities with vehicles, resulting in growing traffic congestions, accompanied by unpredicted emergencies and accidents^[2]. Therefore, monitoring the vehicles on the road and adjusting the traffic flow rationally is an important means to avoid traffic congestion and vehicle collisions.

Conventional technology for traffic measurements, such as inductive loops, sonar or microwave detectors, suffer from serious drawbacks: they are expensive to install, they demand

traffic disruption during installation or maintenance, they are not portable and they are unable to detect slow or stationary vehicles^[3]. Therefore, the commonly used technology for expressway traffic anomaly monitoring in the current intelligent transportation system is video detection technology, but it still has the following shortcomings:

A. Large amount of image processing calculation.

Most video files contain redundant information such as background scenery, which costs a huge amount of storage and computing resources^[4]. Large video data stream, need to storage long time, need to continuous uninterrupted operation for 24 hours. Thus, a huge amount of information is received by traffic video detection, which requires considerable storage capacity, requires enough storage space^[5].

B. High pressure of network transmission bandwidth.

Traditional wireless cellular networks cannot meet the explosive growth of the network demand, especially in terms of high data rates and computing power^[6]. The information acquisition equipment transmits the collected traffic information directly to the server without any processing. Due to the continuity of information acquisition, the information transmission is continuous, which will increase the pressure of network information transmission.

Therefore, expressway still needs fast and efficient image processing algorithm to realize low-cost dynamic vehicle detection. As one of the important topics in computer vision, moving vehicle segmentation has attracted considerable attention of researchers^[7]. The goal of moving vehicle detection is to separate moving vehicles from background, and its detection result has a great impact on post image processing^[8].

In this paper, edge computing device is used. Firstly, vehicles are detected based on background subtraction algorithm. Gaussian model is used to calculate the pixel points collected by the equipment, and parameters are constantly updated according to the formula, so that the new pixel value matches

the Gaussian distribution model, and the error is within a certain range. If the pixel value is detected to match a previous Gaussian distribution, then this pixel is the background point of the expressway traffic image; otherwise, this pixel is the foreground point of the expressway traffic image, and the contour is calculated according to the foreground point and used as the input of SORT algorithm^[9-10]. SORT algorithm is used to track vehicles on highways in real time. Kalman filtering^[11-12] is used to predict and update the target state in vehicle tracking. Hungarian assignment algorithm^[13-14] matches the prediction result of Kalman filter with the prediction result of vehicle detection.

II. VEHICLE DETECTION BASED ON MIXED GAUSSIAN MODEL BACKGROUND SUBTRACTION ALGORITHM

Background subtraction algorithm^[15-16], known as foreground detection, is a common technique in image processing field. Background subtraction algorithm is different due to different background modeling methods and updating process, while mixed Gaussian model can accurately model and update the real-time changes of scene such as light change and leaf swaying, which is a combination of quality and speed, and is an effective processing algorithm^[17]. It can basically meet the real-time requirements of the algorithm in practical application.

Assuming that the observed values of each pixel in the expressway traffic video image collected by the edge computing device are independent of each other, and each pixel in the expressway traffic image is modeled by a mixed Gaussian model containing N Gaussian distributions, the model representation of pixel j at time t is shown in formula (2.1).

$$P(x_j) = \sum_{i=1}^N \omega_{j,t}^i \cdot \eta(x_j, \mu_{j,t}^i, \Sigma_{j,t}^i) \quad (2.1)$$

N represents the number of Gaussian distributions, generally between 3 and 5. $\omega_{j,t}^i$ represents the i th weight coefficient of the gaussian distribution at time t . $\sum_{i=1}^N \omega_{j,t}^i = 1$. x_j represents the value of pixels j at time t . $\mu_{j,t}^i$ and $\Sigma_{j,t}^i$ represents the mean vector and covariance matrix of the i th gaussian distribution at time t . η represents the probability density function of the Gaussian distribution, as shown in formula (2.2).

$$\eta(x_j, \mu_{j,t}^i, \Sigma_{j,t}^i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_{j,t}^i|^{\frac{1}{2}}} \times \exp \left[-\frac{1}{2} (x_j - \mu_{j,t}^i)^T (\Sigma_{j,t}^i)^{-1} (x_j - \mu_{j,t}^i) \right] \quad (2.2)$$

d is the dimension of vector X . Since the pixel of expressway traffic image is RGB three-channel, $d = 3$. x_j is a vector and $x_j = [x_{jR}, x_{jG}, x_{jB}]$.

$$\text{Then } \mu_t^i = (\mu_{t,R}^i, \mu_{t,G}^i, \mu_{t,B}^i), \Sigma_t^i = \begin{bmatrix} \sigma_R^2 & 0 & 0 \\ 0 & \sigma_G^2 & 0 \\ 0 & 0 & \sigma_B^2 \end{bmatrix}.$$

The mixed Gaussian model is initialized with the pixel value of the highway traffic image in the first frame. The mean value of the first Gaussian distribution is assigned to the current pixel value, and the weight value is assigned to 1. The mean value and weight of the Gaussian distribution function other than the first one are initialized to zero.

N Gaussian distributions in the background model of expressway traffic image constructed are sorted according to the order of ω/σ from large to small. The higher the weight is, the smaller the standard deviation is, the higher the gaussian distribution ranking is. Then, the first B Gaussian distributions are selected from the sorting as the background model, and B satisfies the formula (2.3).

$$\arg \min \left(\sum_{i=1}^B \omega_t^i > T \right) \quad (2.3)$$

T is the proportion of the background. ω_t^i is the weight coefficient of the i th Gaussian distribution in the mixed Gaussian model at time t .

In order to make the background model of expressway traffic image robust^[18], the parameters in the mixed Gaussian model need to be updated constantly. Therefore, at time t , the new pixel value X_t of the expressway traffic image is compared with the mean value of the current N Gaussian distributions, then the Gaussian distribution model matching the new pixel value X_t of the expressway traffic image is found. That is, the mean deviation of the Gaussian distribution model is in the range of 2.5σ , which satisfies the formula (2.4):

$$|X_t - \mu_{t-1}^i| \leq 2.5\sigma_{t-1}^i \quad (2.4)$$

X_t is the pixel value at time t . μ_{t-1}^i is the mean value of the i th gaussian distribution at time $t - 1$. σ_{t-1}^i is the standard deviation of the i th gaussian distribution at time $t - 1$.

The weight parameters of the mixed Gaussian model are updated as formula (2.5), and then the weights of each Gaussian distribution are normalized.

$$\omega_t^i = (1 - \alpha)\omega_{t-1}^i + \alpha M(t) \quad (2.5)$$

$$M(t) = \begin{cases} 1, \\ 0. \end{cases} \quad (2.6)$$

When X_t is matched with the i th Gaussian distribution, $M(t) = 1$, otherwise $M(t) = 0$.

ω_t^i and ω_{t-1}^i is the weight coefficient of the i th Gaussian distribution in the mixed Gaussian model at time t and $t - 1$. α is the learning rate, which is specified by user.

If X_t does not match any of the gaussian distributions in the mixed Gaussian distribution model, the gaussian distribution with the minimum weight is deleted and a new Gaussian distribution is added. The weight of the new Gaussian

distribution initializes a smaller value with the mean value of the current pixel and the standard deviation initializes a larger value. Then the weights are updated with the learning rate and normalized.

If X_t is matched with at least one Gaussian distribution in the mixed Gaussian distribution model. If it is matched with the i th Gaussian distribution, then the model parameters of the Gaussian distribution are updated according to the formulas (2.7), (2.8) and (2.9). If X_t is not matched with the i th Gaussian distribution, then the model mean and standard deviation of the Gaussian distribution remain unchanged.

$$\rho = \alpha \cdot \eta(X_t | \mu_{t-1}^i, \sigma_{t-1}^i) \quad (2.7)$$

$$\mu_t^i = (1 - \rho)\mu_{t-1}^i + \rho \cdot X_t \quad (2.8)$$

$$\sigma_t^i = (1 - \rho)\sigma_{t-1}^i + \rho \cdot (X_t - \mu_t^i)^T (X_t - \mu_t^i) \quad (2.9)$$

ρ is the updating rate of gaussian distribution parameters. α is the learning rate. $\eta(X_t | \mu_{t-1}^i, \sigma_{t-1}^i)$ is the Gaussian distribution probability of pixels X_t at time t . μ_t^i and μ_{t-1}^i is the mean value of the i th Gaussian distribution in the mixed Gaussian model at time t and $t - 1$. σ_t^i and σ_{t-1}^i is the variance of the i th Gaussian distribution in the mixed Gaussian model at time t and $t - 1$.

Therefore, the pixel value X_t of the expressway image taken by the edge computing device at time t is matched with the first B Gaussian distribution obtained. If the pixel value X_t matches a distribution of the first B Gaussian distribution, then this pixel is the background point of the expressway traffic image; otherwise, this pixel is the foreground point of the expressway traffic image. The outline of the vehicle is detected according to the foreground point, and the vehicle is marked with a rectangular outline, which is represented as formula (2.10) and used as the input of SORT algorithm.

$$BBox = [x, y, x + w, y + h] \quad (2.10)$$

x is the left edge of the rectangle, y is the right edge of the rectangle, w is the width of the rectangle, and h is the height of the rectangle.

After the vehicle detection based on background subtraction algorithm is completed, tracking of each vehicle is required to realize the counting of vehicles. Chapter 3 will describe the multi-vehicle tracking algorithm^[19] in detail.

III. MULTI-VEHICLE TRACKING ALGORITHM BASED ON SORT

This section uses the Simple Online and Realtime Tracking (SORT) algorithm to realize the real-time tracking of vehicles on the highway. Kalman filter predicts and updates the

target state in the vehicle tracker, and the Hungarian assignment algorithm matches the $BBox$ predicted by Kalman filter with the vehicle detection $BBox$.

The state modeling representation of each target in the vehicle tracker is shown in formula (3.1).

$$state = [x, y, s, r, v_x, v_y, v_s] \quad (3.1)$$

x is the horizontal coordinate of the target center. y is the vertical coordinates of target center. s is the target boundary box area. r is the target boundary frame aspect ratio. v_x is the target velocity in a horizontal component, v_y is a target velocity component in vertical direction, v_s is the area of the target boundary box change rate.

The Kalman filter updates the state vector according to the target observation vector to achieve the optimal estimation of the target state. The basic equations are formula (3.2) state transfer equation and formula (3.3) observation equation.

$$state_t = Fstate_{t-1} + \alpha_t \quad (3.2)$$

$$Y_t = Hstate_t + \beta_t \quad (3.3)$$

$state_t$ and $state_{t-1}$ represents the target state vector at time t and $t - 1$. Y_t represents the target state observation vector. F represents the target state transition matrix. H represents the target state observation matrix. α_t represents the state noise subject to normal distribution. The covariance matrix is Q . β_t represents the observation noise subject to normal distribution, and the covariance matrix is R .

The vehicle tracker is created and initialized with the vehicle boundary detected in the first frame of the highway traffic image, and the $state$ of the target in the vehicle tracker is initialized. Then the vehicle tracker is updated with the vehicle detection box in the highway traffic image at later time t . Firstly, the time state and Kalman estimation error covariance matrix $\hat{P}_t$ of each target in the vehicle tracker are predicted according to formula (3.4) and (3.5).

$$\widehat{state}_{t|t-1} = Fstate_{t-1} + Bu_t \quad (3.4)$$

$$\widehat{P}_{t|t-1} = F \cdot P_{t-1} \cdot F + Q \quad (3.5)$$

$\widehat{state}_{t|t-1}$ is the target state prediction at time t . $state_{t-1}$ is the target state at time $t - 1$. B_t inputs gain matrix. u_t is system input vector. $\widehat{P}_{t|t-1}$ is Kalman estimation error covariance matrix.

The predicted state of the target in the vehicle tracker is then transformed into the $BBox$ of the target, and the IOU (Intersection over Union) of the vehicle $BBox$ detected in the current frame is calculated and the IOU matrix of multiple targets in the vehicle tracker can be obtained by the IOU of the

vehicle *BBox* detected in the current frame. Then algorithm 3.1 was used to get the *IOU* composite index pair with the maximum match.

Algorithm 3.1 Highway Vehicle Detection Box Matching Algorithm

Input: *IOU* matrix $\mathbf{IOU}_{m \times n}$ for multiple vehicle detection boxes in the current frame and multiple targets in the tracker

Output: The *IOU* index assigned to the Indices

Definition:

$\mathbf{IOU}_{m \times n}$: *IOU* matrix for multiple vehicle detection boxes in the current frame and multiple targets in the tracker

$\mathbf{matrix}_{m \times n}$: The operation matrix $\mathbf{IOU}_{m \times n}$ after transformation

$\min_i$: the minimum value of the row *i*

$\min$: Minimum value in an uncovered area

$\mathbf{line}_c$: Number of covered columns when the column $\mathbf{0}^*$ is overridden

$\mathbf{Z0}, \mathbf{Z1}, \mathbf{Z2}$: elements collection

$\mathbf{matrix}_{i,j}$: elements of the matrix

indices: Index pairs for output

Algorithm description:

BEGIN:

IOU matrix of multiple targets in multiple vehicle detection boxes and trackers in the current frame, $\mathbf{IOU}_{x \times y}$

if ($x > y$): $\mathbf{matrix}_{m \times n} = \mathbf{IOU}_{x \times y}^T$

else: $\mathbf{matrix}_{m \times n} = \mathbf{IOU}_{x \times y}$

end

for $i = 1$ to m

$\min_i = \min_j \mathbf{matrix}_{i,j} \quad 1 \leq j \leq n$

for $j = 1$ to n $\mathbf{matrix}_{i,j} = \mathbf{matrix}_{i,j} - \min_i$

end

end

for $i = 1$ to m

for $j = 1$ to n

if ($\mathbf{matrix}_{i,j} == \mathbf{0}^*$ &&

row *i* and cloumn *j* don't contain $\mathbf{0}^*$)

$\mathbf{matrix}_{i,j} = \mathbf{0}^*$

end

end

end

while ($\mathbf{line}_c < m$):

// overrides the column of $\mathbf{0}^*$ and the number of columns overridden is $\mathbf{line}_c$

while (true):

if (the uncovered region $\mathbf{matrix}_{p,q}$ exists in

$\mathbf{matrix}_{m \times n} == \mathbf{0}$):

$\mathbf{matrix}_{p,q} = \mathbf{0}'$

if ($\mathbf{matrix}_{p,q}$ on the row with no element $\mathbf{0}^*$):

The marked uncovered region $\mathbf{0}'$ is $\mathbf{Z0}$, the

element $\mathbf{0}^*$ in the column of $\mathbf{Z0}$ is $\mathbf{Z1}$,

and the element $\mathbf{0}'$ in the row of $\mathbf{Z1}$ is

$\mathbf{Z2}$. Until there is no $\mathbf{0}^*$ in the above

column where $\mathbf{0}'$ exists. Replace the

element $\mathbf{0}^*$ with 0, replace the element

$\mathbf{0}'$ with $\mathbf{0}^*$, clear the overwrite line

break

else: Override the row of $\mathbf{matrix}_{p,q}$, unoverride

or do not override the column of the element $\mathbf{0}^*$

end

else:

$\min$ = Minimum value in uncovered area

each row of uncovered area minus $\min$

break;

end

end

If (overwrite line cleared): Overwrite the column

$\mathbf{0}^*$. The number of overwrite columns is $\mathbf{line}_c$

end

end

for $i = 1$ to m

for $j = 1$ to n

if ($\mathbf{matrix}_{i,j} == \mathbf{0}^*$): *indices*.add(*i*, *j*)

end

end

end

END

The *IOU* composite index pair with the maximum match is filtered to remove the index pair that is less than the *IOU* threshold (typically 0.3). Therefore, targets in vehicle detection boxes and vehicle trackers can be divided into three arrays: an array of matches in vehicle detection boxes and vehicle trackers (*matches*), a group of groups that don't match the number of vehicle detection boxes (*unmatched_trackers*), and an array of targets in vehicle tracers (*unmatched_trackers*). For the matches array, update the target state in the vehicle tracker with the vehicle detection box, and calculate the Kalman gain coefficient equation according to formula (3.6) :

$$K_t = \widehat{P}_{t|t-1} H^T (H \widehat{P}_{t|t-1} H^T + R)^{-1} \quad (3.6)$$

K_t is the Kalman gain coefficient at time *t*. $\widehat{P}_{t|t-1}$ is the prediction Kalman estimate error covariance matrix at time *t*, *H* and *H^T* is the observation matrix of the target state and its transpose matrix, and *R* is the observation noise covariance matrix.

Modify the state of the target in the vehicle tracker according to formula (3.7).

$$\widehat{state}_{t|t} = \widehat{state}_{t|t-1} + K_t(Y_t - H\widehat{state}_{t|t-1}) \quad (3.7)$$

$\widehat{state}_{t|t}$ is the modified state of the target at time t . $\widehat{state}_{t|t-1}$ is the predicted state of the target at time t . K_t is the Kalman gain coefficient at time t . Y_t is the observation vector of the target state at time t , namely the vehicle detection frame information matched with the target, and H is the observation matrix of the target state.

According to formula (3.8), the Kalman estimation error covariance matrix is modified.

$$\widehat{P}_{t|t} = (I - K_t H) \widehat{P}_{t|t-1} \quad (3.8)$$

$\widehat{P}_{t|t}$ is the correction matrix of kalman estimate error covariance at time t . I is the identity matrix. K_t is the Kalman gain coefficient at time t , H is the observation matrix of the target state. $\widehat{P}_{t|t-1}$ is the prediction Kalman estimate error covariance matrix at time t .

For the unmatched_detections array, no target in the vehicle tracker matches the vehicle detection box, create and initialize a new vehicle tracker for the vehicle volume detection box, and initialize the state information with Kalman filter; for the unmatched_trackers array, no vehicle detection box matches the target in the vehicle tracker, and if no successive T frames match the vehicle detection box in the vehicle tracker, the target is considered to disappear and removed from the vehicle tracker.

After the vehicle tracker is updated, the state of each target in the vehicle tracker is converted to a formula (3.9) representation and this information is returned for use as input for other processing.

IV. BACKGROUND SUBTRACTION ALGORITHM AND SORT COMBINED VIDEO VEHICLE COUNTING ALGORITHM

The background differential detection is applied to the vehicle contour in the expressway traffic image, and the vehicle contour $detection = [x, y, x + w, y + h]$ is taken as the input of real-time vehicle tracking, and then the SORT algorithm is applied to update the information of the target in the vehicle tracker $track = [x, y, x + w, y + h, id]$. Compare vehicle trackers of adjacent frames, if the goal of the current frame tracker id exists in previous frame tracker, this goal is the same vehicles, then connect the current frame under the target center of bottom border and previous frame borders center, if this is a straight line and the counting line intersect in the video, vehicle counter plus one, according to the algorithm 4.1 expressway traffic video traffic count results are obtained.

Algorithm 4.1 Highway Traffic Video Traffic flow counting Algorithm

Input: a highway traffic video data

Output: Traffic count results in this video

Definition:

count: Statistics of traffic flow

frame_{count}: Total frames of video

frame_t: highway traffic image at the moment t

BackgroundSubtractor(): Background subtraction algorithm

dets_t: Vehicle detection frame in highway image at time t

Tracker.Update(): The update function of the tracker

tracks_t: Tracking line after updating time t

line: Highway vehicle counting line

line_{t,t-1}: The connecting line of the boundary center point under the boundary box and the corresponding target in

tracks_{t-1} and **tracks_t**

Algorithm description:

BEGIN:

// Enter a highway traffic video data

count=0

for **frame_{count}** = 1 to t

 Read video frame **frame_t**

 // Apply the background subtraction algorithm to the current frame to get the vehicle detection frame

dets_t = **BackgroundSubtractor(frame_t)**

 // Use the vehicle detection box to update the target information in the tracker

 tracks_t = tracker.update(**dets_t**)

 for track in **tracks_t**

 if (the target **id** exists in the tracker **tracks_{t-1}**):

 Connect the boundary center point under the boundary box with the corresponding target in **tracks_{t-1}** and **tracks_t**, and get the line segment **line_{t,t-1}**

 if (**line_{t,t-1}** intersects traffic count line **line**):

 count=count+1

 end

 end

end

end

END

V. EXPERIMENT AND ANALYSIS

A. Experimental Environment

In this chapter, raspberry PI 3 Model B is used as edge computing device^[20], the raspberry PI official 5-megapixel camera is used to collect highway traffic video data. The camera is connected to raspberry PI through a 15-core cable. The device is simply connected as shown in Figure 5.1.

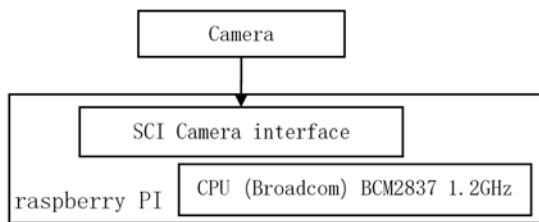


Figure 5.1 Simple Connection of Edge Computing Device Diagram

B. Experimental process and result analysis

Raspberry PI collected expressway traffic video data, and then applied background subtraction algorithm to detect the vehicles in the video. SORT algorithm was used to track the vehicles in the video and count the traffic flow in the expressway traffic video. The experimental content is divided into the following two parts:

1) Vehicle detection: Background subtraction algorithm is used to detect the vehicles in the video. After background subtraction is applied to a frame of the video, the foreground vehicle is obtained, and then the vehicle outline is marked with a rectangular box. The results are shown in Fig 3.2.

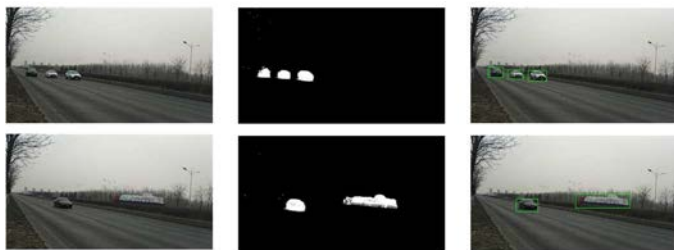


Fig.5.2 Video Image Diagram, Background subtraction effect Diagram and Vehicle contour marking Diagram

2) Vehicle detection: The vehicle detected by background subtraction algorithm is tracked. When the vehicle passes the traffic flow counting line, the counter is added one. Due to the small displacement of vehicles between adjacent frames, in order to reduce the processing time and ensure the correctness of the results, vehicle detection and tracking are carried out every other frame in the processing of video. The longer the traffic flow data takes, the longer the time it takes no vehicle to pass, so video data is collected every 5 seconds.

In general, the vehicle motion can be as a uniform motion in a straight line between adjacent frames, Kalman filter has seven state variables and 4 observation input, the Kalman filter state transition matrix F , observation matrix H , measurement noise covariance matrix R , target state covariance matrix P , initialize system noise covariance matrix Q :

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, R = \begin{bmatrix} 1. & 0 & 0 & 0 \\ 0 & 1. & 0 & 0 \\ 0 & 0 & 10. & 0 \\ 0 & 0 & 0 & 10. \end{bmatrix}$$

$$P = \begin{bmatrix} 10. & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10. & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10. & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10. & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10000. & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10000. & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10000. \end{bmatrix}, Q = \begin{bmatrix} 1. & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1. & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1. & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1. & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01. & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.001 \end{bmatrix}$$

In this experiment, more than 300 sections of expressway traffic video data were processed and analysed, and the traffic flow count results were mainly 0 cars, 1 car, 2 cars and 3 cars. Visualize the counting process of a video, as shown in Fig 5.3. According to statistics, the accuracy rate of traffic flow count results in this paper is 88%.



Fig.5.3 Video-based visualization of Highway Traffic Flow Count Diagram

VI. CONCLUSION

This chapter mainly proposes how to use the equipment with low computing power to carry out vehicle detection. First of all, based on the background subtraction algorithm is combined with the SORT of vehicle detection counting algorithm, for highway traffic video data application background subtraction algorithm is used to detect the vehicle in the video, simple online real-time tracking (SORT) algorithm to detect the vehicle tracking, when the vehicle trajectory and traffic flow between two frame count line intersect, vehicle counter plus one, in the traffic flow in the video data. According to statistics, the accuracy of traffic flow counting results in this paper is 88%, which proves the feasibility and effectiveness of vehicle detection counting method based on background subtraction algorithm and SORT.

REFERENCES

- [1] Sinnu Susan Thomas, Sumana Gupta, Venkatesh K. Subramanian. Event Detection on Roads Using Perceptual Video Summarization[J]. IEEE Trans. Intelligent Transportation Systems, 2018, 19(9): 2944-2954.

[2] Dimitrakopoulos G, Demestichas P. Intelligent Transportation Systems[J]. IEEE Vehicular Technology Magazine, March 2010, Vol.5(1), pp.77-84.

[3] Nicholas a. Mandellos, Iphigenia Keramitsoglou, Chris t. Kiranoudis. A background subtraction algorithm for detecting and tracking vehicles[J]. Expert Systems with Applications,2010,38(3).

[4] A M B, A D T D, A G R. A deep convolutional neural network for video sequence background subtraction[J]. Pattern Recognition, 2018, 76:635-649.

[5] Xue R, Wu Z S, Bai A N. Application of Cloud Storage in Traffic Video Detection[C]// International Conference on Computational Intelligence & Security. IEEE, 2011.

[6] Zhihan Lv, Dongliang Chen, Ranran Lou, Qingjun Wang. Intelligent edge computing based on machine learning for smart city[J]. Future Generation Computer Systems,2021,115.

[7] Wei Zhang and Q.M. Jonathan Wu and Hai bing Yin. Moving vehicles detection based on adaptive motion histogram[J]. Digital Signal Processing, 2010.

[8] Wei Zhan, xiaolingji. Algorithm Research on Moving Vehicles Detection[J]. Procedia Engineering,2011,15.

[9] Sergey Menshov, Yan Wang, Andrey Zhdanov, Eugene Varlamov, Dmitry Zhdanov. Simple Online and Realtime Tracking people with new "soft-iou" metric[P]. Applied Optics and Photonics China,2019.

[10] Bewley A, Ge Z, Ott L, et al. Simple Online and Realtime Tracking[J]. arXiv, 2016.

[11] Yuanchang Xie,Yunlong Zhang,Zhirui ye. Short-Term Traffic Volume Forecasting Using Kalman Filter with Discrete Wavelet Decomposition[J]. Computer-Aided Civil and Infrastructure Engineering, 2007, 22 (5): 326-334. The DOI: 10.1111/1467-8667.2007.00489. J. x.

[12] Yuanfu Mo, Dexin Yu, Jun Song, Kun Zheng, Yajuan Guo, Elmetwally Elabbasy. Vehicle Position Updating Strategy Based on Kalman Filter Prediction in VANET Environment[J]. Discrete Dynamics in Nature and Society, 2016201, 6.

[13] KUHN H w. The method for the assignment problem[J]. Naval Research Logistics, 2005, 52 (1): 7-21.

[14] G. Pei, D. Cui, J. Li and M. Xia, "The research on multi-target matching," Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, Harbin, 2011, pp. 1515-1518, doi: 10.1109 / EMEIT. 2011.6023383.

[15] Lhoussein Mabrouk, Sylvain Huet, et al. Efficient parallelization of GMM background subtraction algorithm on a multi-core Platform for moving Objects Detection [C]. ATSIP, 2018:1-5.

[16] Silvio Ricardo Rodrigues Sanches, Claiton DE Oliveira, et al. Challenging Situations for Background algorithms[J]. Applied Intelligence, 2019, 49(5): 1771-1784.

[17] Lhoussein Mabrouk, Dominique Houzet, et al. Single Core SIMD Parallelization of GMM Background Subtraction for Vehicles Detection[C]. CIST, 2018:308-312.

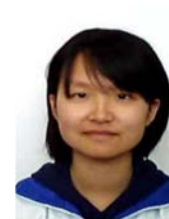
[18] Link W A, Schofield M R, Barker R J, et al. On the Robustness of N-mixture models[J]. Ecology, 2018, 99(7).

[19] Jung S, Kim Y, Hwang E. Real-time car tracking system based on surveillance videos[J]. EURASIP Journal on Image and Video Processing, 2018, 2018(1).

[20] Alberto Alvarellos; Adrian Vazquez. Juan Rabunal. Raspberry Pi Based Inersensor Data Processing System[J]. Proceedings, 2018.



Jun Guo, born in 1974, Ph.D., is an associate professor at Northeastern University. His research interests include edge computing, service computing and cloud computing.



Heyan Gao, Studying a undergraduate in software engineering at Northeastern University, from 2018 to 2022, Shenyang, China, her research interests include edge computing, cloud computing and so on..



Zeyu Yan, born in 2000, undergraduate, majored in software engineering from Northeastern University, Shenyang, China. His research interests include edge computing and cloud computing.



Jiahui Cao, study as an undergraduate student in major of Computer Science and Engineering from Northeastern University, Shenyang, China, from 2018 to 2022. Her research interests include edge computing, Computer Graphics and so on.



Zhenbo Fu, study as an undergraduate student in major of Computer Science and Engineering from Northeastern University, Shenyang, China, from 2018 to 2022. His research interests include edge computing, service computing and so on.