

Retraction

Retracted: Study of Clustering Algorithms in Object Tracking and Image Segmentation

Wireless Communications and Mobile Computing

Received 27 June 2023; Accepted 27 June 2023; Published 28 June 2023

Copyright © 2023 Wireless Communications and Mobile Computing. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

1. Discrepancies in scope
2. Discrepancies in the description of the research reported
3. Discrepancies between the availability of data and the research described
4. Inappropriate citations
5. Incoherent, meaningless and/or irrelevant content included in the article
6. Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] Y. Xing, L. Zhong, and X. Zhong, "Study of Clustering Algorithms in Object Tracking and Image Segmentation," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 7205929, 15 pages, 2022.

Research Article

Study of Clustering Algorithms in Object Tracking and Image Segmentation

Yongfeng Xing ^{1,2}, Luo Zhong ¹, and Xian Zhong ¹

¹School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan 430070, China

²School of Software, Nanyang Institute of Technology, Nanyang 473000, China

Correspondence should be addressed to Yongfeng Xing; xingyongfeng@whut.edu.cn

Received 10 March 2022; Accepted 18 April 2022; Published 16 May 2022

Academic Editor: Mohammad Farukh Hashmi

Copyright © 2022 Yongfeng Xing et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mean Shift is a kind of clustering algorithm, which is mostly used for target tracking, image segmentation, etc. In order to solve the problem that image information is not effectively utilized because of unclear traffic video images and random jitter between image sequences, this paper has studied how to achieve stability of traffic video images and proposed an improved Mean Shift algorithm about how to conduct object centroid registration in compensation for the deviations in space localization and, on this basis, how to select kernel window width to eliminate the errors in scale positioning. This algorithm gets the tracking effect and computation analysis of improved Mean Shift from the perspective of applications, removes or relieves the impact motion has on imaging, improves the quality of the video image information obtained, automatically adjusts the size of window according to the scale changes of moving object in the image, and effectively enhances the stability and real time of object tracking. Besides, in the postprocessing stage, the superpixel based on the Mean Shift algorithm is applied to further optimize the segmentation result; it is a popular mode-seeking clustering algorithm, which makes Mean Shift method ideal on low-dimensional applications such as image segmentation. Finally, we show promising results for remote sensing image segmentation.

1. Introduction

Object tracking refers to the instant velocity of pixel motion of moving object in observation and imaging plane, which makes use of time-domain changes and correlation of image sequence pixel intensity to pin down the motion of the position of each pixel; in other words, it reflects the relationship between the changes of image grayscale over time and the structure and motion of the object in the scene [1]. Traffic video object tracking has already become a hot but difficult topic in the researches about computer vision in recent years. As imaging devices are affected by postures and vibrations, the images to be obtained will not be clear and there will be random jittering between image sequences, thus severely affecting effective utilization of image information [2]. Traffic video image stabilization technique will provide a relatively stable coordinate system for the measurement plane of the instrument in order to ensure accurate measurement results, reduce the possibility of blurring images caused

by involuntary motion, and improve imaging quality [3]. Video object tracking now has already been widely applied in a variety of fields, such as intelligent transportation system and video surveillance, so it is of great significance to study an object tracking algorithm with good real time, strong robustness, and high accuracy. In video image sequences, there are two kinds of motion: one is global motion, namely, the change of the whole image caused by the changes in the positions or parameters of cameras, and the other is local motion, i.e., the change of local image caused by the motions of the object in the scene [4]. Both purposeful and unnecessary motions of cameras will lead to global motion of image sequences, and image stabilization is to eliminate the unnecessary camera motion. Therefore, the global motion of image sequences requires to be estimated. Among object tracking algorithms, such as deep learning and correlation filtering, Mean Shift algorithm has attracted in-depth study and close attention at home and abroad because of its easy implementation and good real

time [5]. Correlation tracking algorithms have been widely used in tracking systems for their simple and practical characters. Correlation filter based on object tracking method has been widely used for its high efficiency. However, reducing model drifting while achieving both high robustness and fast scale estimation is still an open problem. Different from the trend of deep learning in the visual field such as detection and recognition, the application of it in the field of object tracking is not good yet. The main problem is that the lack of large datasets of deep learning causes slower speed in object tracking tasks than that of the Mean Shift algorithm.

Semantic segmentation is a classic and fundamental topic in computer vision. It is a pixel-level classification task, which plays an important role in many fields such as autonomous driving, video surveillance, geographic information systems, and medical image analysis [6].

The special contributions of this paper include the following:

- (i) To analyze and study the stability of traffic video images, eliminate unnecessary camera motion, and obtain smooth object motion of camera so as to achieve image stabilization and smooth camera motion
- (ii) To introduce common object tracking algorithms and procedures, study the difficulties and performance index of object tracking technology, and focus on the analysis and research of Mean Shift object tracking algorithm
- (iii) To study the descriptions of object model and candidate model of traffic video image and weigh the difference or similarity between these two kinds of models. Based on the object localization in traffic video image, search the real position of the object in the current frame
- (iv) Based on the in-depth study of the principle of related object tracking algorithms, propose an improved Mean Shift object tracking algorithm, select different video sequences, conduct detailed analysis and comparison of algorithm performance through experiment, and verify their strengths, weaknesses, and application scope
- (v) Experimental results on widely used high-resolution remote sensing datasets for semantic segmentation tasks, ISPRS Vaihingen, demonstrate competitive performance obtained by the proposed methodology compared to other studied approaches

2. Related Work

With the extensive applications of object tracking techniques, the requirements on object tracking algorithms may be different; in general, there are three basic requirements: accuracy, robustness, and real time. Accuracy includes accuracy on object detection and on object tracking, and it is an important index in the application of surveil-

lance [7]. Image stability is constituted by global motion estimation, motion smoothness, and image reconstruction, and it defines the instant rate of gray change in specific coordinate point in the plane of 2D image as vector. Object tracking refers to the apparent motion of image grayscale mode. It is a 2D vector field, and the information it contains is the vector information of instant motion velocity of each pixel point. In object tracking, each pixel has a motion vector; so it can reflect the motion between neighboring frames. In the recognition of behaviors of the object in traffic video image, robustness is to maintain continuous and stable tracking on the moving object under different external factors [8]. This is also a significant index of surveillance and tracking; meanwhile, as surveillance system usually needs to automatically and constantly monitor the object for a long time, its ability against bumpy roads and noises requires to be improved. For those systems in which real-time and high-speed surveillance is required, computation speed, i.e., real-time, is extremely important; therefore, it appears to be particularly important for how to select effective algorithm to reduce losses and improve antijamming capability of object tracking [9–11].

The concept of Mean Shift was firstly proposed by Fukunaga and Hostetler in an article concerning estimation of probability density gradient function in 1975. It is a non-parametric estimation algorithm, and it searches the peak of distribution along the ascent direction of probability gradient and defines a cluster of kernel functions, the deviation of which makes different contributions to Mean Shift vector over different distances between the sample and the deviated point [12]. Besides, it also sets a weight coefficient, making different sample points have different importance. This has expanded the range of application of Mean Shift. As research is deepened about Mean Shift, it has evolved into a process from a vector and The Mean Shift algorithm generally refers to an iterative process; in other words, firstly, it computes the Mean Shift of the current point, according to which, it moves this point; then, it takes the moved point as the benchmark, continues to compute the Mean Shift, and repeats the above steps in iteration until it meets certain conditions and ends. The Mean Shift algorithm through the kernel function density estimation method of nonparametric density estimation converges the mode centroid of density along the direction of density gradient quickly and effectively [13]. If Mean Shift algorithm satisfies certain conditions, it can converge to the nearest stable centroid of probability density function; so, the Mean Shift algorithm can be used to detect the mode in probability density function. Some scholars have combined scale space with Mean Shift to solve object tracking in real-time change of object size, but its speed is not ideal enough. The Mean Shift algorithm does not have much computation and it can perform real-time tracking when the object region is known. It uses histogram model of kernel function, and it is not sensitive to edge occlusion, object rotation, deformation, and background motion. And as the window width remains the same in the tracking process, the frame region will not expand (or shrink) over the expansion (or shrinkage) of the object. When the object moves fast, the tracking effect is not good

with a lack of histogram features in description of object color features and space information [14–16].

3. Stability of Traffic Video Image

Motion smoothness is to remove unnecessary camera motion and obtain smooth camera object motion so as to achieve image stabilization. Generally, take the estimated motion parameters of the camera as the noisy data of object motion parameters and use low-pass filter to get rid of high-frequency noises in camera motion parameters so as to achieve the purpose of smooth camera motion.

After completing motion smoothness, it can perform calibration transformation on every frame of image in the video according to the difference between the preliminarily estimated camera motion parameters and the smoothed camera motion parameters so as to reconstruct stable video sequences.

- (1) First, preprocess the image data and produce gray coding bit plane from standard binarization. Make the t th image in the K -bit gray image sequence as

$$f^t(x, y) = a_{K-1}2^{K-1} + a_{K-2}2^{K-2} + \dots + a_02^0 \quad (1)$$

The bit plane of this image is represented by the method from the least significant bit $b_0^t(x, y)$ to the most significant bit $b_7^t(x, y)$. The 8-bit gray coding can be provided by the following formula:

$$\begin{aligned} g_{K-1} &= a_{K-1}, \\ g_k &= a_k a_{k+1}, \quad 0 \leq k \leq K-2, \end{aligned} \quad (2)$$

in which a_k is the standard binary coefficient. In this algorithm, gray coding is very important and it makes the next motion estimation possible as this module divides most useful image information coding into several planes and uses one single bit plane for motion estimation. After gray coding, a tiny change in grayscale will lead to correspondingly uniform changes in the binary digit which represents intensity. Therefore, an extremely significant Boolean binary operation can be used to compare the gray coding bit plane of the previous image and the current image.

- (2) Use Boolean operation for the motion estimation in local region, namely, self-motion estimation. Make the error operator to be minimized as

$$\varepsilon(m, n) = \frac{1}{(W^2) \sum \sum g_k^t(x, y) g_k^{t-1}(x+m, y+n)}, \quad -p \leq m, n \leq p, \quad (3)$$

in which W is the slider used for search and p is the number of pixels to be searched

By minimizing this error operator, the value of (m, n) obtained represents the self-motion vector. Then, process

these 4 self-motion vectors and the previous global motion vector with mean operator and obtain the current global motion vector estimation V_g^t .

Use filter for global motion estimation, fusion, and motion compensation. Then, get this global motion estimation through a specially designed filter, which can preserve the conscious motion of camera, such as conscious translation, and which can also remove the useless high-frequency motion. Finally, the filtered estimation result can make the compensation by moving the current frame by pixels of an integer value along the opposite direction of the motion.

4. Implementation of Mean Shift Algorithm

4.1. Kernel Function. Definition of kernel function: X represents a d -dimensional Euclidean space and x is a point in this space, which is represented by a column vector. The mode of x is $\|x\|^2 = x^T x$. R represents the field of real numbers. If there is a section function $k : [0, \infty] \rightarrow R$ in a function $K : X \rightarrow R$, namely,

$$K(x) = k(\|x\|^2), \quad (4)$$

and if the following conditions are satisfied:

- (1) k is nonnegative
- (2) k is nonincreasing, namely, if $a < b$, then $k(a) \geq k(b)$
- (3) k is sectionally continuous and $\int_0^\infty k(r) dr < \infty$

Then, the function $K(x)$ is called as Kernel function.

In Mean Shift, the following three kinds of Kernel functions are very frequently used.

- (a) Uniform kernel

$$K_U(x) = \begin{cases} c & \|x\| \leq 1, \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

- (b) Normal kernel

$$K_N(x) = c \cdot \exp\left(-\frac{1}{2} \|x\|^2\right) \quad (6)$$

- (c) Epanechnikov kernel

$$K_E(x) = \begin{cases} c(1 - \|x\|^2) & \|x\| \leq 1, \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

These three kinds of Kernel functions are as follows [17], as shown in Figure 1.

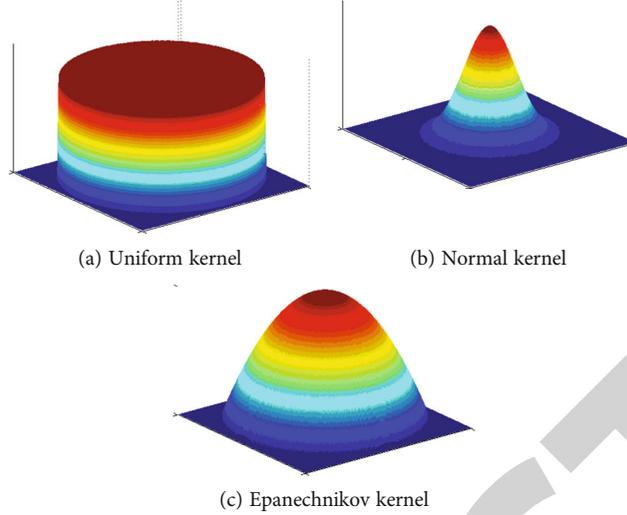


FIGURE 1: Three kinds of kernel functions.

4.2. Mean Shift Vector. It can be seen from Formula (3) that the sampling points falling into S_h make the same contribution to the computation of the final $M_h(x)$ whether it is close or far away from x . However, we all know that generally speaking, the closer the sampling points are away from x , the more significant they are to estimate the statistical characteristics surrounding x . Therefore, we introduce the concept of kernel function and when computing $M_h(x)$, the impact of distance can be taken into consideration; in the meanwhile, we can also suppose that among all these sampling points x_i , they are significant in different ways, so, we have introduced a weight coefficient for every sample.

In this way, we can expand the basic Mean Shift form into

$$M(x) \equiv \frac{\sum_{i=1}^n G_H(x_i - x)w(x_i)(x_i - x)}{\sum_{i=1}^n G_H(x_i - x)w(x_i)}, \quad (8)$$

in which $G(x)$ is a unit kernel function; H is a positive definite and symmetric $d \times d$ matrix, and we usually call it window width matrix; and $w(x_i) \geq 0$ is a weight given to the sampling point x_i , $G_H(x_i - x) = |H|^{-1/2}G(H^{-1/2}(x_i - x))$.

In practical applications, the window width matrix H is generally restricted to a diagonal matrix $H = \text{diag}[h_1^2, \dots, h_d^2]$ or even simply directly proportional to unit matrix, i.e., $H = h^2I$. As the latter form only needs to confirm a coefficient h , Formula (8) can also be written as follows:

$$M_h(x) \equiv \frac{\sum_{i=1}^n G((x_i - x)/h)w(x_i)(x_i - x)}{\sum_{i=1}^n G((x_i - x)/h)w(x_i)}. \quad (9)$$

We can see that if all sampling points x_i satisfy the following conditions:

$$(1) \quad w(x_i) = 1$$

$$(2) \quad G(x) = \begin{cases} 1 & \text{if } \|x\| < 1, \\ 0 & \text{if } \|x\| \geq 1 \end{cases}$$

Mean Shift vector $M_h(x)$ is the normalized probability density gradient and iterative Mean Shift algorithm will converge to a stable centroid of probability density function [18].

4.3. Probability Density Gradient. For a probability density function $f(x)$, it is known that there are n sampling points $x_i, i = 1, \dots, n$, in d -dimensional space, and then, the kernel function estimation (also called as Parzen windowing estimation) of $f(x)$ is as follows:

$$\hat{f}(x) = \frac{\sum_{i=1}^n K((x_i - x)/h)w(x_i)}{h^d \sum_{i=1}^n w(x_i)}, \quad (10)$$

in which $w(x_i) \geq 0$ is a weight given to sampling point x_i and $K(x)$ is a kernel function and it satisfies $\int k(x)dx = 1$.

Besides, the section function of the kernel function $K(x)$ is $k(x)$ and it makes

$$K(x) = k(\|x\|^2). \quad (11)$$

The negative derived function of $k(x)$ is $g(x)$, namely, $g(x) = -k'(x)$, and its corresponding kernel function is as follows:

$$G(x) = g(\|x\|^2). \quad (12)$$

The estimation of gradient $\nabla f(x)$ of probability density function $f(x)$ is

$$\hat{\nabla} f(x) = \nabla \hat{f}(x) = \frac{2 \sum_{i=1}^n (x - x_i)k'(\|(x_i - x)/h\|^2)w(x_i)}{h^{d+2} \sum_{i=1}^n w(x_i)}. \quad (13)$$

According to the definition above, $g(x) = -k'(x)$ and $G(x) = g(\|x\|^2)$. So, the formula above can be written as

$$\begin{aligned}\widehat{\nabla}f(x) &= \frac{2\sum_{i=1}^n(x_i-x)G(\|(x_i-x)/h\|^2)w(x_i)}{h^{d+2}\sum_{i=1}^n w(x_i)} \\ &= \frac{2}{h^2} \left[\frac{\sum_{i=1}^n G((x_i-x)/h)w(x_i)}{h^d \sum_{i=1}^n w(x_i)} \right] \\ &\quad \cdot \left[\frac{\sum_{i=1}^n (x_i-x)G(\|(x_i-x)/h\|^2)w(x_i)}{\sum_{i=1}^n G((x_i-x)/h)w(x_i)} \right].\end{aligned}\quad (14)$$

The part in the second square bracket in the right hand of the above formula is the Mean Shift vector as defined in Formula (9), and the part in the first square bracket is the estimation of probability density function $f(x)$ with $G(x)$ as the kernel function, which is marked as $\widehat{f}_G(x)$. Then, $\widehat{f}(x)$ as defined in Formula (10) is marked as $\widehat{f}_K(x)$, so Formula (12) can be changed into

$$\widehat{\nabla}f(x) = \nabla\widehat{f}_K(x) = \frac{2}{h^2}\widehat{f}_G(x)M_h(x). \quad (15)$$

It can be seen from Formula (13) that

$$M_h(x) = \frac{1}{2}h^2 \frac{\nabla\widehat{f}_K(x)}{\widehat{f}_G(x)}. \quad (16)$$

Formula (16) shows that the Mean Shift vector $M_h(x)$ computed and obtained at point x with kernel function G is proportional to the normalized gradient of probability density function $\widehat{f}_K(x)$ estimated with kernel function K and the normalization factor is the probability density of point x estimated with kernel function G . Therefore, Mean Shift vector $M_h(x)$ always points to the direction the probability density increases the most.

The Mean Shift algorithm is an iterative step. In another word, it computes the Mean Shift of the current point. Then, this point is moved to this Mean Shift and it will become a new starting point. Continue to move until it satisfies the final conditions [19]. This process can be seen in Figure 2 below.

It can be seen from the above process that the Mean Shift is computed in the designated region. Move this point to the point of Mean Shift. Compute the new Mean Shift, and move it. Repeat this process until it meets the final conditions. Then, quit. In the Mean Shift algorithm, the most important key is to compute the Mean Shift of every point and then update the position according to the new Mean Shift.

4.4. Object Tracking Based on Mean Shift. The Mean Shift algorithm can initialize the object to be tracked by auto or semiauto methods. Semiauto is to manually designate the object region of interest in the initial frame while auto is to get the object region to be tracked through object detection method. Object region is the region where kernel function

works, and its size is the window width (scale) of the kernel function. When the Mean Shift algorithm tracks the object, the object features usually choose color feature of the object. Real-time processing shall meet the requirement of low computation, and it can divide every subspace under certain color space into k equal regions, namely, k bins. The color histogram feature of the possible object region in the video image is the description of the candidate model. Generally, Epanechnikov function is chosen as the kernel function. Use the proper similarity function to compute the similarity of the candidate model and the object model, and use the maximum value of similarity function to get the Mean Shift vector, i.e., the shift of object in the current frame. The object will eventually converge to the real position after several iterations so as to achieve the purpose of tracking.

4.4.1. Description of Object Model. Assume that $\{x_i\}_{i=1,2,\dots,n}$ represents the n pixels in the object region and the coordinate of the object center point is x_0 ; then, the probability distribution density of all pixel points in this object region is q_u .

$$\widehat{q}_u = C \sum_{i=1}^n k\left(\left\|\frac{x_i^s - x_0}{h}\right\|^2\right) \delta[b(x_i^s) - u], \quad (17)$$

in which $k(x)$ is the kernel function and it is used to represent the weight of object color histogram. The closer it is to x_0 , the bigger the weight. h is the window width of the kernel function, and $b(x)$ is the color feature of the object region x_i . $\delta(x)$ is a 1D delta function, and it is used to judge whether the index value of x_i is equal to the feature value u , and its value is 1 when they are equal and otherwise, 0. C is the normalization constant.

$$C = \frac{1}{\sum_{i=1}^n k(\|(x_i - x_0)/h\|^2)}. \quad (18)$$

4.4.2. Description of Candidate Model. The possible object region after the initial frame is the candidate region. Make its central position as y , and the pixels in the region are represented with $\{x_i\}_{i=1,2,\dots,n}$. Compare the description of object model, and the probability density of the candidate model is as follows.

The candidate object located in y can be described as

$$\widehat{p}_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{x_i^s - y}{h}\right\|^2\right) \delta[b(x_i^s) - u], \quad (19)$$

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k(\|(x_i - x_0)/h\|^2)}. \quad (20)$$

4.4.3. Similarity Function. Similarity function is used to measure the difference or similarity between the object model and the candidate object model. In the Mean Shift algorithm, Bhattacharyya coefficient is the similarity coefficient and it refers to the approximate computation of the overlapping of two statistical samples and it is used to measure the correlation of two groups of samples [19]. Therefore, object



FIGURE 2: Schematic diagram of the Mean Shift algorithm.

tracking can be simplified into the search for the optimal y to make $\hat{p}_u(y)$ most approximate to \hat{q}_u .

The distribution of the similarity between $\hat{p}_u(y)$ and \hat{q}_u is measured by Bhattacharyya coefficient $\hat{\rho}(y)$, namely,

$$\hat{\rho}(y) \equiv \rho[p(y), q] = \sum_{u=1}^m \sqrt{p_u(y)q_u}. \quad (21)$$

Perform Taylor expansion at point $\hat{p}_u(\hat{y}_0)$ on Formula (21), and obtain

$$\rho[p(y), q] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{p(y_0)q_u} + \frac{1}{2} \sum_{u=1}^m p_u(y) \sqrt{\frac{q_u}{p_u(y_0)}}. \quad (22)$$

Bring Formula (19) into the formula, and get

$$\rho[p(y), q] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{p(y_0)q_u} + \frac{C_h}{2} \sum_{i=1}^n w_i k\left(\left\|\frac{y-x_i}{h}\right\|^2\right), \quad (23)$$

in which

$$w_i = \sum_{u=1}^m \delta[b(x_i) - u] \sqrt{\frac{q_u}{p_u(y_0)}}. \quad (24)$$

4.4.4. Object Localization. To search the real position of the object in the current frame, it is to find the largest candidate region. First, localize the object center in the previous frame as the object center of current frame and search the optimal matching region starting from this point. Make the center as y . Assume that various feature distribution densities of object candidate region are $\{p_u(y)\}_{u=1,2,\dots,m}$ and the coordinate of the central point of the object in the last frame is y_0 ; then, the central position of the tracking object in the current frame is y_1 .

$$y_1 = \frac{\sum_{i=1}^m x_i w_i g(\|(y_0 - x_i)/h\|^2)}{\sum_{i=1}^m w_i g(\|(y_0 - x_i)/h\|^2)}, \quad (25)$$

in which $g(x) = -k'(x)$ and w_i is the weight.

$$w_i = \sum_{u=1}^m \sqrt{\frac{q_u}{p_u(y_0)}} \delta(b(x_i) - u). \quad (26)$$

Compute the value of y_1 through iteration. When the two neighboring iterative values are smaller than a certain threshold, the position in the current frame of the tracking object can be confirmed and the real position of the object can be found after several iterations.

5. Application of Improved Mean Shift Algorithm in Object Tracking of Traffic Video Image

The Mean Shift algorithm is a nonparametric method of density function gradient estimation, and it localizes the object by obtaining the local maximum of density estimation function through iteration. The Mean Shift algorithm is fast in matching and needs no parameters so that it has been widely applied in the field of video object tracking. However, its object model is fixed, preventing the window width of kernel function from being updated timely and hurting the tracking accurately. If the size of motion object changes too significantly, the window width of fixed kernel function will lead to inaccurate object tracking or even lose the object tracking.

To realize the practicability of object tracking, detailed analysis is made of the computation amount when applying the algorithm of this paper in object tracking. Real time has a high requirement in traffic video. Describe the motion object with grayscale and color space HS (hue and saturability). Assume that the center of the object is located in x_0 ; then, the object is represented with the following formula:

$$\hat{q}_u = C \sum_{i=1}^n k \left(\left\| \frac{x_i^s - x_0}{h} \right\|^2 \right) \delta[b(x_i^s) - u]. \quad (27)$$

The candidate object located in y can be described as

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k \left(\left\| \frac{x_i^s - y}{h} \right\|^2 \right) \delta[b(x_i^s) - u]. \quad (28)$$

Therefore, object tracking can be simplified into the search for the optimal y , making $\hat{p}_u(y)$ the most similar to \hat{q}_u . The most similarity between $\hat{p}_u(y)$ and \hat{q}_u is measured by Bhattacharyya coefficient $\hat{\rho}(y)$, namely,

$$\hat{\rho}(y) = \rho[p(y), q] = \sum_{u=1}^m \sqrt{p_u(y)q_u}. \quad (29)$$

Perform Tylor expansion to Formula (29) at point $\hat{p}_u(\hat{y}_0)$, and obtain

$$\rho[p(y), q] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{p(y_0)q_u} + \frac{1}{2} \sum_{u=1}^m p_u(y) \sqrt{\frac{q_u}{p_u(y_0)}}. \quad (30)$$

Put Formula (28) into the above formula, and obtain

$$\rho[p(y), q] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{p(y_0)q_u} + \frac{C_h}{2} \sum_{i=1}^n w_i k \sqrt{\left\| \frac{y - x_i}{h} \right\|^2}, \quad (31)$$

in which $w_i = \sum_{u=1}^m \delta[b(x_i) - u] \sqrt{q_u / (p_u(y_0))}$ optimize Formula (29) and weight coefficient.

The process of this algorithm is as follows: give an initial point x , kernel function $G(x)$, and allowable error ε . Through iterations, continuously move along the gradient

direction of probability density until converging to the peak density in data space. The step length in iteration is not only related to the size of gradient but also related to the probability density of this point, and it is a gradient ascent algorithm with changeable step length.

Assume A is a finite set embedded in n -dimensional Euclidean space x . The Mean-Shift vector at $x \in X$ is defined as

$$ms = \frac{\sum_a k(a-x)\omega(a)a}{\sum_a k(a-x)\omega(a)} - x, \quad a \in A, \quad (32)$$

in which k is kernel function and W is weight. The reverse direction of Mean-Shift vector ms computed at X points to the gradient direction of convolution surface:

$$J(x) = \sum_a g(a-x)\omega(a), \quad (33)$$

in which g is shadow kernel. Move the central position of kernel function continuously along the direction of ms until convergence, and find the neighboring mode matching position.

5.1. Affine Model and Feature Point Matching. Here, only two types of common motion are considered: translation and scale. So, the affine model of the object is given by the following form:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_x \\ e_y \end{pmatrix}, \quad (34)$$

in which $(x, y)'$ and (x', y') are the positions of the same object feature point in frames i and $i+1$, respectively. $e = \{e_x, e_y\}$ is the translation parameter, and $s = \{s_x, s_y\}$ is the scale amplitude in the horizontal and vertical directions of the object. By making use of s , the kernel window width can be updated by the following method:

$$r = r \max(s_x, s_y). \quad (35)$$

For rigid object, corner points can better depict its spatial structure and are easy to detect. Therefore, the matching corner points in the two frames are used as samples to estimate the parameters of the affine model.

Assume that there are N and N' corner points in the tracking windows T_i and T_{i+1} of frames i and $i+1$, respectively. T_i and T_{i+1} have the same radius, and both their centers coincide with the corresponding object centroids. In other words, the object centroid has already gone through registration in the adjacent two frames. Given a corner point $P_c|T_i$ located in T_i , its corresponding corner point $P_c|T_{i+1}$ in T_{i+1} shall satisfy the following:

$$I(P_c|T_{i+1}) = \min \{ |I(P|T_i) - I(P_j|T_{i+1})| \}_{j=1,2,\dots,n}, \quad (36)$$

in which I is the grayscale of the pixel. n candidate points are located in a small given window G with the position of $P_c|T_i$ as the center in frame T_{i+1} .

5.2. Object Centroid Registration. Assume that in frame i , the object with o_i as the centroid is selected by the initial tracking window T_i with $c_i = o_i$ as the center. When the object of increased size appears in frame $i + 1$, for the current object centroid o_{i+1} , there should be a deviation of $d = c_{i+1} - o_{i+1}$. Here, c_{i+1} is the center of the tracking window T_{i+1} in frame $i + 1$. This deviation is the deviation of spatial localization caused by the Mean Shift tracking algorithm of fixed kernel window width. As the size of the current object is larger than that of T_{i+1} , it only contains certain part of the object. At last, the position of the object centroid in frame $i + 1$ is estimated as follows:

$$o_{i+1} \leq c_{i+1} - d'. \quad (37)$$

Therefore, when an object image is given, Formula (37) can be used to register the object centroid in the current frame and then use Formula (36) to match corner point. This processing method can effectively remove mismatching points.

5.3. Algorithm Description

Step 1. Get the initial tracking window T_i after selecting the object in frame i , and perform Mean Shift tracking in frame $i + 1$ to obtain T_{i+1} .

Step 2. With T_{i+1} as the initial tracking window, perform Mean Shift tracking in frame i and obtain T'_i .

Step 3. Move T_{i+1} according to the position of central point between T_i and T'_i , and expand its size $r = r \cdot \varepsilon$, $\varepsilon > 1$.

Step 4. Extract the corner point for matching from T_i and T_{i+1} .

Step 5. Perform regression to the horizontal and vertical coordinates of matching point and get s_x and s_y .

Step 6. Update the size of T_{i+1} with $r = r \cdot \max(s_x, s_y)$.

6. Application of Mean Shift Algorithm in Remote Sensing Image Semantic Segmentation

The Mean Shift algorithm has also been successfully applied in many other fields, such as image smoothing and image segmentation, which belong to the part of pattern recognition or computer vision in artificial intelligence.

In the postprocessing stage, a super pixel algorithm combined with traditional machine learning is proposed to further optimize the segmentation results [20].

Although the deep convolution neural network has achieved good segmentation results in the semantic segmen-

tation of remote sensing images, due to the inherent nature of the deep convolution neural network, its segmentation results often still have the following shortcomings:

- (1) For high-resolution remote sensing images, due to the rich structure and texture of ground objects, the deep convolution neural network is not accurate for the semantic boundary segmentation of ground objects. This is due to the following reasons: in order to output the segmentation results consistent with the size of the original image, the deep convolution neural network generally uses upsampling or deconvolution to improve the size of the output feature image
- (2) In remote sensing images, an object is often composed of a series of regions. In these areas, the color, illumination, and texture of the object change very little. Because the final segmentation result of deep semantic segmentation network is to classify each pixel, there is no relevant information that these regions belong to the same object

In order to solve the above problems, we propose a region method based on Mean Shift algorithm superpixel segmentation for the postprocessing of deep convolution neural network semantic segmentation results, so as to improve the accuracy of remote sensing image semantic segmentation and the accuracy of boundary pixel classification.

In the semantic segmentation network proposed in this paper, the segmentation performance of the network itself has reached a high level. In order to further improve the segmentation accuracy through postprocessing, an improved conditional random field based on traditional machine learning superpixel algorithm (SLIC) is proposed.

Superpixel algorithm is an image oversegmentation method, which can aggregate the pixels in the image into irregular blocks according to their texture, color, brightness, and other characteristics. Each block can well reflect the boundary of the object.

According to this characteristic, the oversegmentation template obtained by the superpixel algorithm is first applied to the network prediction results, and the categories of all pixels in each over segmented block are unified into the classes with the largest number. Then, the optimized prediction results and the input RGB image are sent to the condition ci in Figure 3.

The detailed Algorithm 1 flow is as follows.

7. Experiment Result and Analysis

Turn the video sequence image with vibrations into stable video sequence image. After the object affine model is obtained through regressive calculation, the current kernel window width is updated. The updated kernel window width is used to, on the one hand, correct the size of the current tracking window so as to reduce the deviations of scale localization and, on the other hand, determine the number of samples in Mean Shift iteration in tracking the next frame.

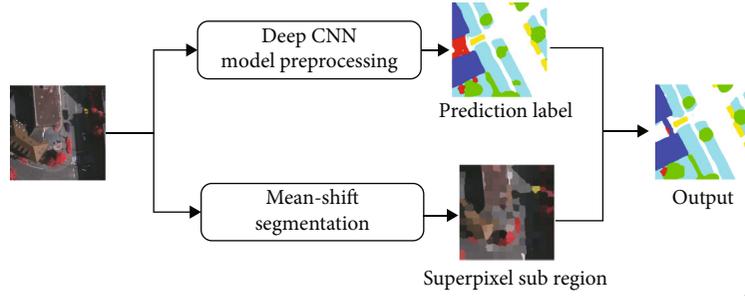


FIGURE 3: The architecture of our proposed method combining DCNN and superpixel segmentation.

- (1) The image to be segmented input to the network is defined as I , and the prediction image result of the network is defined as O . All object classes of the dataset are defined as $C = \{c_1, c_2, \dots, c_k\}$.
- (2) Calculate the superresolution algorithm of image I , and generate superpixel blocks with Mean Shift method, which are defined as $S = \{s_1, s_2, \dots, s_k\}$. s_j is one of the super-pixel block.
- (3) For $i = 1 : M$
All N pixels in S_i are defined as $T = \{t_{i1}, t_{i2}, \dots, t_{ik}\}$
Get the prediction result I_{ij} ($I_{ij} \in C$) of each pixel of p_{ij} .
Count the number of pixels of each classes in s_j , and the result is $\{\text{count}(c_1), \text{count}(c_2), \dots, \text{count}(c_j)\}^*$
- (4) Change the prediction classes of each pixel p_{ij} in s_i to
 $Z_{s_i} = \max \{\text{count}(c_1), \text{count}(c_2), \dots, \text{count}(c_j)\}$
End for
- (5) Update the output image to Z .
- (6) Calculate the input image and the updated prediction result image Z to obtain the final prediction result L_{out} .

ALGORITHM 1

In this way, the system can better get adjusted to the change of object scale and overcome the restrictions of fixed kernel window width. Experiment 1 is the tracking experiments on bumpy motion object in traffic video through the algorithm of this paper and the Mean Shift tracking algorithm.

Experiment 1. Use the video with a frame rate of 30 frames/second and a resolution of $256 * 256$ where the vehicle goes on bumpy road. The tracking effect of the Mean Shift algorithm is shown in Figure 4 while that of the algorithm in this paper is shown in Figure 5.

By comparing the motion object tracking effects by the algorithm of this paper and the Mean Shift algorithm, it is clearly evident that both algorithms can perform accurate tracking on the object the size of which cannot change dramatically. But as the algorithm of this paper can automatically adjust the kernel function window width, it can frame the object more accurately, namely, that such features as the color of the object obtained are more accurate, and it improves the tracking robustness. As a nonparametric density estimation method, it is clear that the tracking algorithm of this paper has better accuracy and stability on object tracking. Besides, it does not need much computation and it can satisfy the real-time requirement.

Figure 4 indicates the effects of motion object tracking of the 3rd, 16th, 42nd, and 66th frames in the video using the

Mean Shift algorithm, and the red circles have marked the final object position determined by this algorithm. We can see that although the tracking box can always track the object, the effect is not ideal; it may lose the object if the object becomes increasingly big.

Figure 5 shows the effects of motion object tracking of the 3rd, 16th, 42nd, and 66th frames in the video using the algorithm of this paper, and the red circles have marked the final object position determined by the algorithm of this paper. Compared with the Mean Shift algorithm, it can be seen that the algorithm mentioned in this paper can perform complete tracking of the entire motion object and it can also better extract the feature information of the motion object and effectively prevent the loss of object.

Experiment 2. The video is a real video of a car driving. The frame rate of the video is 30 frames per second, and the resolution is $1280 * 720$. The car is the object we need to track, and the scale of the object is getting larger. Intercepted 5th, 47th, 76th, and 119th frames were intercepted for analysis. Figure 6 is the experimental result of the Mean Shift algorithm, and Figure 7 is the experimental result of our algorithm.

As shown in Figure 6, we can see that the traditional Mean Shift algorithm gradually loses most of the pixels of the tracking object when the object becomes larger.



FIGURE 4: Tracking effect of bumpy vehicle via the Mean Shift algorithm.

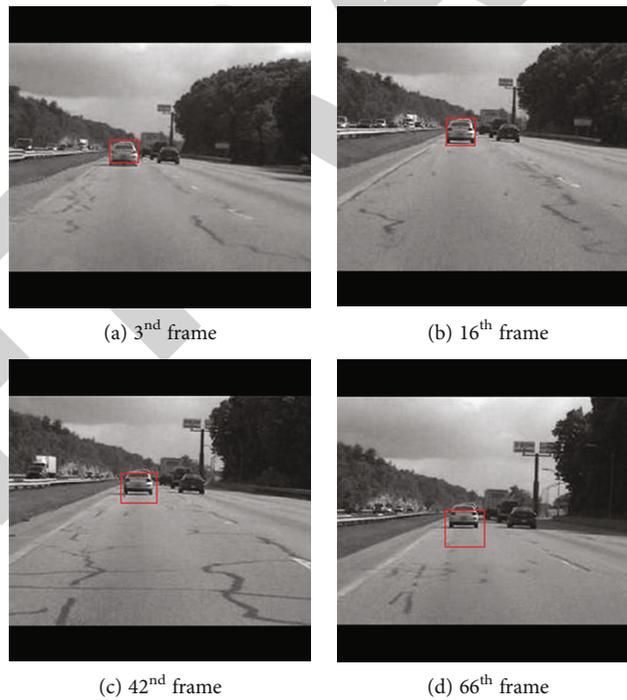


FIGURE 5: Tracking effect of bumpy vehicle via the algorithm of this paper.

Although the window becomes larger with the increase of the object, there are serious deviations in the location of the object center. Figure 7 adopts the algorithm in this paper. The object scale can be adjusted adaptively. It can not only update the object scale accurately but also locate the center of the object accurately. Besides, Figure 8 shows

the comparison curves about the size of object tracking box (unit: pixel) and the real size of the object through the algorithm of this paper and the Mean Shift algorithm.

In Figure 8, the blue “◆” line represents the actual scale of the object, the green “▲” line represents the Mean Shift algorithm, and the red “■” line represents the algorithm in



FIGURE 6: Tracking effect of vehicle from far to near by the Mean Shift algorithm.



FIGURE 7: Tracking effect of vehicle from far to near by our algorithm.

this paper. It can be clearly seen from the graph that the object scale of the Mean Shift algorithm is increasing, but it is far away from the actual object scale, and the algorithm in this paper is very close to the actual object scale. Table 1 is the statistics of relative error of size of some frame object tracking boxes in the video sequences and after analysis.

It can be observed from above experimental results that the tracking algorithm in this paper will have deviation in spatial localization when the size of object keeps growing and it is bigger than the kernel window width. After this

deviated position is obtained, back processing can be conducted, namely, backtracking. At this time, reverse video sequence is equivalent to gradual shrinkage in object size. In this way, it can accurately identify the corresponding deviated point in the reverse frame, making compensation for this deviation possible. So, the change of translation and scaling is obtained for the position of window, making the object window adaptively change over the changing object size. Based on the assumption that the rigid motion object in consecutive frames meets the affine model, firstly,

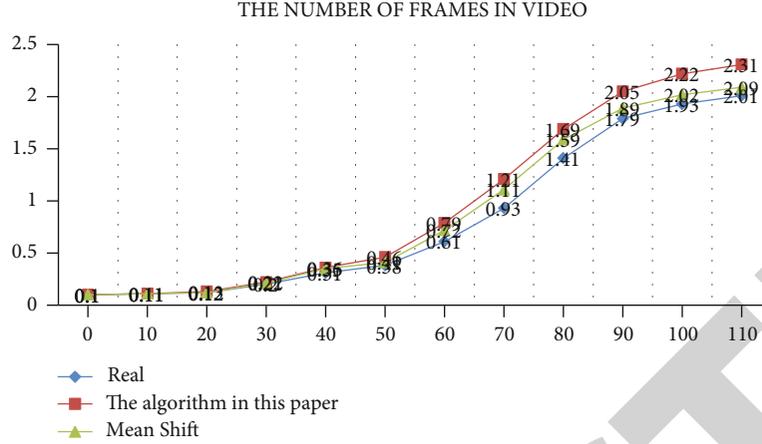


FIGURE 8: Analysis diagram of the algorithm in this paper and Mean Shift tracking window size.

TABLE 1: Error comparison between the algorithm of this paper and Mean Shift tracking box.

	Real size/pixel of object	Size/pixel of this algorithm tracking box	Size/pixel of mean shift tracking box	This algorithm relative error (%)	Mean shift relative error (%)
1 st frame	567	616	619	8.64%	9.17%
5 th frame	609	650	896	6.73%	47.13%
47 th frame	2280	2472	2754	8.42%	20.79%
76 th frame	5626	6258	6732	11.23%	19.66%
119 th frame	18158	19573	20198	7.79%	11.23%
131 st frame	19636	21289	22815	8.41%	16.19%

TABLE 2: Comparison results on Vaihingen dataset ζ .

Vaihingen	OA (%)	Precision	Recall	Mean F_1
FCN	88.11	84.56	85.91	85.68
SegNet	90.07	88.93	88.83	88.62
BiseNet	90.13	88.79	88.89	88.71
DeepLabv3+	90.06	88.95	88.97	88.67
FCN+*	88.33	84.79	86.02	85.86
SegNet +*	90.19	88.92	88.86	88.89
BiseNet+*	90.21	88.93	88.97	88.67
DeepLabv3+*	90.25	88.94	88.99	88.97

DeepLabv3+* stands for DeepLabv3 + and superpixel segmentation method based on Mean Shift.

register the object centroid in compensation for deviation in spatial localization with a back tracking method. On the basis of registration, the feature point coordinates in the tracking window of neighboring frames are normalized into the coordinate system with the object centroid as the origin.

In this way, compared with the matching feature points of two unregistered tracking windows, the method of this paper can effectively reduce mismatching so as to lay a solid foundation for accurate estimation of scaling amplitude in the affine model of object; the kernel window width is updated according to scaling amplitude and verification from different perspectives. The algorithm mentioned in this paper is obviously better than the Mean Shift algorithm after being verified from different perspectives [21–25].

To evaluate the performance of different methods for semantic segmentation of aerial images, F_1 score and overall accuracy are used as evaluation criteria. The overall accuracy (OA) is defined in Equation (14). The OA is the ratio of the number of correctly classified pixels to the total number of pixels.

$$OA = \frac{TP + TN}{P + N}, \quad (38)$$

where $TP + TN$ represents the number of correctly classified pixels and $P + N$ represents the total number of pixels.

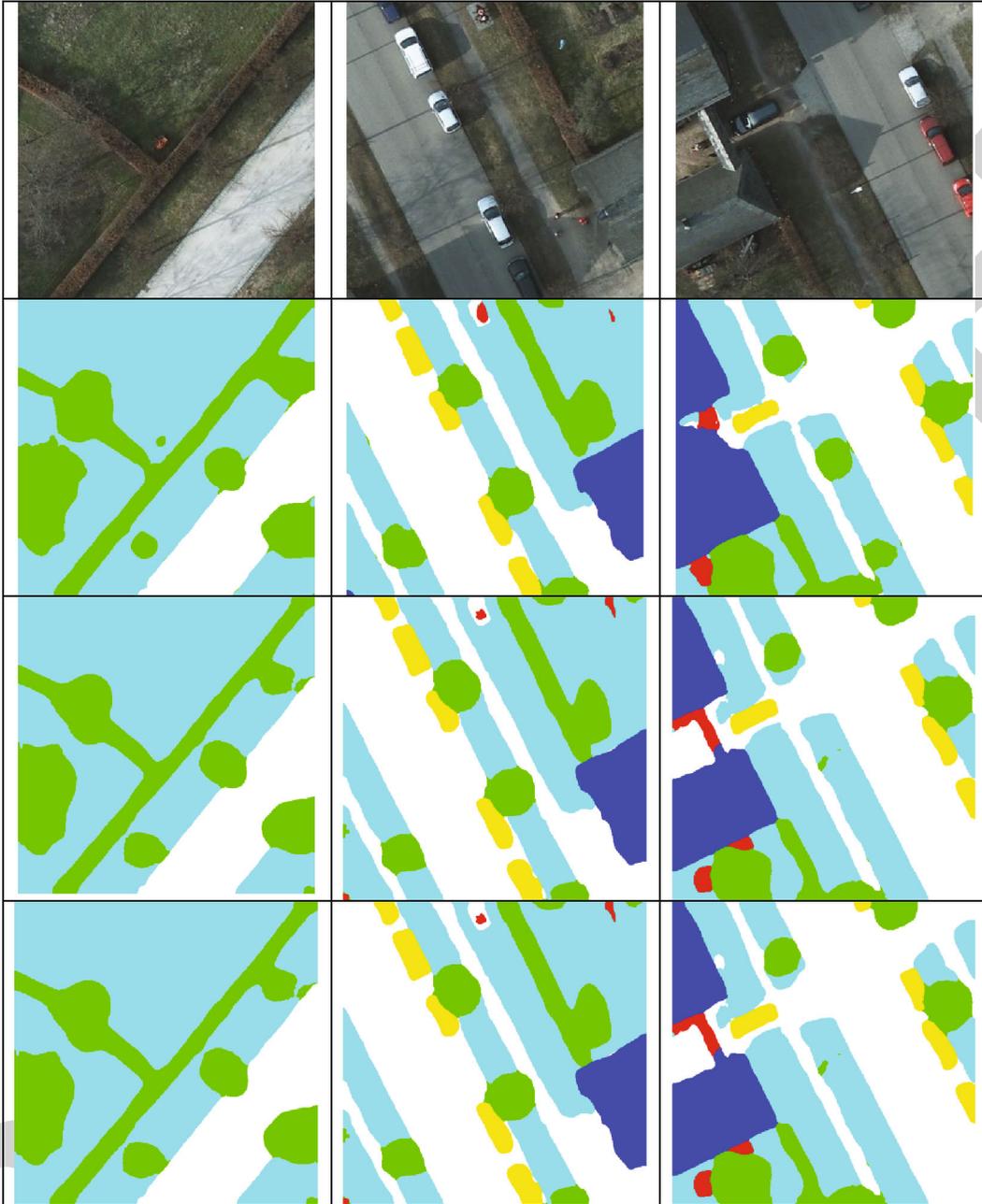


FIGURE 9: Example results on the Vaihingen dataset.

The accuracy of each category is evaluated using the F_1 score.

$$\begin{aligned}
 \text{Precision} &= \frac{TP}{TP + FP}, \\
 \text{Recall} &= \frac{TP}{TP + FN}, \\
 F_1 &= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}},
 \end{aligned} \tag{39}$$

where TP, FP, and FN are the numbers of true positives, false positives, and false negatives, respectively.

To verify the effectiveness of the proposed method, we perform comparisons against two state-of-the-art semantic segmentation networks, i.e., SegNet [26] and DeepLabv3+ [27], which are two most widely used models in semantic segmentation of aerial images.

The setting is the same as the cited paper [27]. Table 2 presents results on the ISPRS Vaihingen dataset, and we can see that the proposed algorithm significantly outperforms other methods on both F_1 score and overall accuracy. Compared to DeepLabv3+, the proposed method (DeepLabv3+*) increases the mean F_1 score and overall accuracy by 0.3 and 0.19%, respectively; in comparison with SegNet, increments on mean F_1 score and overall accuracy are 0.27% and 0.12%, respectively.

On the ISPRS Vaihingen dataset, the first line is the original image, the second line is the result of DeepLabv3puls, the third line is the combination of DeepLabv3plus and Mean Shift superpixel segmentation, and the last line is the BiSeNet with shift superpixel method.

As can be seen from Figure 9, the segmentation method of DeeplabV3+ and Mean-Shift fusion proposed in this paper realizes the separation of different plots, and the boundary line of the divided area is well matched with the boundary line of the real objects, while maintaining the integrity of the plot. In summary, the results of the DeeplabV3+ and Mean Shift based on superpixel segmentation methods proposed in this paper are significantly better than the current best-performing DeepLabv3+ methods.

8. Conclusion

Object tracking contains not only motion information of the object in the image but also rich information of 3D physical structure; therefore, it can be used to pin down the object motion and reflect other image information. Traffic video object tracking is an important method to analyze motion sequence image. Objected at the flaws of Mean Shift object tracking algorithm in complex road conditions, this paper has proposed an improved Mean Shift object tracking algorithm. Apart from such strengths as adaptive size of tracking framework and high real time, this algorithm can also adaptively adjust the stability of video image, remove background interference similar to object colors, and effectively track the object. The experiment has proved that the algorithm in this paper is effective. In addition, a model that combines the Mean Shift algorithm and deep convolutional neural network is proposed, which is beneficial to improve the semantic segmentation of aerial images. The experimental results show that our method can achieve higher segmentation accuracy with less running time compared with the existing advanced methods and fully proves its superiority.

Data Availability

The simulation experiment data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Science Foundation of China (Grant No. 61303029).

References

- [1] V. Rowghanian and K. Ansari-Asl, "Object tracking by mean shift and radial basis function neural networks," *Journal of Real-Time Image Processing*, vol. 15, no. 4, pp. 799–816, 2018.
- [2] A. Mondal, A. Ghosh, and S. Ghosh, "Scaled and oriented object tracking using ensemble of multilayer perceptrons," *Applied Soft Computing*, vol. 73, no. 12, pp. 1081–1094, 2018.
- [3] S. Gehly, B. A. Jones, and P. Axelrad, "Search-detect-track sensor allocation for geosynchronous space objects," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 6, pp. 2788–2808, 2018.
- [4] Y. Inoue, T. Ono, and K. Inoue, "Real-time frame-rate control for energy-efficient on-line object tracking," *Ieice Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. E101A, no. 12, pp. 2297–2307, 2018.
- [5] J. M. Gambi, M. L. Garcia del Pino, J. Gschwindl, and E. B. Weinmüller, "Post-Newtonian equations of motion for LEO debris objects and space-based acquisition, pointing and tracking laser systems," *Acta Astronautica*, vol. 141, pp. 132–142, 2017.
- [6] L. Dong, M. N. Satpute, W. Wu, and D.-Z. Du, "Two-phase multidocument summarization through content-attention-based subtopic detection," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 6, pp. 1379–1392, 2021.
- [7] D. Koniar, L. Hargas, and Z. Loncova, "Visual system-based object tracking using image segmentation for biomedical applications," *Electrical Engineering*, vol. 99, no. 4, pp. 1349–1366, 2017.
- [8] W. Chen, L. Cao, X. Chen, and K. Huang, "An equalized global graph model-based approach for multicamera object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 11, pp. 2367–2381, 2017.
- [9] M. Firouznia, K. Faez, H. Amindavar, J. Alikhani Koupaei, P. Pantano, and E. Bilotta, "Multi-step prediction method for robust object tracking," *Digital Signal Processing*, vol. 70, pp. 94–104, 2017.
- [10] S. Park, "Guidance law for standoff tracking of a moving object," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 11, pp. 2948–2955, 2017.
- [11] J. Yun, Y. Sun, C. Li et al., "Self-adjusting force/bit blending control based on quantitative factor-scale factor fuzzy-PID bit control," *Alexandria Engineering Journal*, vol. 61, no. 6, pp. 4389–4397, 2022.
- [12] S. Seol, E.-K. Lee, and W. Kim, "Indoor mobile object tracking using RFID," *Future Generation Computer Systems-The International Journal of Escience*, vol. 76, pp. 443–451, 2017.
- [13] S. M. Frank, L. Sun, L. Forster, P. U. Tse, and M. W. Greenlee, "Cross-modal attention effects in the vestibular cortex during attentive tracking of moving objects," *Journal of Neuroscience*, vol. 36, no. 50, pp. 12720–12728, 2016.
- [14] J. M. Pak, C. K. Ahn, and Y. H. Mo, "Maximum likelihood FIR filter for visual object tracking," *Neurocomputing*, vol. 216, pp. 543–553, 2016.
- [15] I. Elafi, M. Jedra, and N. Zahid, "Unsupervised Detection and tracking of moving objects for video surveillance applications," *Pattern Recognition Letters*, vol. 84, pp. 70–77, 2016.
- [16] W. Shu, K. Cai, and N. Xiong, "Research on strong agile response task scheduling optimization enhancement with optimal resource usage in green cloud computing," *Future Generation Computer Systems*, vol. 124, pp. 12–20, 2021.
- [17] A. Zea, F. Faion, M. Baum, and U. D. Hanebeck, "Level-set random hypersurface models for tracking nonconvex extended objects," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 6, pp. 2990–3007, 2016.

- [18] A. Tran and J. E. Hoffman, "Visual attention is required for multiple object tracking," *Journal of Experimental Psychology. Human Perception and Performance*, vol. 42, no. 12, pp. 2103–2114, 2016.
- [19] L. Lv and C. Yan, "The mean shift tracking algorithm with adaptive bandwidth based on affine transformation," in *The 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 1601–1606, Zhangjiajie, China, 2016.
- [20] J. Yang, J. Liu, R. Han, and J. Wu, "Generating and restoring private face images for internet of vehicles based on semantic features and adversarial examples," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2021.
- [21] Z. Chen, Y. Zhang, W. Chaozhong, and B. Ran, "Understanding individualization driving states via latent Dirichlet allocation model," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 2, pp. 41–53, 2019.
- [22] W. Wei, X. Fan, H. Song, X. Fan, and J. Yang, "Imperfect information dynamic Stackelberg game based resource allocation using hidden Markov for cloud computing," *IEEE Transactions on Services Computing*, vol. 11, no. 1, pp. 78–89, 2018.
- [23] K. Qiao, J. Zhang, W. Wei, R. Damasevicius, and M. Wozniak, "Adaptive independent subspace analysis (AISA) of brain magnetic resonance imaging (MRI) data," *IEEE Access*, vol. 7, no. 1, pp. 12252–12261, 2019.
- [24] H. Dou, Y. Qi, W. Wei, and H. Song, "A two-time-scale load balancing framework for minimizing electricity bills of internet data centers," *Personal and Ubiquitous Computing*, vol. 20, no. 5, pp. 681–693, 2016.
- [25] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [26] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [27] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, Munich, Germany, 2018.