Hindawi
Complexity
Volume 2020, Article ID 6635446, 14 pages
https://doi.org/10.1155/2020/6635446

Research Article

A Crowd Density Detection Algorithm for Tourist Attractions Based on Monitoring Video Dynamic Information Analysis

Lina Li

School of Business Administration, Shandong Management University, Jinan 250357, Shandong Province, China

Correspondence should be addressed to Lina Li; 14438120040108@sdmu.edu.cn

Received 12 November 2020; Revised 9 December 2020; Accepted 12 December 2020; Published 28 December 2020

Academic Editor: Wei Wang

In this paper, we analyze and calculate the crowd density in a tourist area utilizing video surveillance dynamic information analysis and divide the crowd counting and density estimation task into three stages. In this paper, novel scale perception module and inverse scale perception module are designed to further facilitate the mining of multiscale information by the counting model; the main function of the third stage is to generate the population distribution density map, which mainly consists of three columns of void convolution with different void rates and generates the final population distribution density map using the feature maps of different branch regressions. Also, the algorithm uses jump connections between the top convolution and the bottom void convolution layers to reduce the risk of network gradient disappearance and gradient explosion and optimizes the network parameters using an intermediate supervision strategy. The hierarchical density estimator uses a hierarchical strategy to mine semantic features and multiscale information in a coarse-to-fine manner, and this is used to solve the problem of scale variation and perspective distortion. Also, considering that the background noise affects the quality of the generated density map, the soft attention mechanism is integrated into the model to stretch the distance between the foreground and background to further improve the quality of the density map. Also, inspired by multitask learning, this paper embeds an auxiliary count classifier in the count model to perform the count classification auxiliary task and to increase the model's ability to express semantic information. Numerous experimental results demonstrate the effectiveness and feasibility of the proposed algorithm in solving the problems of scale variation and perspective distortion.

1. Introduction

With the development of science and technology to a new level, the quality of life of the people has been gradually improved [1]. At the same time, with the increase of the world population and the rise of the tertiary industry, the world economic development has ushered in a new wave [2]. The goal of the research on the topic of crowd counting and density estimation is to serve the daily needs of people, which is of great practical significance for crowd counting and density estimation in real scenarios [3]. Therefore, crowd counting and density estimation can be extended to the following three applications; in real-life scenarios, train stations, airports, large sports arenas, tourist attractions, and large shopping malls are crowded gathering places, and the number of people gathered in these places is usually very large [4]. The staff through the electronic camera equipment monitor such locations in real-time crowd dynamic information, and thus, the relevant technology is used to analyze the potential safety hazards, to nip the catastrophic event in the bud [5]. Crowd counting studies can also provide early warning of anomalous changes in the number of people at certain key locations (government sites, etc.). Crowd counting and density estimation techniques can also be used to gather intelligence to analyze and extrapolate events [6]. In the case, where annual holiday travel has become a normal part of the population, it is useful to analyze the flow of people at major tourist destinations in the country to manage road traffic and make adjustments to the overall tourism policy based on the travel preferences and interests of the population at each time of the year [7]. Also, the analysis of crowd dynamics in some tourist destinations can

enable managers to control the maximum number of people in the corresponding areas, thus reducing the likelihood of accidents such as crowding and trampling [8]. In addition to the aforementioned, the analysis of the number of people flow for different shelf positions in large shopping malls can be used to assess people's preferences for various types of goods and adjust the placement of goods and other arrangements based on the results of the analysis to maximize economic efficiency [9]. The results and analyses of crowd counting and density estimation research outputs can likewise provide reliable mathematical models for virtual-reality conversions, through which the evolution of various environments in virtual scenarios can be further enhanced to fully fit the real world [10]. These are often applied to large-scale computer platform-based action games, cinematic special effects, and the rehearsal of various real events (e.g., crowd evacuation in critical situations, public space design, and adjustment) [11].

Jing et al. proposed a semisupervised elastic net (SSEN) regression method to improve the "time-space intensive" assumption that a high enough video frame rate is required to capture the smoothness of the crowd on the time axis, by constructing a regular term based on the order information between an unlabeled sample and its neighbors in the time domain. These are all penalized by unreasonable predicted changes [12]. Coşar argues that the real world is difficult to meet the requirements when considering issues such as data bandwidth, storage space, and actual hardware devices; it proposes an alternative semisupervised regression framework that uses information about the underlying population distribution geometry to perform transfer learning, while relaxing the requirements based on the "time-space intensive." The requirement of the assumption not only makes the model smoother in exploring the structure of the inherent population distribution in time-space but also makes the population counting method more general and reliable [13]. Chebiyyam et al., in an accurate counting study considering dense populations, proposed to represent dense populations as irregular and inhomogeneous textures and to use Fourier analysis in local blocks, head detection, and SIFI-based points-of-interest three subschemas to achieve regional population counts and finally aggregate multisource (Fourier, head detection, and points of interest) counting information in the Markov random field to globally constrain the counts [14]. In addition, the authors introduced a dataset of image samples of size 50 frames containing 63,974 individual head marker information, called UCF_CC_50 [15]. Yang et al. introduced a new concept of cumulative properties according to which learning produces a count regression model when only sparse and unbalanced data are available [16]. Both results exhibit excellent performance results. Saon et al. were the first to propose a study to learn the linear mapping between local patch blocks and corresponding density maps to achieve population counting, which successfully avoids the shortcomings of detection-based and regression-based approaches by introducing a new method based on density estimation [17]. In a subsequent scientific study, Lu et al. found the task of learning the linear mapping between image-density maps to be very difficult and thus proposed a strategy for learning the nonlinear mapping between image patch blocks and density maps to solve the counting problem [18]. Basiri et al. clustered the feature spaces of image blocks into subspaces and learned their embedding by collecting the target population of these subspaces, with the difference that instead of continuing the previous learning strategy (learning the mapping between the overall population features and their corresponding density maps), the authors used the correspondence between the images and the density maps in the feature space to form a density estimation strategy based on subspace learning, and the relevant experiments proved that its counting accuracy was competitive with the mainstream scheme at that time, and the running speed in the testing phase was considerable [19]. In the same period, Yi et al. concluded that the existing population density estimation scheme is computationally expensive and has some drawbacks in processing the number of features and proposed a random forest embedded in the tree nodes as a regression model to combine more extensive and richer image features to realize the counting estimation and the counting strategy have achieved certain results [20].

Although the population counting algorithm based on the traditional approach has achieved some scientific results, its shortcomings are also obvious. It is not effective in dealing with high densities and other challenging problems such as significant scale variations, severe perspective distortions, and obscure population features. In this paper, we use dynamic information analysis to encode the shallow texture features of the same population distributed at different resolutions by using the image pyramid set as input and the feature encoder to enhance the multiscale representation of the counting model. The population distribution density map is generated by the density map regression module, which contains multiple cross-branching subnetworks, and these subnetworks take the convolution of voids with different void rates as the main element to expand the model's sensory field without increasing the weighting parameters.

2. Design of Crowd Density for Monitoring Video Dynamic Analysis

2.1. People Density Monitoring Video Dynamic Analysis Model. Crowd counting and density estimation research aim to design efficient intelligent algorithms to analyze image data and mine effective pedestrian feature information to achieve crowd counting. Earlier researchers used traditional machine learning algorithms to solve the counting task problem [21]. With the emergence and development of deep learning, the convolutional neural network-based approach shows great potential in the field of crowd counting, as shown in Figure 1.

As shown in Figure 1, the population counting study based on a convolutional neural network (CNN) is divided into four main parts: data preprocessing, model training, testing the model, and generating density maps and counting results, where M denotes the total number of

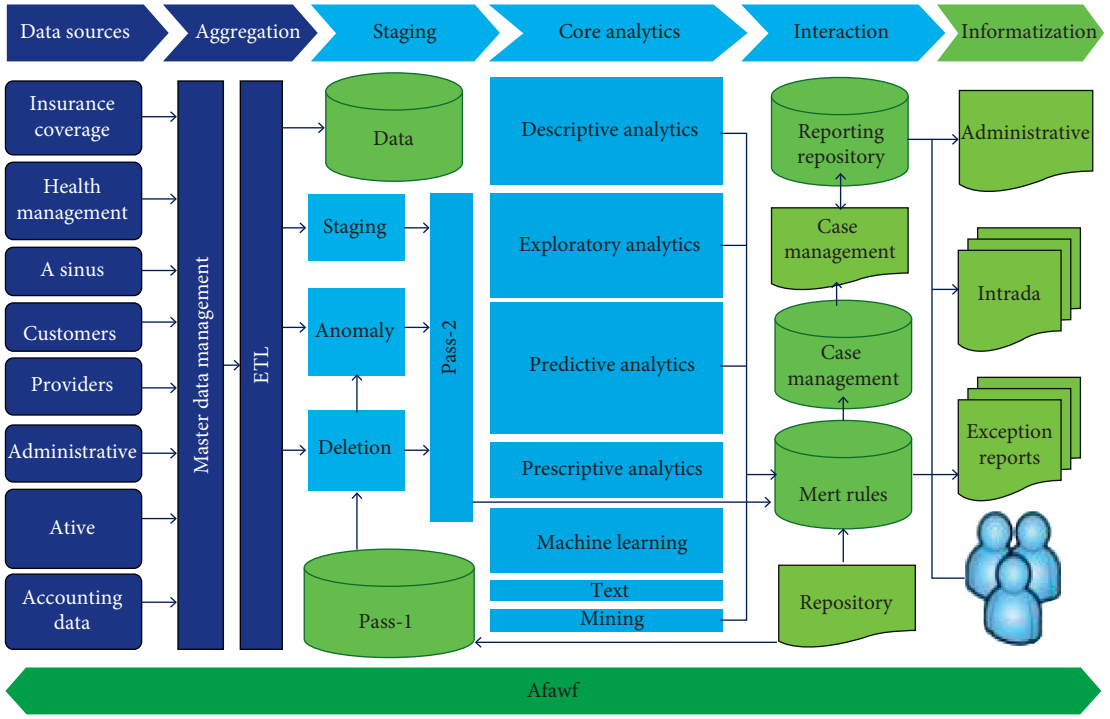

Figure 1: Crowd counting process architecture.

iterations. Parameter learning and optimization is an important part of model training [22]. The process from image or video data input to density map or count result output is the forward propagation process. The process of model parameter learning and optimization is the backward propagation process, in which the optimizer optimizes the weighting parameters iteratively to obtain the optimal counting model. Common optimizers are Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam), and Batch Gradient Descent (BGD), which are mainly used for gradient descent. The main purpose of the test model is to verify the validity of the algorithm [23].

A continuous image is assumed to be represented by $f(x, y)$, and to calculate the magnitude and direction of the gradient value at a point, the partial derivative of that point needs to be obtained. And since the digital image is discrete, it is possible to replace the first-order bias with a first-order differential [24]. The first-order differential function of the two-dimensional discrete function of the image in the horizontal direction is defined as follows:

$$f(x-1, y) - f(x, y) = x. \tag{1}$$

Accordingly, the first-order difference in the vertical direction is defined as

$$f(x, y-1) - f(x, y) = y. \tag{2}$$

The Roberts operator is simpler because it uses local differencing to find the edges of an image. It detects edges by using the gradient magnitude of a pixel to approximate the absolute value of the difference between two adjacent pixels in the diagonal direction [25]:

$$\text{grad}(x, y) = |f(x, y-1) - f(x, y)| + |f(x-1, y) - f(x, y)|. \tag{3}$$

The Roberts operator locates edges with high accuracy, but it is also sensitive to noise and is suitable for segmenting scenes with distinct edges and little noise [26]. The Roberts algorithm uses local differencing to obtain the edges of the image but has the disadvantage that the edges are not smooth in the resulting image after Robert's algorithm is used. The reason is that Robert's algorithm usually generates a wide impulse response in the region adjacent to the image edge, so refinement is often required before detection with Robert's algorithm, which in turn affects the accuracy of edge localization.

A texture feature is a global feature that describes the surface features of the corresponding image in the scene, but since the texture is only one characteristic of an object's surface, it does not reflect all the inherent features of the object [27]. We cannot expect to get a deeper picture of an image just by looking at its texture features. The texture feature is a statistic of the pixels about each pixel. Such regional features are superior in pattern matching, and there are no local deviations that fail to match. Texture features are usually used for retrieval between images with significant differences in texture information. As a statistical feature, it is usually rotationally invariant and has strong noise immunity. However, when the resolution of the image is low, the texture calculation may have a large deviation, or when the differences between the textures are small, it is difficult

for human visual perception to distinguish small differences between the textures, and it is no longer appropriate to use texture features to distinguish between images. The flow of people in a surveillance scene is very complex, and people of different densities will show different texture information in the image. Especially in the case of high-density people, due to severe blocking, if you still use the traditional pixel-based feature method to estimate the density of people, the error will be large. However, using a texture-based crowd density detection algorithm can significantly improve the accuracy in high-density crowd flow situations.

Assuming that the window to be tracked is W, the KLT algorithm is a method to calculate the squared grayscale difference between video frames for this window (sum of squared intensity differences) as the standard tracking algorithm. For grayscale images, in the KLT algorithm, we assume a feature window W which contains feature texture information. The image frame $I(x, y, t)$ at moment t and the position in I at moment t satisfy the following equation [28]:

$$I(x, y, t + \partial) = I(x - \Delta x, y - \Delta y, t), \tag{4}$$

that is, each pixel belonging to the next moment frame $I(x, y, t)$ can be obtained by translating $M = (x, y)$ units of the pixel of the response window in the previous frame $I(x, y, t)$. This is also described in the previous section; the KLT algorithm is designed to find the process for solving d:

$$M(X - d) = M(x - \Delta x, y - \Delta y, t). \tag{5}$$

In general, there are

$$G(x) = M(X - d) + n(X), \tag{6}$$

where $n(X)$ is the noise generated by the change in illumination in τ time. The squared integral of $n(X)$ over the entire window is thus obtained for the squared sum of the grayscale differences in the window image (SSD) [29]:

$$\varepsilon = \iint_v n(X)^2 w(X) \mathrm{d}X = \iint_v A(X - d) n(X)^2 w(X) \mathrm{d}X. \tag{7}$$

When d is much less than X, Taylor expansion of $A(X - d)$, rounding off the higher term, yields

$$M(X - d) = M(X, t). \tag{8}$$

Substitute (8) into (7) and simultaneously derive and make the result 0 for d on both the left and right sides of equation (7), which gives

$$\frac{\partial \varepsilon}{\partial d} = \iint_v A(X - d)\, n(X)^2 w(X) \times g w(X) \mathrm{d}X. \tag{9}$$

In this case, ε takes a very small value. Equation (9) can, in turn, be transformed into

$$\iint_v n(X)^2 w(X) \mathrm{d}X = \iint_v A(X - d)\, n(X)^2 w(X) \mathrm{d}X. \tag{10}$$

To reduce the effect of motion on the image processing quality, we model an affine transformation in the image sequence. The affine model requires the selection of feature points on the image and tracking them, and the affine transformation parameters of the whole image are obtained by fitting all the feature point offsets, thus establishing the affine relationship between images. Therefore, feature tracking and feature extraction become the key to solve the problem. The concept of feature extraction in image processing is a process of analyzing and processing the external world using the theoretical knowledge of computer vision. The edge, texture, color, and other features of an image are used to describe and process the image. We can select feature points by the grayscale variation rate of image edges, the order of texture primitives, and the geometric features of image edges. In general, the two basic problems of feature point tracking and extraction are how to select feature points suitable for tracking and how to achieve continuous tracking of feature points from frame to frame.

2.2. Classification of Density of People and System Flow. Pedestrian density detection is commonly used in public places such as shopping malls, stations, and large entertainment facilities. The crowd density is the average number of pedestrians in the scene area. For businesses or event managers, this value should not be too large or too small. If it is too big, it is not easy for people to evacuate and can lead to accidents. If it is too small, it will not achieve the desired effect. The crowd density can be divided into five levels, and organizers or managers can make timely adjustments based on the real-time levels. In real life, the definition of crowd density levels can be varied depending on the application scenario. In this paper, we adopt the definition proposed by Hemanth et al. [30], which divides the density of people into five levels from low to high, as shown in Table 1.

The crowd density monitoring system is a system that uses computer vision technology to analyze and process the image signals containing crowd scenes in real time. Crowd density detection mainly includes motion detection tracking and density estimation. The general processing steps are as follows: firstly, the movement behavior and features of the crowd are extracted as the movement foreground. Based on pixel statistics, texture analysis or individual characteristics of the motion foreground are used to classify the crowd. Then, the classification results are transmitted to the control system for processing.

The camera continuously picks up images of the flow of people in the target area scene and feeds them back to the video processing module. The captured video images are transformed into video sequences, which are analyzed by video image processing algorithms to obtain crowd density information. According to the density processed by the video processing module and the corresponding scene, the crowd density detection algorithm makes appropriate judgment and then carries out the next action automatically or manually. The most important thing in the whole block diagram is the video processing module, and the core of the module is the video image processing algorithm. The selection of the processing algorithm directly affects the accuracy of the observation results [31].

Table 1: Classification of people flow density levels.

Parameter	Time (ms)	Frames
Struck	12.67	5.7
KCF	12.6	64.0
TLD	60.4	14.8
CT	17.2	56.9
MST-ours	6.5	132.9
MIX-ours	64.2	23.4

Foreground extraction is a common tool used in crowd density detection studies. The aim is to extract the moving objects of interest to us from the background of the image sequence, ignoring other information that is of little help to us. For the images in the video, it is possible to use techniques that use motion detection to obtain motion objects or targets [32]. This is very important for postprocessing such as tracking moving objects and analyzing image anomaly events. Usually, the common foreground extraction methods used in video surveillance are optical flow method, interframe difference method, the probabilistic method, foreground modeling method, and so on.

Most of the existing pedestrian detection techniques use full-body feature extraction, so care needs to be taken to collect full-body photos of pedestrians in the original video extraction. However, due to the variability of the pedestrian target motion, most often there is mutual occlusion between pedestrians or between pedestrians and objects, and when the full body is not captured, the accuracy of the existing dataset will be greatly reduced during training. In this section, an existing method of taking full-body photographs of moving pedestrians is used [33]. The first step of the method is also to collect the raw image dataset, and then, the collected raw images are detected and identified using an existing network model of convolutional neural networks with high accuracy, where the dataset is trained using the existing VOC2012 dataset.

The convolutional neural network algorithm is used to frame out the pedestrians in the image and calculate their confidence level, the confidence interval of this system is set to be greater than 50%, but most of the framed pedestrian targets in the image have a confidence level of less than 50%, so there is severe occlusion between these targets and they are unidentifiable, but in reality, the targets are present, so the accuracy of this solution is not high [34–40]. However, when combined with the actual situation and the observation of multiple images, it is easy to see that there is a minimal occlusion in the pedestrian head region, so this paper uses head recognition to detect pedestrians.

Data annotation is a key factor in the accuracy of deep learning algorithms, and good data annotation can efficiently improve the accuracy of deep learning. At the same time, data annotation will also be the step with the largest amount of deep learning tasks, and the general data standard uses manual annotation, which requires a lot of wasted working time. Therefore, in this section, to reduce the manual workload in data annotation, a combination of manual and model annotation method is adopted. That is, the deep learning-based target detection model can identify some obvious features and annotate them by itself during the training process to reduce the manual task. This thesis is mainly to achieve the statistics of pedestrians, and in this study, deep learning is mainly to achieve the training and detection of the head target, so in the data, labeling is, the data that need to be labeled mainly selected head target candidate box location information as well as the information of the target. This section first selects 60% of the video sequences from all the collected raw videos as the training set, which is about 20 hours long, while this paper strictly follows the sequence of video shooting to ensure the independence of all training and testing. For the continuous videos, since the surging pedestrian target is generally not fast, all the frames within one second or a few seconds do not change much, so a sampling method is used for these images to delete some of the same or very small difference between the frames to reduce the workload and labeling time. Therefore, the video deletion process is done by combining the existing mature OpenCV image processing library, and finally, the 42,600 images in the video are retained as the original training images.

3. Experimental Design Analysis of Detection Algorithms

3.1. Design and Experiments of Crowd Density Detection Algorithm. In traditional deep learning, the feature network is typically VGG-Net, which detects correctly and with high accuracy when tested on Pascal VOC datasets. As a result, the VGG-Net-based network is one of the highest performing image feature extraction infrastructure network architectures for deep learning algorithms. However, in recent years, deep learning has become more demanding for network architectures and convolutional neural network algorithms have matured, e.g., ResNet and Inception-ResNet, which have achieved even better results than the VGG-Net base network for feature extraction [35]. Therefore, these methods will be the preferred algorithms for the deep learning network architecture used in the thesis for detecting head targets. From the research problem of this thesis, the thesis does headcount for surveillance videos, which requires high real-time performance, so there are certain requirements for the processing speed of the algorithm. The image frame rate of the surveillance video used in this thesis is 15, which means that the processing time for each image frame is required to be no more than 0.067 seconds. Therefore, the requirement of time in the assessment index for the application of deep learning algorithms is less than 0.067 seconds, and the algorithm that is as effective as possible in target detection is selected on the premise of meeting the time standard [36]. Reading a large number of references, it can be found that deep learning detection time and accuracy are often inversely proportional, i.e., the higher the number of network architecture layers and more input parameters in deep learning, the better the accuracy of target detection and the longer the time required, as shown in Figure 2.

After considering time and accuracy, the system adopts the network architecture of ResNet-101 + RFCN as the deep learning feature extraction network architecture for pedestrian

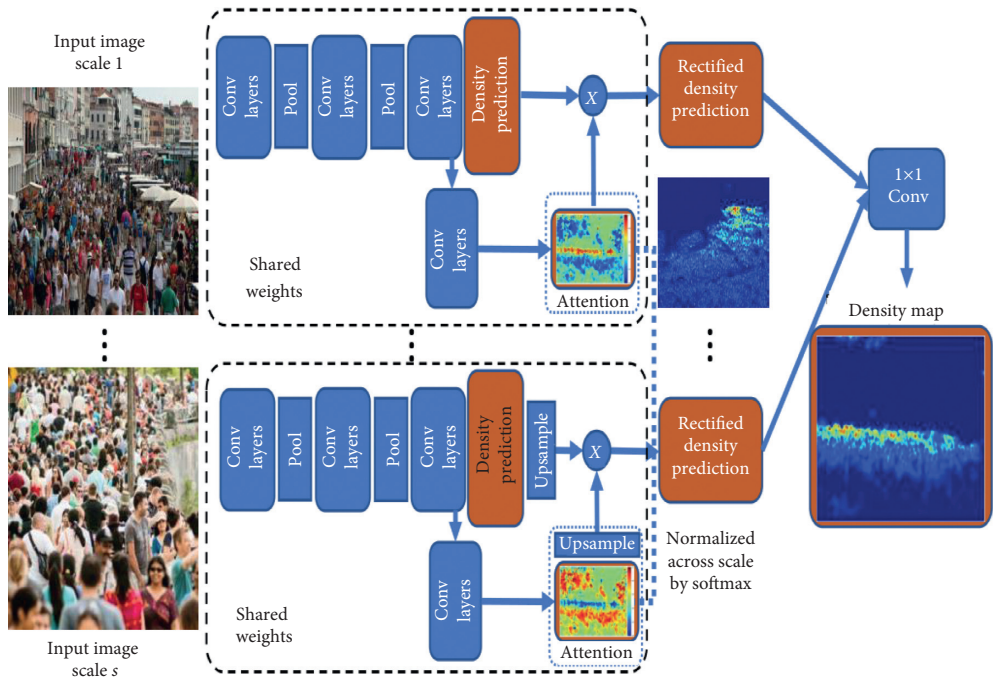

Figure 2: Experimental design framework for human flow density detection.

detection. In this section, the network architecture is explained in detail as well as experimentally tested and analyzed. For the pixel-based population density classification, the video is selected as the experimental object and processed by OpenCV, which is roughly divided into three categories: low density, very low density, and texture analysis population. The algorithm of crowd density classification based on pixel statistics is simple, but it is only applicable to the low-density population. Special attention is paid to the threshold selection; here, it is used to analyze multiple videos extracted from the low density, very low density, and texture analysis of the population obtained from the corresponding images, to reduce the human subjectivity, more persuasive. If pixel-based population density classification methods are used, high-density populations appear in the overlap between populations, although the algorithm is simple, the result has a relatively large error, so the texture analysis based on the population density classification method is used, and this method is a complex algorithm and the computational volume is relatively large. It generally first extracts the grayscale-based co-occurrence matrix and gets the eigenvalues on this basis and classifies the eigenvalues using the support vector machine.

The individual multigrained interaction network mainly considers the mutual constraints between the individual's trajectory information and action information to predict a more reasonable outcome. As shown in Figure 3, in the process of an individual changing the direction of motion, the overall motion direction of the trajectory tends to reflect the overall motion trend, and in turn, the motion changes also contain trajectory motion information, i.e., the two paths are mutually influenced.

Therefore, there is a need to model the interaction of individually predicted trajectory and motion information to jointly constrain the two to more realistic results. The entire individual multigrained information interaction network consists of three parts: the encoding network, the interaction network, and the decoding network. First, the inputs to the network are the trajectory change values and action change values predicted by the interindividual interaction network. The reason for using change values as inputs is that, on the one hand, they help to avoid a network memory environment and, on the other hand, the relationship between raw trajectory features and action features is difficult to model.

The input dimension of trajectory change value Δtra is 32, while the action change value Δact dimension is 128, so both need to be encoded before input to the bidirectional LSTM. The coding network adopts the Multilayer Perceptron (MLP), which on the one hand encodes the two channels into the same dimension and, on the other hand, is equivalent to encoding the two channels into the same implicit space before the information fusion interaction. Since a bidirectional LSTM can fuse information in two directions, this chapter adopts two time-step bidirectional LSTMs, one to input the coded trajectory information at one moment and the other to input the action change information at another moment. Then, after the forward-backward information interaction fusion, the fused information is output at each moment. The first moment corresponds to new information about the action after considering the trajectory information, and the other moment outputs new information about the trajectory after considering the information about the action. To predict the result, a different decoding network is also adopted to decode the output of the bidirectional LSTM network into the trajectory and action of the next frame. The decoding network also consists of multilayer perception machines, one decoding trajectory

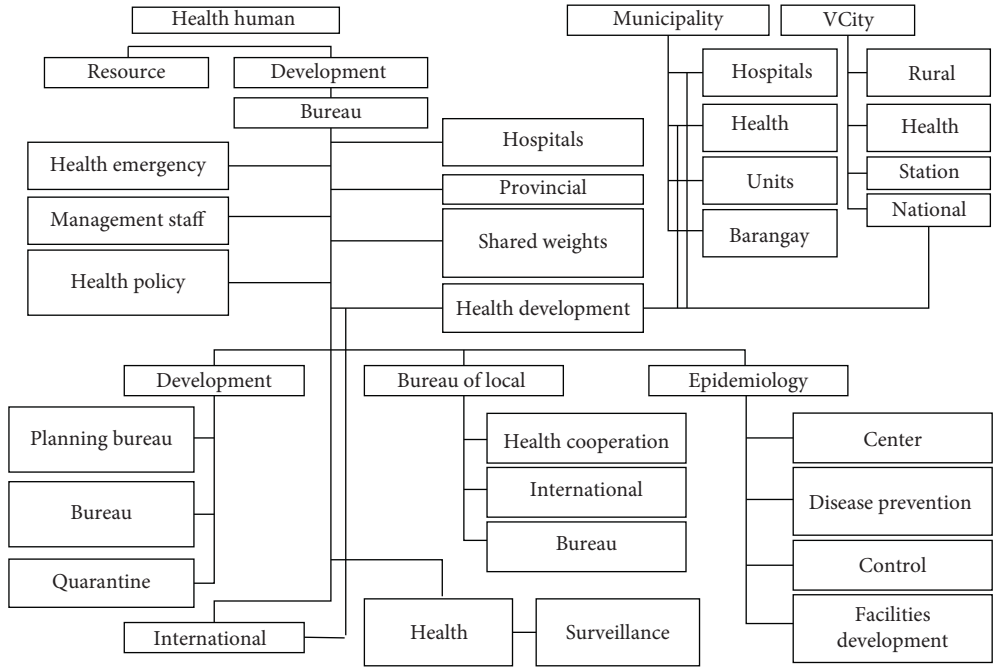

Figure 3: Multigrained information interaction within an individual.

information into the final predicted 2-dimensional Δ'tra and the other decoding motion information into the final predicted motion variation Δ'act. The above process completes the intraindividual interaction modeling of time t, i.e., the trajectory and motion prediction of time t. The prediction of time t is then recursively predicted for a long period as the input of the next time. Then, the prediction result at time t is used as input for the next time, and the long period prediction is recursively made.

3.2. Evaluation Index Design Analysis. For evaluating the performance of feature detection algorithms, the focus is generally on calculating the accuracy and recall of the detection results. The mAP is one of the most common performance evaluation metrics available, where AP is the integral area of the curve for accuracy and recall, m is the average of all APs, and m is usually taken as P. The higher the value of m, the higher the performance of the detection algorithm. The specific mAP calculation is as follows:

$$m\text{AP} = \frac{\sum_q^Q \text{ave}P(a)}{Q}, \tag{11}$$

where Q is the number of categories in all images of this design system and ave P (a) is the average accuracy (AP) of the a-th of all categories. For deep learning detection and training, a correspondingly high-performance computer is needed to match it, and the better the computational performance, the faster the training and testing will be. The computer used in this system has an Intel Core I7-8700K CPU and an NVIDIA GTX1080TI GPU.

To quantitatively validate our proposed concept of aggregation, we compare it with human movement. The database of crowd movements used in the experiment contains 413 video clips of 62 movement scenes. Each video clip contains 100 frames of images. To get close to the real-life results, we classified the crowd movements in the videos into three levels: high, medium, and low. We then proposed two criteria to evaluate the agreement between the agglomeration we described in the paper and the delineated actual crowd motion. The first criterion is the link between the crowd score and the degree of aggregation. We add these scores together as the crowd score for this video. The range of this score is [0, 20]. From the modified graph, we can see the magnitude of aggregation, crowd score, and speed. Using the KLT algorithm introduced in Section 3 , we extract the feature points of each video clip and can calculate the agglomeration degree and the speed of crowd movement in this video clip. Figure 4 depicts the relationship between crowd score and velocity for all video clips. Agglomeration is more strongly correlated with crowd score, and the results for agglomeration are consistent with our human perception.

The quality of the density map depends on the correct counting of the number of people in the image data. The first step is to save the pixel coordinates of the center position of the labeled head in the image employing annotation, the pixel coordinates of the center position of all heads in the image are filled with the number 1, and the other positions are filled with the number 0 to form a sparse matrix which is consistent with the size of the original image sample; the second step is to convert the corresponding sparse matrix into a density map by Gaussian filtering. The 2D density plot is a plot of the sum of all density values and the total number of people in the corresponding image. The 1 in the sparse matrix is formed by a convolutional operation with a fixed-size Gaussian kernel G to form the density map matrix, and the range of density values in the density map matrix corresponds to the theoretical head size, which is affected by

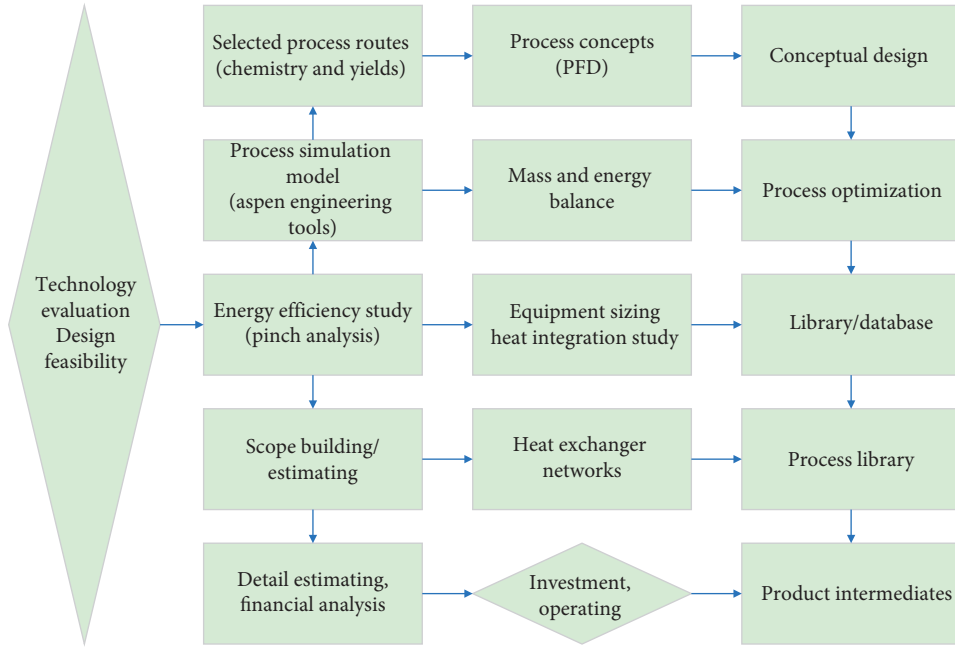

Figure 4: Evaluation of the metric design process.

the size of the Gaussian kernel G. Therefore, the counting performance of the population counting and density estimation model is closely related to the way the density map is generated. During previous studies, researchers have divided the methods into two types by whether the Gaussian kernel G size used is fixed or not: the fixed kernel density map generation method and the geometric adaptive kernel density map generation method.

The fixed Gaussian kernel lacks the adaptability to pedestrian size changes in the input image due to its fixed size, i.e., the estimation error for head size is large in densely populated scenes, while it has an advantage in sparsely populated scenes; however, the density map generated by the geometric adaptive kernel is exactly the opposite of the density map generated by the fixed kernel, and the average distance between the k nearest neighbors is small in densely populated scenes. And thus, small errors are in the head size estimation, whereas in sparse scenarios, the average distance between k's nearest neighbors is too large, which affects the quality of the density map generation and the final count results. Therefore, the way of generating density map is one of the key factors affecting the quality of density map, whether it is based on fixed Gaussian kernel (the initialized size of the fixed kernel also affects the quality of population density map) or based on the geometric adaptive kernel to generate population distribution density map; both have certain advantages and disadvantages.

4. Analysis of Results

4.1. Implementation and Performance Comparison of Different Detection Algorithms. The convolutional neural network is used as the base network for the training and testing of the already organized head trait dataset, where the convolutional neural networks base network connects the RPN with the fifth convolutional layer. The training algorithm is Stochastic Gradient Descent (SGD), the learning rate is set to 0.005, and the learning rate is adjusted to 0.1 when the training iteration reaches the 30000th round. ResNet, ResNet-50, and ResNet-101 are treated as network models for the convolutional neural network algorithm, where all the base networks are performed on the TensorFlow framework. The ResNet-50 and ResNet-101 network architectures are searched using their layer 4 convolutional neural network layer as the output layer. All the data features are from this layer, and the feature extraction box on its RPN network is formed without a direct 7×7 grid, using the clipping function of the TensorFlow framework, which automatically frames the clipping at the location of the feature, then zooms in to 14×14, and finally performs a maximum pooling operation to adjust the feature box to 7×7. The base network's Inception-ResNet uses its sixth convolutional neural network layer to detect the features of the target. Throughout the learning training, all the network structures are SGD algorithms. The values of the accuracy detection criteria mAP for the three network structures of ResNet-50, ResNet-101, and Inception-ResNet during deep learning training are shown in Figure 5.

In Figure 5, (a) shows the mAP curve of the detection index of the ResNet-50 network architecture training, (b) shows the mAP curve of the detection index of the ResNet-101 network architecture training, and (c) shows the mAP curve of the detection index of the Inception-ResNet network architecture training. From Figure 5, we can visually compare the feature detection performance of the three network architectures, in which the ResNet-50 network structure has a smooth detection performance in the late iterations, and the other two have large fluctuations;

however, the Inception-ResNet network structure has the best detection performance, and the ResNet-50 network structure has the worst detection performance. After the performance comparison is completed, this section also accurately measures the detection performance of the three network architectures, i.e., 500 images are selected as the original images to be detected and then tested on the already trained models and datasets, and the specific accuracy results of the three network architectures are shown in Figure 6.

From Figure 6, it can be seen that the accuracy metric *m*AP for the Inception network structure is 96.3%, the accuracy metric *m*AP for the ResNet-101 network structure is 94.7%, the accuracy metric *m*AP for the ResNet-50 network structure is 94.1%, and the algorithm has the lowest accuracy, with only 91.4% for *m*AP. At the same time, the Inception-ResNet network structure was found to be the slowest in terms of speed, about 3 times faster than the other three, with the ResNet-101, ResNet-50, and VGG-16 network structures being able to detect at a similar speed, all below the set detection rate (0.067 seconds).

However, accuracy is the key to the selection of the base network, provided that the time required for testing is met. Therefore, it can be seen from Figure 7 that the *m*AP of ResNet-101 is higher than the other two, so ResNet-101 should be chosen as the base network for this system. In the previous experiment, the paper chose ResNet-101 as the base network. Then, the deep learning algorithm needs to be selected for the ResNet-101 base network. The RFCN, SSD, and Faster-RCNN are tested experimentally in this paper. In the training experiment for the three network structures, the initial learning is 0.001 and the batch size is 4. The results of the three network structures are shown in Figure 7.

In Figure 7, the RFCN has the highest detection accuracy with an accuracy standard *m*AP of 95.7% when the detection process converges, and the SSD has the lowest accuracy with an *m*AP of only 90.3% when it converges, which is seriously below the minimum system design standard. In terms of detection speed, Faster-RCNN is the fastest, with an average of 0.057 s per sheet, and RFCN is the slowest, at 0.061 s. Although RFCN has the slowest processing rate, it is still within the allowable orientation (i.e., processing time ≤0.067 s). Through the above series of experimental simulations, we can clearly understand the specific performance parameters and corresponding indicators of all network structures. After analysis and comparison, the Res RFCN combination is selected as the network structure for deep learning. In the network structure selection experiment in this section, there is no cross-testing of the three base networks in the three network structures, but first, determine the selection of the base network, and then use the base network to test the network structure and get the final desired results. This effectively reduces the number of experimental tests based on achieving the objective. After determining the use of the Res RFCN, the specific performance of the network was also simulated and tested. For this test, the initial learning rate was 0.005 and the decay was 0.1, for 20,000 iterations. After the experiment, the accuracy standard *m*AP of its detection process was recorded and its change curve is shown in Figure 8.

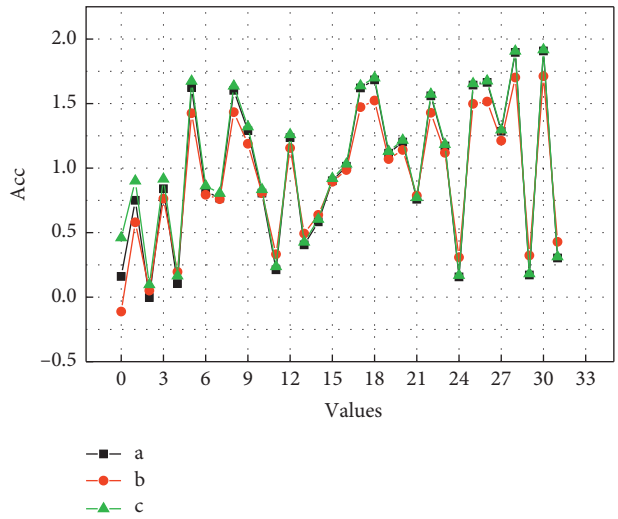

Figure 5: The *m*AP plots of the three base networks.

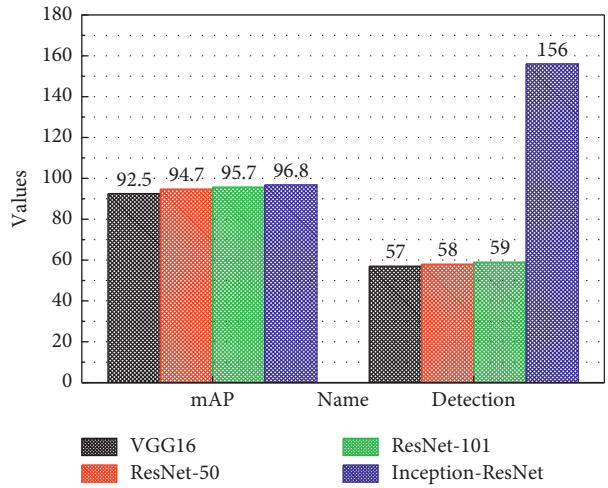

Figure 6: Test results of different base networks.

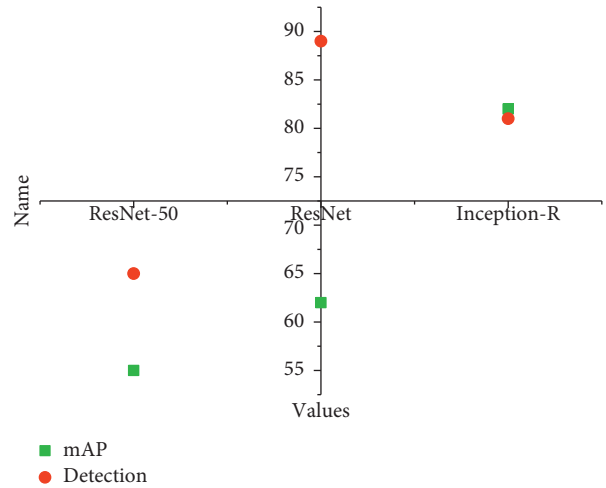

Figure 7: Test results for the three networks after combining ResNet-101.

As can be seen from Figure 8, when the accuracy index *m*AP reaches the maximum value of 0.957, the number of iterative rounds is around 26,500. The rate of detection at this point is 62 ms, which meets the minimum requirements of the system to adopt this result. Using the experimental structure described above, the detection results obtained from the real detection on the dataset can be concluded: the final method adopted in this section, ResNet-101 + RFCN, can achieve the detection of a pedestrian head target in the surveillance environment; at the same time, the accuracy and rate of the algorithm can slow down the system requirements. The design of the base network is the key part that affects the performance of the whole system because a series of feature extraction operations of deep learning is completed on the base network.

CNN utilizes an intermediate supervised strategy for deeper optimization of the counting model parameters. Therefore, to verify the reasonableness of the parameter settings in the intermediate supervision strategy and the impact of different parameter values on the counting performance, SACNN performs evaluation experiments on part A of the dataset. First, SACNN evaluates the performance of the parameters in the range of 10 to 0.00001, as shown in Figure 9. It should be noted that the parameter values are set to 0 for all the parameter evaluation experiments, and all other settings are the same. From the data in the table, SACNN has the best counting performance when the value of the parameter is 1 and the worst counting performance when the value is 0.00001. Also, to better show the trend of SACNN counting performance with the parameter values, this section converts the data in Figure 9 into a line graph to evaluate the trend change. As shown in Figure 9, the trend of SACNN's counting performance shows an increasing and then decreasing trend, with a significant change from $1e-5$ to $1e-4$, followed by a gentle increasing trend and an optimal performance when the value is 1, after which the counting performance shows a slowly decreasing trend. It can be concluded that when the value is small enough (0.00001), the counting performance of SACNN is mainly dominated by the loss for density estimation, and the loss has little effect; when the value is in the range of 0.0001~10, the loss starts to play a role, and the counting performance of SACNN is more stable to the sensitivity of the parameters.

4.2. Analysis of Evaluation Index Results. Figure 10 shows the visualization results of the density map generated by the SACNN algorithm, in which each column of images is represented as the original image, the true density map, and the generated density map, respectively, and the three image maps in each row belong to the same sample. Also, the "GT" and "ET" in the figure represent the true density and estimated density, respectively. Observing the comparison between the true density map and the estimated density map, it can be concluded that the quality of the generated density map is close to that of the true density map. By comparing the true density value and the estimated density value of the population in the test sample, it

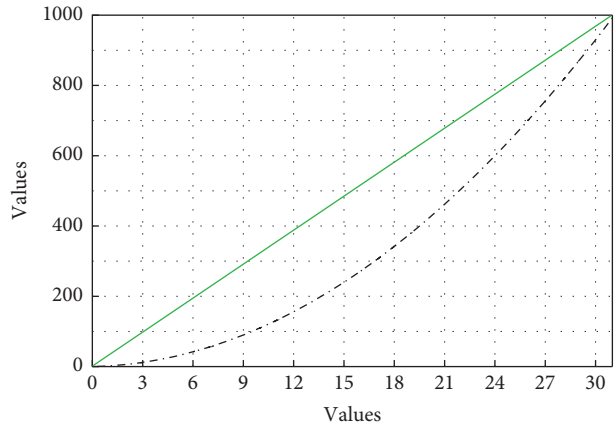

Figure 8: The *m*AP values of Res RFCN.

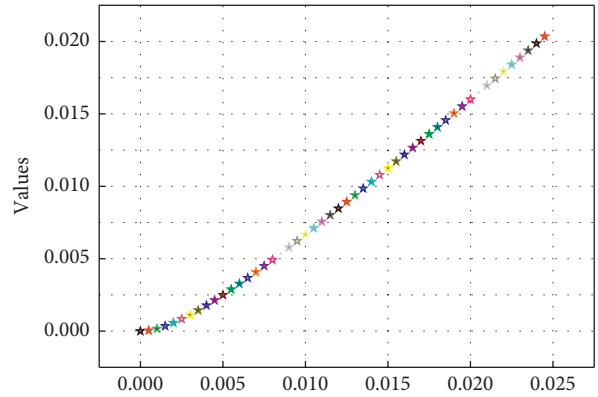

Figure 9: The *m*AP values of Res SACNN.

is found that the counting accuracy of the algorithm in this section is high. It also shows that sufficient multiscale information contributes to the performance improvement of the counting model.

Having done the simulation of the Kalman filter tracking algorithm on the thought motion, the Kalman filter algorithm has a good effect on position tracking. But the motion encountered in life is two-dimensional, so we continue to simulate the two-dimensional Kalman filter tracking algorithm. An ideal environment for the motion is then drawn using Monte Carlo simulation to compare the measured position, the true position, and the filtered tracking predicted position, as described in Figure 11.

In the experiments on the two-dimensional plane of Kalman's tracking algorithm, it can be found that under the same environment, Kalman's tracking algorithm has good tracking effect on the moving target, and from Figure 11, it can be seen that the tracking covariance of Kalman's algorithm will gradually become smaller with time, and its fluctuation will also become smaller and finally gradually approach 0. So according to the above figure, we can conclude that Kalman's filtering can achieve the following effect on the moving target tracking, but a certain amount of tracking time is required (the number of tracking steps in the above figure should preferably be set to 40 or more).

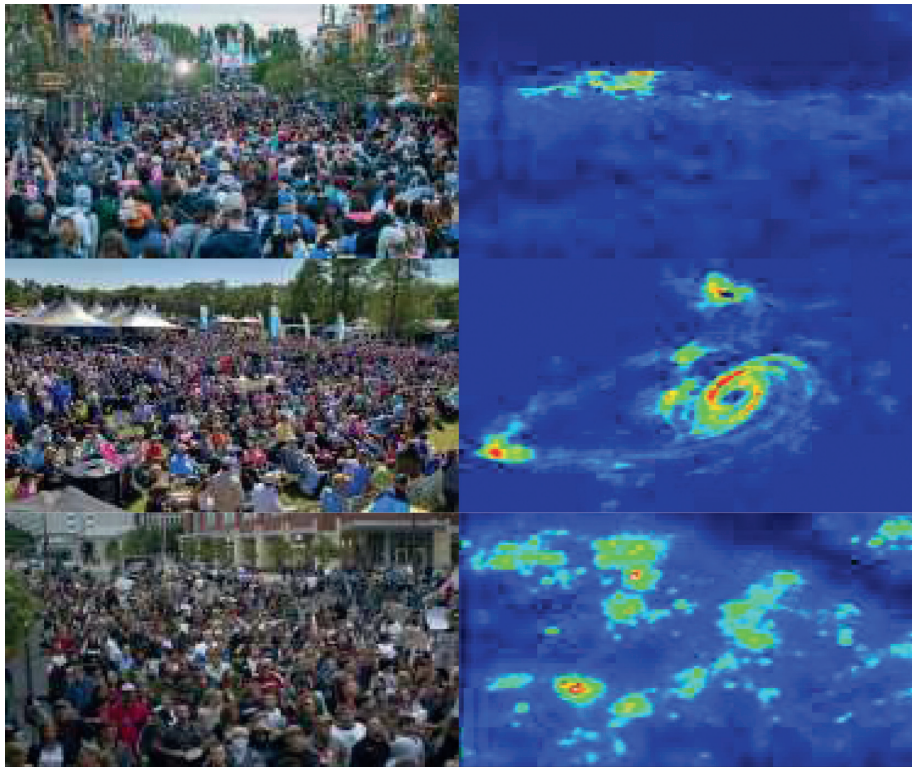

Figure 10: Visualization of the crowd density map.

Figure 11 shows the number of people in a complex environment, with the top left corner showing the number of people in real time and the top right corner showing the number of people being tracked. The first picture has two pedestrians just crossed the line and a pedestrian from the other direction to wipe the line, and the count shows that the number of people is 0 and the number of people being tracked is 0; the second picture shows the tracking process of the target in the virtual line box, and the count shows that the number of people is 0, the number of people being tracked is 2, and we can see that the tracking status is good; the third picture shows that all three pedestrians enter the virtual line box, and start tracking, the count is 0 and the number of people being tracked is 2. The fourth figure shows that three pedestrians have stepped out of the virtual wireframe to complete the count, and two more pedestrians have stepped into the virtual wireframe with a count of 3 and a count of 0. By counting the number of people in the video for the three scenarios, the detection results of the three scenarios are recorded as shown in Figure 12.

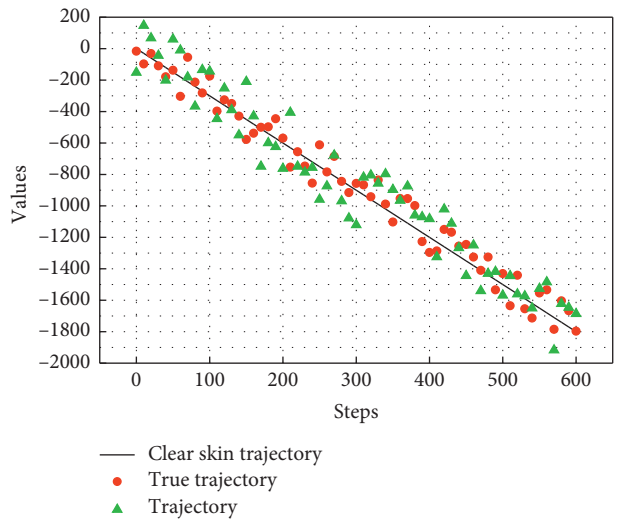

Figure 11: The number of people in a complex environment.

It also demonstrates the running flow of the software and the design of the upper computer interface; finally, the headcount experiments are completed for three different scenarios with simple to complex detection backgrounds. According to the test results, it is found that the people counting system designed in this paper can efficiently handle monitoring places with uncomplicated backgrounds and can meet the requirements of the universal situation. The properties of the model have been first described: the velocity is constant and the direction is the average direction of all individuals. Then, the concept of aggregation and the measurement method in the SDP model is introduced and analyzed. The three main properties of agglomeration are discussed briefly. Then, the steps of the population aggregation algorithm are listed. The accuracy of agglomeration is then verified by simulation for both single and mixed group motion. In a further study to demonstrate the SDP model, we compare the aggregation response with our human perceptions of the actual crowd situation and conclude that the aggregation response matches our actual perceptions. This was immediately followed by a revalidation of the broad applicability of the SDP model through movement in

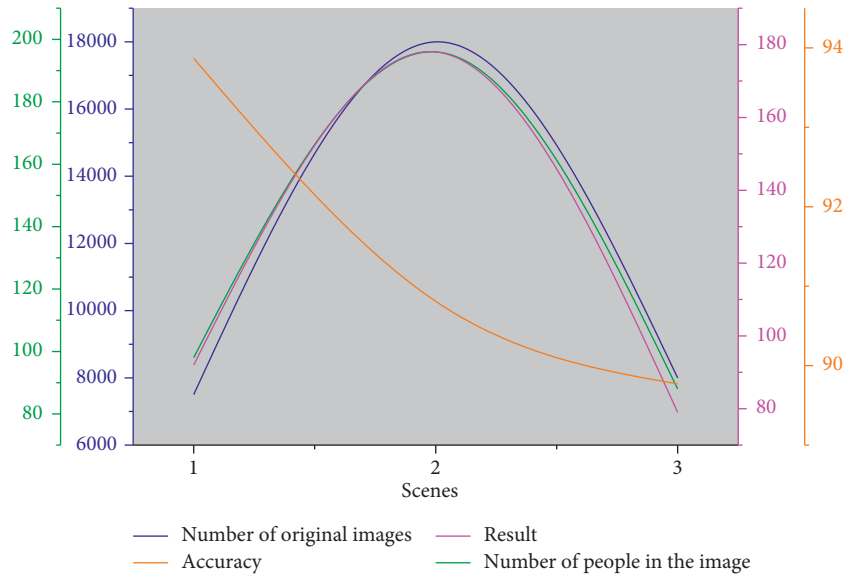

Figure 12: Detection results of the three scenes.

bacterial colonies. By combining the aggregation degree with the crowd density, we get the information of speed, direction, density, and aggregation degree of the crowd for better analysis of the crowd movement state. Finally, the range of agglomeration degree under different density levels is given.

5. Conclusion

This paper first introduces the significance of flow density detection and the status of research on flow density detection at home and abroad and then the knowledge of image preprocessing, which is the theoretical basis of crowd density detection. In the study of the feature extraction and tracking algorithm, the focus is on the feature extraction and tracking based on the surveillance video dynamic information analysis algorithm, which is compared with the traditional feature extraction method, and finally, the extraction of the feature points of the video image is completed using the surveillance video dynamic information analysis algorithm. The research in this section is to lay the foundation for aggregation response crowd density. In the study of crowd density detection algorithms, the mainstream algorithms based on pixel statistics and based on texture analysis are analyzed. A comparison of the two algorithms was completed experimentally, where the texture analysis-based algorithm used a grayscale co-occurrence matrix. After elaborating on the concept of aggregation after an in-depth analysis of the SDP (Self-Driven Particles) model, the SDP model was tested by simulation. Aggregation describes the degree to which an individual participates in a group movement. Aggregation enables managers to quickly identify the dominant individual in a video campaign. Agglomeration also qualitatively reflects the density of the crowd, with a high density of crowd being associated with a high degree of aggregation and vice versa. Finally, by combining crowd density and aggregation, the practicality and accuracy of aggregation are verified, and the value range of aggregation under five density levels is given. The combination of agglomeration and crowd density is the innovation of this paper, which also provides a reference for the future development of a complete system.

Data Availability

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

Conflicts of Interest

The author declares no known conflicts of financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Shandong Social Science Planning Project: Research on the Development of Urban Leisure in Shandong Province (no. 17CLYJ24).

References

[1] W. Wang, X. Yang, B. C. Ooi, D. Zhang, and Y. Zhuang, "Effective deep learning-based multi-modal retrieval," *The VLDB Journal*, vol. 25, no. 1, pp. 79–101, 2016.

[2] W. Tang, A. Panahi, H. Krim, and L. Dai, "Analysis dictionary learning based classification: structure for robustness," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 6035–6046, 2019.

[3] A. Almagbile, "Estimation of crowd density from UAVs images based on corner detection procedures and clustering analysis," *Geo-Spatial Information Science*, vol. 22, no. 1, pp. 23–34, 2019.

[4] J. M. Grant and P. J. Flynn, "Crowd scene understanding from video: a survey," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 13, no. 2, pp. 1–23, 2017.

[5] S. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy, "Trajectory-based surveillance analysis: a survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 7, pp. 1985–1997, 2018.

[6] M. Kumar and C. Bhatnagar, "Zero-stopping constraint-based hybrid tracking model for dynamic and high-dense crowd videos," *The Imaging Science Journal*, vol. 65, no. 2, pp. 75–86, 2017.

[7] M. S. Zitouni, A. Sluzek, and H. Bhaskar, "Towards understanding socio-cognitive behaviors of crowds from visual surveillance data," *Multimedia Tools and Applications*, vol. 79, no. 3-4, pp. 1781–1799, 2020.

[8] M. S. Kaiser, K. T. Lwin, M. Mahmud et al., "Advances in crowd analysis for urban applications through urban event detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 10, pp. 3092–3112, 2017.

[9] Y. Yoshimura, A. Krebs, and C. Ratti, "Noninvasive bluetooth monitoring of visitors' length of stay at the louvre," *IEEE Pervasive Computing*, vol. 16, no. 2, pp. 26–34, 2017.

[10] Y. Yang, J. Cao, X. Liu, and X. Liu, "Door-monitor: counting in-and-out visitors with COTS WiFi devices," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1704–1717, 2019.

[11] X. Kong, X. Song, F. Xia, H. Guo, J. Wang, and A. Tolba, "LoTAD: long-term traffic anomaly detection based on crowdsourced bus trajectory data," *World Wide Web*, vol. 21, no. 3, pp. 825–847, 2018.

[12] Y. Jing, B. Guo, Z. Wang, V. O. Li, J. C. Lam, and Z. Yu, "CrowdTracker: optimized urban moving object tracking using mobile crowd sensing," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3452–3463, 2017.

[13] S. Coşar, G. Donatiello, V. Bogorny, C. Garate, L. O. Alvares, and F. Brémond, "Toward abnormal trajectory and event detection in video surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 683–695, 2016.

[14] M. Chebiyyam, R. D. Reddy, D. P. Dogra, H. Bhaskar, and L. Mihaylova, "Motion anomaly detection and trajectory analysis in visual surveillance," *Multimedia Tools and Applications*, vol. 77, no. 13, pp. 16223–16248, 2018.

[15] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, "A survey of vision-based traffic monitoring of road intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2681–2698, 2016.

[16] X. Yang, L. Tang, C. Ren, Y. Chen, Z. Xie, and Q. Li, "Pedestrian network generation based on crowdsourced tracking data," *International Journal of Geographical Information Science*, vol. 34, no. 5, pp. 1051–1074, 2020.

[17] S. Saon, H. Hashim, M. A. Ahmadon, and S. Yamaguchi, "Cloud-based people counter," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 1, pp. 284–291, 2020.

[18] Y. Lu, Z. Zeng, H. Wu, G. G. Chua, and J. Zhang, "An intelligent system for taxi service: analysis, prediction and visualization," *AI Communications*, vol. 31, no. 1, pp. 33–46, 2018.

[19] A. Basiri, P. Amirian, A. Winstanley, and T. Moore, "Making tourist guidance systems more intelligent, adaptive and personalised using crowd sourced movement data," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 2, pp. 413–427, 2018.

[20] S. Yi, H. Li, and X. Wang, "Pedestrian behavior modeling from stationary crowds with applications to intelligent surveillance," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4354–4368, 2016.

[21] X. Hu, Y. Huang, X. Gao, L. Luo, and Q. Duan, "Squirrel-cage local binary pattern and its application in video anomaly detection," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 1007–1022, 2018.

[22] Y. A. Sahara and H. Zarei, "Sense of immersion in computer game using single and stereoscopic augmented reality," *International Journal of Human-Computer Interaction*, vol. 34, no. 2, pp. 187–194, 2018.

[23] A. A. García, I. G. Bobadilla, G. A. Figueroa, M. P. Ramírez, and J. M. Román, "Virtual reality training system for maintenance and operation of high-voltage overhead power lines," *Virtual Reality*, vol. 20, no. 1, pp. 27–40, 2016.

[24] B.-J. Sohn, D.-S. Park, and J. Choi, "Attitude confidence and user resistance for purchasing wearable devices on virtual reality: based on virtual reality headgears," *Journal of Intelligence and Information Systems*, vol. 22, no. 3, pp. 165–183, 2016.

[25] E. A. O'Connor and J. Domingo, "A practical guide, with the theoretical underpinnings, for creating effective virtual reality learning environments," *Journal of Educational Technology Systems*, vol. 45, no. 3, pp. 343–364, 2017.

[26] C. Gomez, S. Chessa, A. Fleury, G. Roussos, and D. Preuveneers, "Internet of things for enabling smart environments: a technology-centric perspective," *Journal of Ambient Intelligence and Smart Environments*, vol. 11, no. 1, pp. 23–43, 2019.

[27] G. Tieri, G. Morone, S. Paolucci, and M. Iosa, "Virtual reality in cognitive and motor rehabilitation: facts, fiction and fallacies," *Expert Review of Medical Devices*, vol. 15, no. 2, pp. 107–117, 2018.

[28] A. Hughes, A. Moreno, and T. H. My, "Which destination is smarter? Application of (SA) 6 frameworks to establish a ranking of smart tourist destinations," *International Journal of Information Systems and Tourism (IJIST)*, vol. 4, no. 1, pp. 19–28, 2019.

[29] X. Yang, L. Lin, P.-Y. Cheng, X. Yang, Y. Ren, and Y.-M. Huang, "Examining creativity through a virtual reality support system," *Educational Technology Research and Development*, vol. 66, no. 5, pp. 1231–1254, 2018.

[30] J. D. Hemanth, U. Kose, O. Deperlioglu, and V. H. C. de Albuquerque, "An augmented reality-supported mobile application for diagnosis of heart diseases," *The Journal of Supercomputing*, vol. 76, no. 2, pp. 1242–1267, 2020.

[31] J. Martín-Gutiérrez, C. E. Mora, B. Añorbe-Díaz, and A. González-Marrero, "Virtual technologies trends in education," *EURASIA Journal of Mathematics, Science and Technology Education*, vol. 13, no. 2, pp. 469–486, 2017.

[32] N. Chung, H. Lee, J.-Y. Kim, and C. Koo, "The role of augmented reality for experience-influenced environments: the case of cultural heritage tourism in Korea," *Journal of Travel Research*, vol. 57, no. 5, pp. 627–643, 2018.

[33] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 10, pp. 8065–8080, 2019.

[34] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: an overview," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6690–6709, 2019.

[35] Y. Ariji, M. Fukuda, Y. Kise et al., "Contrast-enhanced computed tomography image assessment of cervical lymph node metastasis in patients with oral cancer by using a deep learning system of artificial intelligence," *Oral Surgery, Oral*

Medicine, Oral Pathology and Oral Radiology, vol. 127, no. 5, pp. 458–463, 2019.

[36] Y. Chen, K. Zhu, L. Zhu, X. He, P. Ghamisi, and J. A. Benediktsson, "Automatic design of convolutional neural network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 7048–7066, 2019.

[37] M. T. Nguyen, P. Siritanawan, and K. Kotani, "Saliency detection in human crowd images of different density levels using attention mechanism," *Signal Processing: Image Communication*, vol. 88, Article ID 115976, 2020.

[38] H. Fradi, V. Eiselein, J.-L. Dugelay, I. Keller, and T. Sikora, "Spatio-temporal crowd density model in a human detection and tracking framework," *Signal Processing: Image Communication*, vol. 31, pp. 100–111, 2015.

[39] M. Kordestani, A. A. Safavi, and A. Sadrzadeh, "A new method to diagnose the type and location of disturbances in FARS power distribution system," in *Proceedings of the 2016 24th Iranian Conference on Electrical Engineering (ICEE)*, vol. 10, pp. 1871–1876, Shiraz, Iran, May 2016.

[40] H. Ullah, M. Uzair, M. Ullah, A. Khan, A. Ahmad, and W. Khan, "Density independent hydrodynamics model for crowd coherency detection," *Neurocomputing*, vol. 242, pp. 28–39, 2017.