Hindawi
Journal of Sensors
Volume 2022, Article ID 7247757, 19 pages
https://doi.org/10.1155/2022/7247757

Research Article

Research on Local Counting and Object Detection of Multiscale Crowds in Video Based on Time-Frequency Analysis

Guoyin Ren,[1] **Xiaoqi Lu**,[1,2] **and Yuhao Li**[1]

[1]*School of Mechanical Engineering, Inner Mongolia University of Science & Technology, Baotou 014010, China*
[2]*Inner Mongolia University of Technology, Hohhot 010051, China*

Correspondence should be addressed to Xiaoqi Lu; lan_tian1234@hotmail.com

Received 28 March 2022; Revised 1 July 2022; Accepted 15 July 2022; Published 12 August 2022

Academic Editor: Giorgio Pennazza

Objective. It has become a very difficult task for cameras to complete real-time crowd counting under congestion conditions. *Methods*. This paper proposes a DRC-ConvLSTM network, which combines a depth-aware model and depth-adaptive Gaussian kernel to extract the spatial-temporal features and depth-level matching of crowd depth space edge constraints in videos, and finally achieves satisfactory crowd density estimation results. The model is trained with weak supervision on a training set of point-labeled images. The design of the detector is to propose a deep adaptive perception network DRD-NET, which can better initialize the size and position of the head detection frame in the image with the help of density map and RGBD-adaptive perception network. *Results*. The results show that our method achieves the best performance in RGBD dense video crowd counting on five labeled sequence datasets; the MICC dataset, CrowdFlow dataset, FDST dataset, Mall dataset, and UCSD dataset were evaluated to verify its effectiveness. *Conclusion*. The experimental results show that the proposed DRD-NET model combined with DRC-ConvLSTM outperforms the existing video crowd counting ConvLSTM model, and the effectiveness of the parameters of each part of the model is further proved by ablation experiments.

1. Introduction

Video image processing to estimate the density distribution of crowd gathering areas is the key to crowd counting. Video crowd counting has important application value in the fields of traffic management, disaster prevention, and public administration [1–6]. When the number of crowded people reaches the safety limit, pedestrians are reminded to evacuate safely from the enclosed area to escape the danger, so estimating the number of crowds is helpful for early warning. This issue is particularly important for emergency escape behavior [7–10].

In addition to its role in the field of crowd counting, video image counting is also used in cell microscopic counting, vehicle counting, and other fields. Achieving crowd counting with computer vision techniques is an important challenge. At present, crowd counting mainly includes three types of methods: detection-based methods [11–13], regression-based methods [14, 15], and density map regression-based methods [16, 17]. However, in the face of occlusion, scale changes, background clutter, low resolution, and viewing angle changes, there are still some limitations in the realization of the above technologies.

Counting crowds in high dense aggregates is a challenging task. High-density crowds have serious occlusion problems, making traditional person detection methods ineffective [18]. Crowd video images capture the target from multiple angles, which results in the opposite sex of crowd scaling, so the target scale changes significantly. Each image includes hundreds or thousands of pedestrians, so annotating images of highly dense crowds is hard work. Large-scale crowd counting dataset labeling becomes even more infeasible. The application of crowd counting has potential practical value in many fields, such as dense crowd anomaly detection [19], crowd management in specific areas [20], and estimation of crowd size in public places [21]. These application domains share a common premise work: crowd feature extraction followed by crowd counting using object detection or regression models. The complex shape of the group makes it difficult to extract group features.

Recent advances in video image processing technology offer some possibilities for dense crowd counting. This field has received more and more attention from researchers in recent years, simultaneously providing empirical insights and data contributions for crowd counting [22].

Pedestrian detection can currently only handle crowds ranging from dozens to hundreds of people. However, this detection task is no longer applicable in the face of high-density crowd scenes with more than a few hundred people.

The biggest problem of object detection in crowded crowds is the low recall rate of pedestrian detection boxes [23, 24]. While regression-based crowd counting can estimate the total number of crowds, it cannot provide the spatial location information of the total crowds in the image. CNN-based methods outperform handcrafted feature extraction methods such as Scale Invariant Feature Transform (SIFT) [25], Histogram of Oriented Gradients (HOG) [26], and Local Binary Patterns (LBP) [27]. The multicolumn CNN network or scale aggregation module embeds more scale information in crowd image features and enhances the adaptability to scale changes. However, multiple convolution kernels and multiple feature maps need to store a lot of parameters and a lot of computing resources, so they are not suitable for real-time video processing.

At present, most of the detection-based crowd counting methods have difficulty in detecting small target groups, while the regression-based methods can only estimate the number of people in the entire image, but cannot locate pedestrians, so they cannot achieve different scales in the same image. Local crowd counts at different locations. In contrast, RGBD depth images are helpful for head size estimation, and high-quality density maps can be generated with the help of a Gaussian-distributed depth-adaptive kernel [28]. High-quality density maps can be used to train more robust regression networks, providing closer-to-realistic priors for crowd detection. One of the reasons why the previous detection methods cannot detect small heads is due to the lack of scale perceptron or the limitation of its own structure, and for those small/tiny heads, an effective multiscale perceptron should be designed. Fortunately, a multimodal image fusion perceptron of RGB images and RGBD depth is able to provide a prior for estimating head size, which helps to adaptively detect human heads at different scales. As shown in Figure 1, the recognition effect is better for crowd images with uniform scales, but for crowds that are too dense, it is still impossible to achieve head detection that covers the crowd at all scales.

Compared with still images, video images can provide more continuous features in time and space. The network structure of ConvLSTM can make full use of temporal information to improve the performance on this dataset [29]. We propose an improved ConvLSTM method combined with a deep fusion perceptron to estimate the number of dense crowds in videos and achieve head detection of crowded crowds in videos.

Contributions based on this study are as follows:

(i) Through the analysis of previous work, it is found that BiConvLSTM has the possibility of further improvement. The DRC-ConvLSTM network adds a two-way reverse conduction mechanism to BiConvLSTM, which further improves the accuracy of video dense crowd counting. DRC-ConvLSTM points out how to effectively enhance this spatio-temporal perception in crowd images. At the same time, the DRC-ConvLSTM network combined with a depth-adaptive Gaussian kernel can extract the depth space edge constraint feature of the image to better complete the crowd counting in the video

(ii) Previous work has rarely added RGBD depth features to the ConvLSTM model. Our DRC-CONVLTM model integrates depth features with visible light features to further improve the robustness of dense crowd counting under different lighting conditions. The depth image training is evaluated on the MICC dataset to verify its effectiveness

(iii) Previous work on local counting and detection of people with different scales is very little. In this paper, density-based feature point clustering can analyze and separate local groups with different scales in crowd scenes. By using improved DBSCAN density clustering, the local counting and detection of a variable density population are realized to understand the size of different populations

(a) (b)

Figure 1: Human head detection in images with large scale averages and large scale gaps: (a) human heads at same scales; (b) human heads at different scales.

2. Related Work

At present, there are few methods for local crowd counting and crowding calculation in public scenes. However, many countries have some grading standards for measuring the level of local crowd congestion for specific environments. For example, the transportation department has some grading standards to measure the standing density and crowding standards of passengers. The standards of passenger standing density are divided into two categories: comfortable and crowded. For example, in the United States, TCQSM provides a definition and reference of pedestrian level of service (LOS) for evaluating passenger congestion [30]. In China, passenger standing density can be divided into three different levels [31].

In the past, there are many literatures that use computer vision and image processing technology to complete the evaluation of video crowd crowding. For example, Alhadhira

et al. [32] proposed that crowd analysis using surveillance cameras can generate real-time crowd counting and give early warning of stampede results. However, no optimal balance strategy is given in terms of computational performance and accuracy. Chen et al. [33] implemented moving object detection with background, which can quickly estimate the number of pedestrians in a video. This method is only suitable for low-density crowds and will fail when computing still images. Therefore, researchers Ullah et al. [34] proposed a method that can handle high-density counting by computing optical flow fields to achieve cross-scene moving crowd counting, but this method is ineffective in still images. The attention network (ADCrowdNet) proposed by Yu and Zhang [35] provides local congestion crowd counts for density maps with the help of an attention map generator. Li et al. [36] proposed a RAZN model, which will localize highly ambiguous regions, improve their sharpness, and count them iteratively. Zhao et al. [37] established a crowd density estimation based on crowd classification criteria. The study found that the type of transportation that passengers choose has an important relationship with the changes in the number of crowded people. Shivapuja et al. [38] evaluated the influencing factors of the construction of intercity rail transit in Mumbai, India, by analyzing the changes in the population. Yin et al. [39] analyzed the impact of crowd congestion on the demand for public transport systems. This method analyzes and optimizes the passenger flow diversion of the Beijing rail transit. Sekasi and Martens [40] analyzed the passenger flow and divided the service level of the station platform into five categories to evaluate the traffic congestion of the light rail. Liang et al. [41] completed the service level division of Beijing rail transit corridors. Mahmoud et al. [42] predicted the conflict degree of vehicles turning right by evaluating the pedestrian congestion of right-turning at intersections and analyzing the correlation between the number of pedestrians and congestion. Gao et al. [43] estimated crowd numbers directly from features extracted from images. The congestion map obtained by this method has a low resolution and cannot help vehicle dispatchers to accurately locate the crowd.

Luckner et al. [44] proposed that the primary task of estimating the level of crowd congestion is to correctly estimate the number of crowds. This has been reported in the literature on the relationship between local population density estimates and crowd counts, such as by Khan et al. [45] where finding the population size of an entire region to provide hazard warnings is not reliable because the hazard may only occur locally. In most cases, even if the overall density is normal, localized crowding can be very dangerous. Therefore, estimating the population count in the local area is more in line with the actual situation.

More robust detection methods can compute and locate local population counts. In video surveillance, target detection works better in low-density situations, but often fails in high-density situations due to fewer pixels per capita, severe occlusion, and background clutter. Localizing and counting crowd objects in videos face many problems [46–48]. Since detection performance can be severely affected in overcrowded real-time scenes, detection-based methods are often outperformed by density map regression-based methods. The success of density map-based regression methods can be attributed to their ability to bypass explicit detection and map input images directly to scalar values [49–51]. However, although the method based on density regression can perceive the distribution of the crowd, it loses the ability to generate the individual localization of the crowd, so it is difficult to further study the dense crowd tracking and reidentification technology in surveillance.

Reference [52] has demonstrated that the performance of object detection counting can be further improved. Attempts have been made to apply more advanced detection frameworks to improve crowd counting performance. For example, Xue et al. [53] proposed an end-to-end person detector that can cope with crowded scenes. Most previous object detection methods do not handle small object crowd counts well.

Since there are few RGBD crowd counting datasets currently, there is not much work to complete crowd counting based on RGBD images [54, 55]. In these studies, depth information usually provides prior knowledge of head position for RGB image segmentation. Luo et al. [56] accomplish head detection improvement, which is directly related to the help of RGBD depth images. Wu [57] used the depth image collected by the Kinect sensor to complete the performance improvement of head detection. In [58], Liciotti et al. utilized RGBD images to complete head and shoulders detection and implement crowd counting. In this paper, we introduce a large-scale RGBD dataset for crowd counting, and we generate more accurate density maps and head detections.

Video stream data contains three main features: temporal features, spatial features, and periodic features. Many advanced methods combine CNN and LSTM to generate a ConvLSTM module. Kim et al. [59] used the ConvLSTM module to process short-term crowd flow data in adjacent regions. Chang and Luo [60] used bidirectional BiConvLSTM to process the real-time crowd data at the prediction point and extracted the periodic and spatiotemporal features of the video stream data. Chen et al. [61] proposed an end-to-end deep learning crowd flow prediction structure without data preprocessing and data feature extraction to predict the number of crowd flows. The above various improved ConvLSTM modules are used to extract spatiotemporal features of video streams, among which BiConvLSTM (bidirectional LSTM) has significantly improved the performance of extracting periodic features of video streams.

Different density groups in crowd scenes can be analyzed and separated by employing a modified DBSCAN clustering [62]. Density-based clustering methods rely on head center points to count local populations in order to understand population local sizes.

Our work improves method [63] in that we both train a detection network for crowd counting. While training fully supervised detectors with bounding box annotations, we only train weakly supervised detectors with point-level annotations [64]. Unlike our method, which only focuses on the number of people, our goal is to predict the number of people and generate appropriately sized detection boxes.

3. Methods

3.1. Gaussian Kernel Density Map

3.1.1. Kernel Density Estimation Method for Adaptive Bandwidth. If the bandwidth of the Gaussian kernel is not fixed, but varies depending on the location of the samples, a method of kernel density estimation called adaptive or variable bandwidth results. Due to the shooting distance of crowd images, there may be scale differences in the images. Therefore, in this paper, a variable bandwidth is used for kernel density estimation.

This part is used to explain the principle of adaptive bandwidth kernel density estimation method. The kernel density estimation method of adaptive bandwidth is obtained by modifying the bandwidth parameters on the basis of the fixed bandwidth kernel density function, and its form is shown in the following formula:

$$k(x) = \frac{1}{M}\sum_{j-1}^{M}\frac{1}{(wh_j)^n}K\left(\frac{x - x^{(j)}}{wh_j}\right), \tag{1}$$

$$K(x) = \frac{1}{\sqrt{(2\pi)^n|S|}}\exp\left(-\frac{1}{2}x^T S^{-1}x\right), \tag{2}$$

$$h_j = \left\{\frac{\left[\prod_{k=1}^{M} f\left(x^{(k)}\right)\right]^{1/M}}{f\left(x^{(j)}\right)}\right\}^{\alpha}. \tag{3}$$

Here $k(x)$ is the kernel density estimation function with bandwidth h_j, M is the number of individuals in the crowd, and each point j has a bandwidth h_j, so the bandwidth can be adaptive or variable. $K(x)$ is the kernel function, and the Gaussian kernel function is used here. $0 \le \alpha \le 1$ is a sensitivity factor; usually, α is 0.5. When $\alpha = 0$, the kernel density estimation of adaptive bandwidth becomes the kernel density estimation of fixed bandwidth. The kernel density estimate of the fixed bandwidth is the kernel density estimate $k(x)$ mentioned earlier. ω represents the parameter of the bandwidth.

3.1.2. Depth-Adaptive Gaussian Kernel Density Map. The adaptive Gaussian kernel can make the density map regression clearer and produce a regression density map that is closer to the true density map. The adaptive Gaussian kernel can be closer to the real head size, and the density map generated by regression can provide the deep network with the prior knowledge of head detection, which can guide the position and size of the detection frame.

The center point of the human head image label is calculated with a standard Gaussian kernel function and converted into a crowd density map. Assuming that $C = \{x_1, x_2, \cdots, x_n\}$ is a dataset in d-dimensional space, assuming that an image has n head instances, and the number of instances is n, then the distribution density of the data can be expressed as

$$\widehat{f}(x) = \frac{1}{nh^d}\sum_{i=1}^{n}K\left(\frac{x - x_i}{h}\right). \tag{4}$$

The multivariate Gaussian kernel function is given by

$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{2\pi^{d/2}}\cdot\exp\left(-\frac{\|x - x_i\|}{2h^2}\right). \tag{5}$$

Among them, $\|x - x_i\|$ is the Euclidean distance between x and x_i, h is the bandwidth, and the dimension is d. When the bandwidth h is equal to the head diameter h corresponding to the depth image, the estimated amount of data is n and can be expressed as

$$n = \frac{4}{h^{(d+4)}(d+2)}. \tag{6}$$

When the bandwidth h is equal to the depth image corresponding to the head diameter $h = R_{\text{deep}}$,

$$n = \frac{4}{R_{\text{deep}}{}^{(d+4)}(d+2)}. \tag{7}$$

For a multimodal dataset $X = \{x_1, x_2, \cdots, x_n\}$, x_n represents each instance; let its class label set $F = \{c1, c2, \cdots, cf\}$, where the class c_i (the number of instances in f) is N_{ci}; then, the density of instance x_i with respect to category c_i is calculated as

$$f_{ci}(x_i) = \frac{1}{N_{ci} - 1}\sum_{i\neq j, i=1}^{n}\frac{1}{R_{\text{deep}}^{d}}\cdot L(x_i)\cdot K\left(\frac{x_i - x_j}{h}\right), L\left(x_j\right) = c_j. \tag{8}$$

Among them, $L(x_i)$ and $L(x_j)$ represent the labels of instances x_i and x_j, respectively.

3.2. Density Map-Guided Detection. Detection-based models such as RetinaNet cannot detect these small/tiny heads because the detection subnet cannot adaptively adjust the depth perceptron of these heads. However, our network benefits from adapting the Gaussian kernel density map with depth. The size of the density map header is related to how many pixels the Gaussian kernel occupies in the density map. Therefore, the decoding layer of RGBD based on the learning feedback of our network detects heads of different scales to guide the detection of small heads. The RGBD map is downsampled to the same size as the density map. The RGBD pixel values for each head in M_l are used for reinforcement learning of our estimated density map. Specifically, for a given training head RGBD depth size, it is assumed that the size of the head to be detected is the size of the labeled rectangle. Then, we generate a head frame feature matrix M_l through training and further fuse M_l with the density map function $D^A(x)$ to generate an RGBD-adaptive Gaussian kernel constraint density map:

$$D_l^A(x) = D^A(x) \otimes M_l. \tag{9}$$

Here, $\otimes$ represents feature fusion, $D_l{}^A(x)$ is the density map constrained by the RGBD-adaptive Gaussian kernel, M_l is the head box feature matrix, and $D^A(x)$ is the adaptive Gaussian density map function.

3.3. Gaussian Kernel Density Map

3.3.1. Kernel Density Estimation Method for Adaptive Bandwidth. RGBD multilayer perceptron can solve the problem of adaptive perception of head size changes in the same scene. The RGBD multilayer perceptron is composed of visible light image and RGBD dual-modal image feature fusion network MLP-CNN. The network fuses the multimodal features of visible light images and RGBD images by importing them through the middle layer of the model.

This paper proposes an effective fusion of RGB features and RGBD depth features to improve the accuracy of object recognition. Taking both the RGB image and the RGBD depth image as the input original data vector and connecting them, the input data can be expressed as $\{x_{r1}, x_{r2}, \cdots, x_{rn}; x_{d1}, x_{d2}, \cdots, x_{dn}\}$, where $\{x_{r1}, x_{r2}, \cdots, x_{rn}\}$ and $\{x_{d1}, x_{d2}, \cdots, x_{dn}\}$ denote RGB and RGBD depth image vectors, respectively. Then, the parameter matrix A corresponding to the input data can be expressed as

$$A = \begin{bmatrix} A_{11}, A_{12} \cdots A_{1r_n} & A_{1(r_n+1)}, \cdots, A_{1(r_n+r_d)} \\ A_{21}, A_{22} \cdots A_{2r_n} & A_{2(r_n+1)}, \cdots, A_{2(r_n+r_d)} \\ \vdots & \vdots \\ A_{k1}, A_{k2} \cdots A_{kr_n} & A_{k(r_n+1)}, \cdots, A_{k(r_n+r_d)} \end{bmatrix}. \tag{10}$$

The first half of A is

$$A_{\mathrm{RGB}} = \begin{bmatrix} A_{11}, A_{12} \cdots A_{1r_n} \\ A_{21}, A_{22} \cdots A_{2r_n} \\ \vdots \\ A_{k1}, A_{k2} \cdots A_{kr_n} \end{bmatrix}. \tag{11}$$

A_{RGB} is the parameter corresponding to the RGB image vector. The second half is

$$A_{\mathrm{Depth}} = \begin{bmatrix} A_{1(r_n+1)}, \cdots, A_{1(r_n+r_d)} \\ A_{2(r_n+1)}, \cdots, A_{2(r_n+r_d)} \\ \vdots \\ A_{k(r_n+1)}, \cdots, A_{k(r_n+r_d)} \end{bmatrix}. \tag{12}$$

A_{Depth} is the parameter corresponding to the RGBD depth image vector, and k represents all possible class labels.

$$\lambda_{\mathrm{RGB}} = \begin{bmatrix} \lambda_{r1} & & & \\ & \lambda_{r2} & & \\ & & \ddots & \\ & & & \lambda_{rk} \end{bmatrix}, \tag{13}$$

$$\lambda_{\mathrm{Depth}} = \begin{bmatrix} \lambda_{d1} & & & \\ & \lambda_{d2} & & \\ & & \ddots & \\ & & & \lambda_{dk} \end{bmatrix}. \tag{14}$$

The overall cost function is shown in the following equation.

$$J_{\mathrm{sparse}} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{w,b}(x^i) - x^i\right\|^2\right) + \frac{1}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l^r}\sum_{j=1}^{s_{l+1}^r}\left(\lambda_{\mathrm{RGB}}\left(A_{\mathrm{RGB}}^l\right)_{ij}\right)^2 + \frac{1}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l^d}\sum_{j=1}^{s_{l+1}^d}\left(\lambda_{\mathrm{RGB}}\left(A_{\mathrm{Depth}}^l\right)_{ij}\right)^2 + \beta P(x). \tag{15}$$

The weight decay parameter λ_{RGB} corresponding to the RGB feature of the object is initialized to a smaller value, reducing its penalty and extracting more RGB features. λ_{RGBD} is the weight decay parameter corresponding to the depth feature, and λ_{depth} initializes a larger value to increase its penalty and extract fewer depth features.

3.3.2. Depth Perception Network. For the depth branch, we train a NiN variant and fit the depth data. NiN consists of multiple modules, each of which consists of a convolutional layer and multiple 1×1 convolution kernels whose kernels capture spatial information. This module is equivalent to a multilayer perceptron. For classification, a global average pooling layer produces a score for each class. We follow and discard global average pooling, resulting in an FCN that predicts scores for each pixel and class.

RGBD network: as shown in Figure 2, we identified differences in fusion. First, the RGB and depth inputs can be directly concatenated, and we call this model early fusion. In theory, a multimodal CNN with the aforementioned early fusion is more expressive than a midlevel fusion, which can exploit the correlation between early low-level CNN features. However, the better the fusion performance, the higher the required training cost. The benefit of late fusion is that most of the network weights can be reused directly without adjusting the network weights based on additional input cues. Unfortunately, it does not allow the network to learn about such high-level interdependencies among individual input modalities, and ultimately, only results at the classification level are fused.

3.4. Design of Real-Time Analysis Model for Dense Crowd

3.4.1. Real-Time Crowd Prediction Model. The model consists of three parts, and the specific process is as follows:

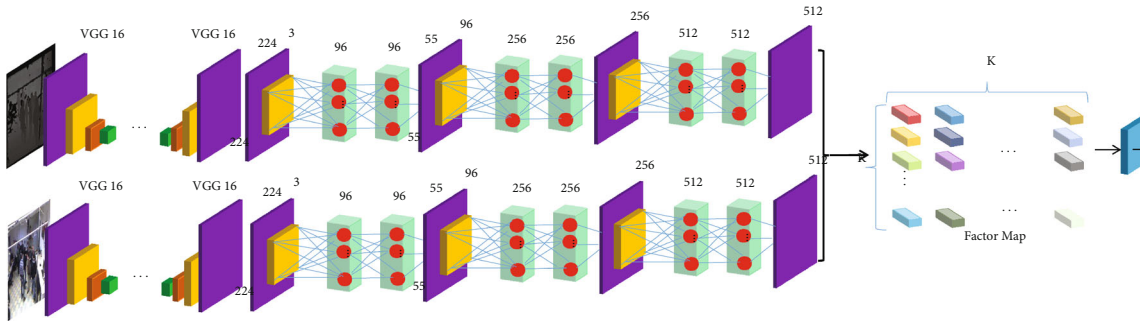

Figure 2: Structure diagram of the depth perception network.

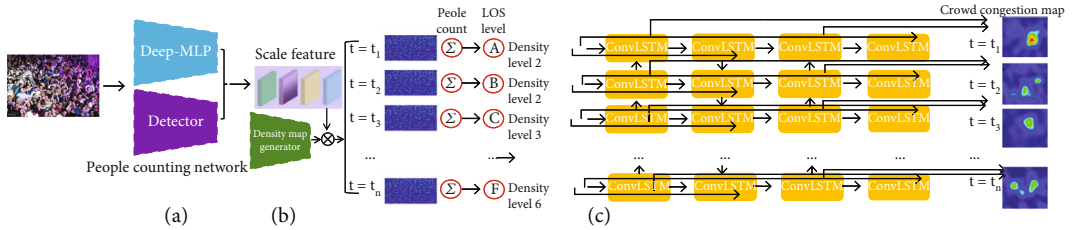

Figure 3: The network structure diagram proposed in this paper: (a) MLP-CNN; (b) predicting density map network; (c) DRC-ConvLSTM.

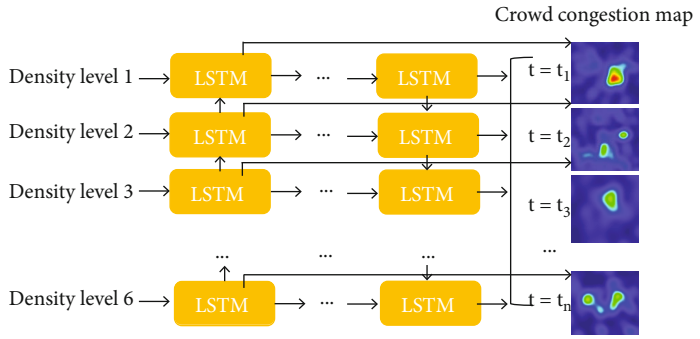

Figure 4: BIConvLSTM model network structure diagram.

(1) Each input image is first processed by (a) MLP-CNN. (2) The extracted features are fed to (b) for obtaining representations with spatial context information from different depths of the network and predicting density maps. (3) The spatial context representation is sent to (c) DRC-ConvLSTM to detect the crowd, as shown in Figure 3.

3.4.2. From ConvLSTM to Bidirectional ConvLSTM. ConvLSTM replaces matrix multiplication with a convolution operation for each gate in the LSTM unit. In this way, it captures the underlying spatial features by performing convolution operations in multidimensional data. The main difference between ConvLSTM and LSTM is the input dimension. Since LSTM input data is one-dimensional, it is not suitable for spatial sequence data such as video, satellite, and radar image datasets.

Inspired by the literature [59–61], the ConvLSTM model is extended to a bidirectional BIConvLSTM model, which can access bidirectional long-range information. Figure 4 depicts the bidirectional ConvLSTM model for crowd counting. Its inputs are the same as those in the ConvLSTM model. It works by computing the forward hidden sequence $\overrightarrow{H}$ and the backward hidden sequence $\overleftarrow{H}$ and updating the output layer by iterating backwards from $t = t$ to $t = 1$ and forwards from $t = 1$ to $t = t$ to compute the output sequence.

If we denote the state update function in (2) as H_t, $C_t = \text{ConvLSTM}(X_t, H_{t-1}, C_{t-1})$, the formula of the bidirectional ConvLSTM is shown in the following equation:

$$\begin{aligned} \overrightarrow{H}^1_t, \overrightarrow{C}^1_t &= \text{ConvLSTM}\left(X_t, \overrightarrow{H}^1_{t-1}, \overrightarrow{C}^1_{t-1}\right), \\ \overleftarrow{H}^1_t, \overleftarrow{C}^1_t &= \text{ConvLSTM}\left(X_t, \overleftarrow{H}^1_{t-1}, \overleftarrow{C}^1_{t-1}\right), \end{aligned} \tag{16}$$

where Y_t is the timestamp, and Chang and Luo [60] found that bidirectional ConvLSTM consistently outperformed unidirectional ConvLSTM in crowd counting.

Inspired by the above methods, the BIConvLSTM model can be further extended to DRC-ConvLSTM, which can

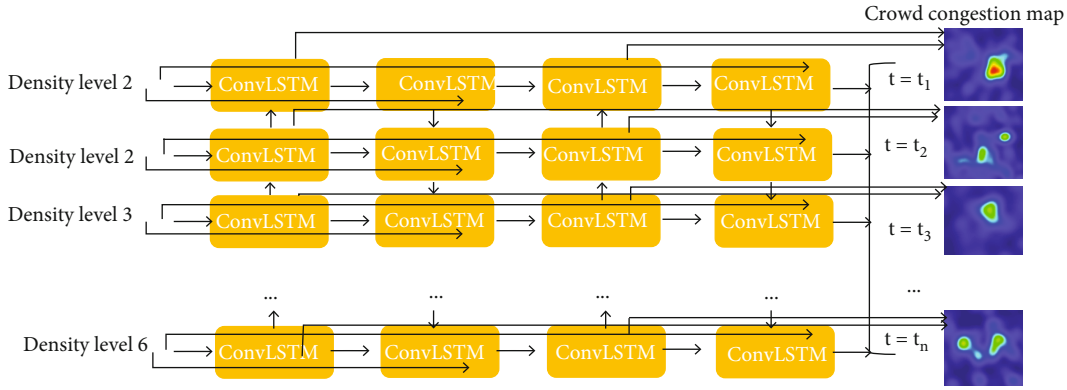

Figure 5: DRC-ConvLSTM model network structure diagram.

access two sets of parallel bidirectional long-range information. Figure 5 depicts DRC-ConvLSTM for crowd counting. Its inputs and outputs are the same as those in the ConvLSTM model. Its working principle is to calculate the forward hidden sequence $\vec{H}$ and the backward hidden sequence $\overleftarrow{H}$ twice by means of feature stacking and by iterating backward from $t = t$ to $t = 1$ and iterating forward from $t = 1$ to $t = t$ and then updating the output layer to calculate the output sequence; using the double-layer reverse ConvLSTM feature flow is an attempt at feature augmentation theory.

$$\begin{aligned}
\vec{H}^1_t, \vec{C}^1_t &= \text{ConvLSTM}\left(X_t, \vec{H}^1_{t-1}, \vec{C}^1_{t-1}\right),\\
\overleftarrow{H}^1_t, \overleftarrow{C}^1_t &= \text{ConvLSTM}\left(X_t, \overleftarrow{H}^1_{t-1}, \overleftarrow{C}^1_{t-1}\right),\\
\vec{H}^2_t, \vec{C}^2_t &= \text{ConvLSTM}\left(X_t, \vec{H}^2_{t-1}, \vec{C}^2_{t-1}\right),\\
\overleftarrow{H}^2_t, \overleftarrow{C}^2_t &= \text{ConvLSTM}\left(X_t, \overleftarrow{H}^2_{t+1}, \overleftarrow{C}^2_{t+1}\right).
\end{aligned} \tag{17}$$

3.5. DBSCAN Algorithm. There are two important parameters Eps and MinPts in the DBSCAN algorithm. Eps is the field radius when defining density, and MinPts is the threshold when defining core points. The variables Eps and MinPts are globally unique. Therefore, when the data distribution is uneven or the parameters are not selected properly, the quality of the traditional DBSCAN clustering algorithm is poor. The relevant definitions of the algorithm are as follows:

$$r = \{\text{dist}(p, q) \le R\}. \tag{18}$$

$\text{dist}(p, q)$ is the distance between two points, and M is the dataset.

The principle of the DBSCAN algorithm is as follows.

DBSCAN searches for clusters by examining the Eps neighborhood of each point in the dataset and creates a cluster with p as the core object if the Eps neighborhood of point p contains more than MinPts.

The iterative aggregation of DBSCAN directly obtains the density-reachable objects from these core objects and merges some of the density-reachable points.

The process ends when no new points are added to any clusters.

The DBSCAN algorithm can divide the dataset into core points, boundary points, and noise points and can determine the number of clusters at the same time, but the clustering effect of datasets with uneven density distribution is poor.

Because the population distribution has the characteristics of uneven density, the Gaussian mixture model can better classify the uneven data. A Gaussian mixture model (GMM) was used to classify the population density before running DBCSCAN clustering. When the Gaussian mixture model is used for clustering, the parameters in the GMM can be iteratively calculated by the EM algorithm (expectation maximization algorithm).

The parameters of the final initialized Gaussian mixture model can be expressed by the following equation:

$$\begin{aligned}
\mu'_k &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})x_n,\\
N'_k &= \sum_{n=1}^{N}\gamma(z_{nk}),\\
\sum_k{}' &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})(x_n-\mu_k)(x_n-\mu_k)^T.
\end{aligned} \tag{19}$$

According to the Bayesian formula, the posterior probability $p(x|z)$ is obtained:

$$\gamma(z_k) = p(z_k = 1|x) = \frac{\pi_k N(x|\mu_k, \sum_k)}{\sum_{j=1}^{K}\pi_j N\left(x\middle|\mu_j, \sum_j\right)}, \tag{20}$$

where $\gamma(z_k)$ represents the posterior probability of the kth component. Assuming $x = \{x_1, x = \cdots x =\}$, there are 3 parameters in the GMM model that need to be estimated, μ is the mean position of the Gaussian peak of the mixture Gaussian kernel, Σ is the shape mean of the mixture Gaussian curve, π_k is equivalent to the value of each component

Table 1: Statistics of the five datasets.

Dataset	Resolution	Color	Num	FPS	Max	Min	Ave	Total	Modality
MICC	480 × 640	RGB+D	3358	—	11	0	5.2	17,630	Depth
UCSD	158 × 238	Grey	2000	10	46	11	24.9	49,885	Grey
Mall	640 × 480	RGB	2000	<2	53	11	31.2	62,315	RGB
FDST	1920 × 1080	RGB	15,000	—	50	13	26.3	394,081	RGB
CrowdFlow	1280 × 720	RGB	3200	25	1451	98	462	1,478,406	RGB

$N(x|\mu_k, \Sigma_k)$ weight, and $z_k\,(1 \le k \le K)$ can only take two values of 0 or 1; $z = 1$ means the probability of selecting the kth class $(p(z_k = 1)) = \pi_k)$; $z_k = 0$ means that the kth class is not selected as the class.

The detection network (MRCNN) is based on the improved DBSCAN density clustering algorithm. GMM can adaptively determine the optimal density threshold with different density clusters and has good clustering results on datasets with uneven density distribution. The improved density clustering algorithm of DBSCAN can solve the problem of denoising and multidensity clustering, which improves the cluster detection of people of different scales in the crowd.

4. Experiments

4.1. Dataset Selection Training Configuration

4.1.1. Dataset Introduction and Evaluation Criteria. The video crowd counting method has been evaluated experimentally on five labeled sequence datasets, the MICC dataset (RGBD), CrowdFlow dataset, FDST dataset, Mall dataset, and UCSD dataset. Experiments on the RGBD dataset and RGB dataset verify the feasibility and applicability of our proposed method in two different video modalities. We first give the parameters of the five key datasets of video frames used in the experiments. Then, the comparison results between the method used in this paper and the current state-of-the-art video crowd counting methods under these datasets are given, and the crowd detection results with high recall rate are given. Finally, this paper conducts ablation experimental studies to demonstrate the independent effectiveness of each unit method in our method ensemble.

(1) MICC Dataset. The MICC dataset is a sequence of frames shot from an indoor fixed scene. A total of 3358 frames of crowd RGBD images are obtained. The resolution of the frames is 480 × 640 pixels. The image acquisition mode is RGB+RBGD. The maximum number of head annotations in a single frame is 11. The minimum number of head annotations in the frame is 0, and there are 17,630 head annotations in total. The MICC dataset contains three kinds of video sequences: stream sequence, team sequence, and group sequence. The flow sequence includes a total of 1260 frames containing 3542 pedestrian bounding box annotations. There are 5031 pedestrian bounding box annotations in the 918 frames of the queue sequence and 9057 pedestrian bounding box annotations in the 1180 frames of the group sequence, and each RGB image corresponds to an RGBD image. In streaming sequences, people walk from one location to another and the frame rate is lower. In the queuing sequence, the acquisition frame rate is larger and pedestrians move slowly. In the group sequence, people are constrained to the area of action, as shown in Table 1.

(2) UCSD Dataset. The UCSD dataset is a sequence of scenes captured by surveillance cameras in the school. A total of 2000 frames of crowd gathering RGB images were obtained. The resolution of the frames was 238 × 158 pixels. The maximum number of head annotations in a single frame was 46, and the minimum number of head annotations in a single frame was 11. There are 49,885 head annotations in total. The acquisition frame rate is 10 fps. The dataset gives the center ground truth label for each pedestrian. ROI and perspective maps are provided in the dataset, as shown in Table 1.

(3) Mall Dataset. The Mall dataset was collected from close-up shots in a public area of a shopping mall. A total of 2000 frames of aggregated RGB annotated images of moving and stationary pedestrians were acquired. ROI and perspective maps are also provided in the dataset. The resolution of each frame is 640 × 480 pixels, the maximum number of head annotations in a single frame is 53, and the minimum number of head annotations in a single frame is 11, with a total of 62,315 head annotations. The dataset gives ground truth labels for each pedestrian. This dataset is more challenging with chiaroscuro conditions and reflective shadows on glass. ROI and perspective maps are provided in the dataset, as shown in Table 1.

(4) FDST Dataset. The FDST dataset includes both indoor and outdoor scenes. The indoor scene is collected from close-range monitoring of a public area of a shopping mall, and the outdoor scene is collected from close-range monitoring of road and street scenes, including a total of 100 images captured from 13 different scenes. The video consists of a total of 150,000 RGB images, which contain 394,081 header comment boxes. And the resolution of the frames is 1920 × 1080 pixels. The maximum number of head annotations in a single frame is 50, and the minimum number of head annotations in a single frame is 13, with a total of 394,081 head annotations. The dataset gives local annotations for each pedestrian, as shown in Table 1.

(5) CrowdFlow Dataset. The dataset consists of 5 different VR scenes and 10 sequences of RGB-rendered sequence

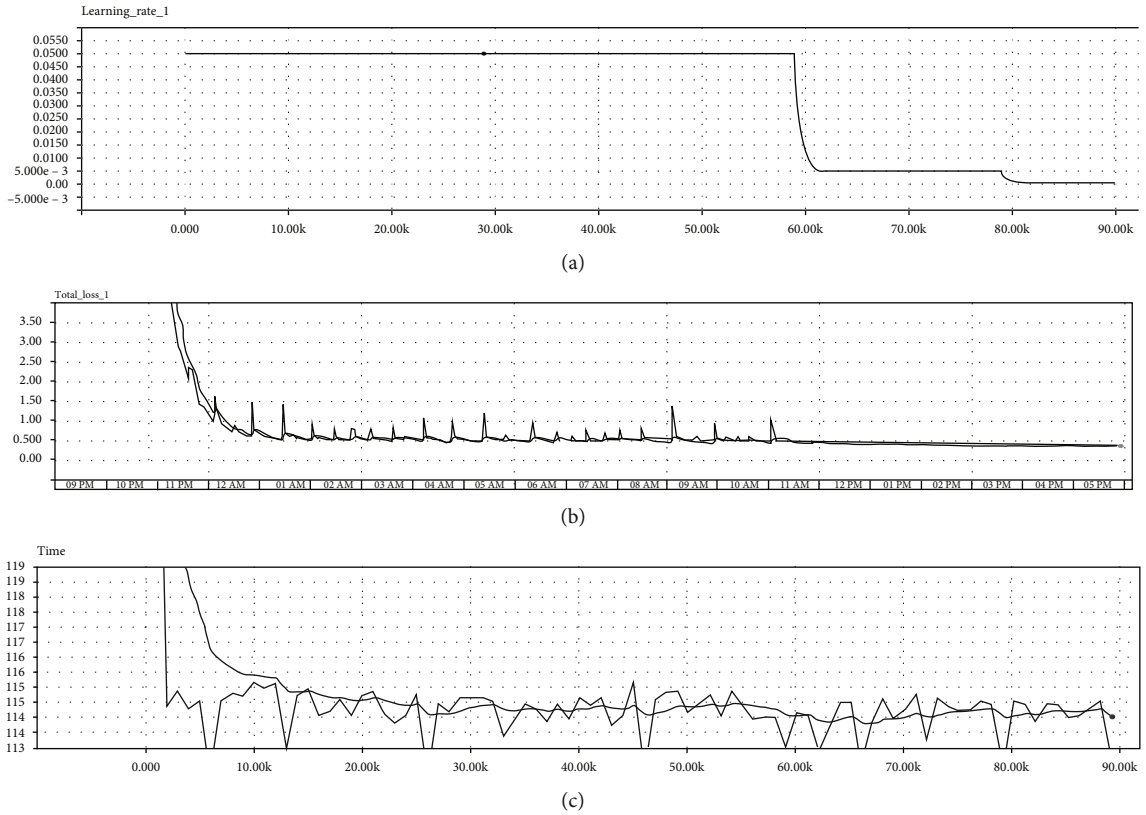

Figure 6: Training process: (a) learning rate setting curve; (b) loss function change curve; (c) total training time change curve.

images. The rendering process includes still camera rendering and moving camera rendering (simulating drone surveillance). A total of 3200 crowd-gathered RGB-rendered images and 1,478,406 annotated pedestrians were collected at a frame rate of 25 fps with a frame resolution of 1280 × 720 pixels. The maximum number of head annotations in a single frame is 1451, and the minimum number of head annotations in a single frame is 98. The dataset contains very dense crowd picture sequences, so it is very challenging to count dense crowd videos, as shown in Table 1.

4.1.2. Dataset Parameter Setting and Training

(1) MICC Dataset. Since the three sequences of flow, groups, and queue have the same scene and participant characteristics, 20% of the RGB images of the flow, group, and queue scenes and their corresponding RGBD images can be input as the training set, and the remaining 80% of the RGB images can be input. The images and their corresponding RGBD images are used as the test set. This dataset is for training a crowd counting network with depth perception.

(2) UCSD Dataset. In order to be able to compare with the baseline method, 601 to 1400 frames are selected as the training set, and the remaining 1200 frames are used as the test set.

(3) Mall Dataset. We use the first 1-800 frames in the Mall dataset as the training set and the remaining 801-2000 frames as the test set.

(4) FDST Dataset. 9000 frames of 60 videos in the FDST dataset are used as the training set, and 6000 frames of 40 videos are used as the test set to complete the training. This training method is adopted by most of the comparison methods.

(5) CrowdFlow Dataset. In the CrowdFlow dataset, we use the first three sequences of static crowd images in the IM05_hDyn folder and moving camera scenes in the IM05 folder for training and validation, and the last two sequences for testing crowd counts.

(6) Training Settings. We train MLPCNN in an end-to-end fashion. The Gaussian parameter in MLPCNN is set to 0.5, and the standard deviation is set to 0.02. In our experiments, MLPCNN chooses Stochastic Gradient Descent (SGD) with momentum to train the model with a small learning rate for the MICC deep dataset, CrowdFlow dataset, FDST dataset, Mall dataset, and UCSD dataset, the initial learning rate Set to 0.005 and Momentum to 0.85. In this way, the convergence speed is faster during training, and the training process is shown in Figure 6. In particular, the perspectives provided by the Mall dataset and the UCSD dataset are ground truth density maps adjusted by setting $\sigma = 0.3M(p)$, and the other datasets are initialized with default parameters as training. The implementation of our method is carried out under the Pytorch framework, and the hardware uses three NVIDIA 1080 Ti GPU graphics cards and four Intel(R) E5-2630 v4 CPUs to ensure the performance requirements of graphics cards and computing units.

Table 2: Comparison of different state-of-the-art methods on MICC dataset, UCSD dataset, Mall dataset, FDST dataset, and CrowdFlow dataset.

Method	MAE (D0)	MSE (D0)	MAE (D1)	MSE (D1)	MAE (D2)	MSE (D2)	MAE (D3)	MSE (D3)	MAE (D4)	MSE (D4)
RetinaNet [65] (M1)	1.641	2.554	—	—	—	—	—	—	212	235
DetNet [66] (M1)	1.541	2.382	—	—	—	—	—	—	195	224
Idrees et al. [63](M1)	1.396	2.642	—	—	—	—	—	—	148.6	206
Gaussian regression [67] (M2)	—	—	2.24	7.97	3.72	20.1	—	—	—	—
Ridge regression [68] (M2)	—	—	2.25	7.82	3.59	19	—	—	—	—
MCNN [69] (M2)	1.5	2.259	1.07	1.35	2.24	8.5	3.77	4.88	172.8	216
Switch-CNN [70] (M2)	—	—	1.62	2.1	—	—	—	—	—	—
CSRNet [71] (M2)	1.359	2.125	1.16	1.47	—	—	—	—	137.8	181
MCNN-adaptive [63] (M1 & M2)	1.489	2.114	1.02	1.26	2.12	8.1	3.54	4.65	168.5	205
CSRNet-adaptive [63] (M1 & M2)	1.343	2.007	1.01	1.19	2.09	7.9	3.51	4.57	136.5	180.5
RDNet [72] (M1 & M2)	1.38	2.551	—	—	—	—	—	—	170.5	225
RPNs [73] (M1 & M2)	—	—	0.97	1.12	—	—	—	—	—	—
FCN-rLSTM [63] (M3)	—	—	1.54	3.02	—	—	—	—	—	—
ConvLSTM-nt [63] (M3)	1.581	2.568	1.73	3.52	2.53	11.2	—	—	—	—
ConvLSTM [63] (M3)	1.48	2.256	1.3	1.79	2.24	8.5	4.48	5.82	—	—
BiConvLSTM [63] (M3)	1.356	2.105	1.13	1.43	2.1	7.6	4.12	4.48	—	—
DRC-ConvLSTM (M4)	—	—	1.02	1.35	2.02	7.48	4.05	4.39	141.6	190.1
Our method (M4) (DRC-ConvLSTM+DAM+DAGK)	1.226	2.001	—	—	—	—	—	—	—	—

4.2. Comparison with State-of-the-Art Methods

4.2.1. Crowd Counting in Video. We collected four different categories of crowd counting methods: object detection-based methods (M1), regression statistics-based crowd counting methods (M2), regression-guided detection-based crowd counting methods (M1+M2), and ConvLSTM-based crowd counting methods (M3). Compare specific experimental data citations for the most representative methods in each field. Finally, the performance of each type of method on these same datasets is given, and the difference and obvious improvement of the method used in this paper with the current crowd counting methods are given. From Table 1, it can be found that different categories of methods have certain limitations and room for improvement. This paper only compares the specific performance of the four types of counting methods related to this paper, as shown in Table 1.

(1) Metrics. We use mean absolute error (MAE) and mean squared error (MSE) to evaluate different methods based on commonly used metrics in existing crowd counting work:

$$\mathrm{MAE} = \frac{1}{N}\sum_{1}^{N}|z_i - \widehat{z}_i|, \tag{21}$$

$$\mathrm{MSE} = \sqrt{\frac{1}{N}\sum_{1}^{N}(z_i - \widehat{z}_i)^2}, \tag{22}$$

where N is the total number of test images, z_i is the actual number of people in the ith test image, and $\widehat{z}_i$ is the estimated number of people in the ith image.

(2) Detection Methods (M1). As shown in Table 2, RetinaNet utilizes ResNet and efficient network feature pyramid and adopts anchor box adaptive anchors, and the detection crowd size error MAE = 1.641 and MSE = 2.554. The CrowdFlow dataset detects crowd size error MAE = 212 and MSE = 235. The structure of DetNet is improved on the basis of ResNet50, because ResNet50 itself has excellent performance and improves detection, so the results achieved 0.1/0.172 improvement in MAE and MSE indicators of the MICC dataset and achieved MAE and MSE indicators of CrowdFlow dataset 17/11 improvement. Using SIFT head detection, Idrees et al. [63] achieved a 0.145 improvement in the MAE indicator of the MICC dataset compared to DetNet and a 46.4/18 improvement in the MAE and MSE indicators of the CrowdFlow dataset. Although these detection frameworks can detect some small-scale objects, the M1 method cannot cope with dense crowds with serious occlusion.

(3) Regression Methods (M2). Table 2 [67, 68] shows the traditional regression methods, and most traditional regression methods are regression methods based on CNN density map. Compared with detection-based methods, CSRNet [71] compared to Idrees et al. [63] and MAE and MSE indicators on the CrowdFlow dataset achieve a 0.8/25 boost. But the M2 method lacks deeper spatial features, so the method will be more likely to fail. And the pedestrian space position cannot be located.

Table 3: Error comparison between the method in this paper and some recent most representative methods on UCF_QNRF and UCF-CC-50 image datasets.

Methods/year	UCF_QNRF		UCF-CC-50	
	MAE	MSE	MAE	MSE
DFN [74]/2021	218.2	357.4	402.3	434.1
SS-CNN [75]/2021	115.2	175.7	229.4	325.6
SD-CNN [76]/2019	—	—	235.7	345.6
Ours	146.4	216.7	358.5	389.3

(4) Density Regression-Guided Detection Methods (M1 and M2). As shown in Table 2, the M1+M2 method implements the crowd counting method for regression-guided detection. MCNN-adaptive and CSRNet-adaptive add the depth perception branch on the basis of density map regression, so compared with MCNN and CSRNet [10], the MAE and MSE indicators of the MICC dataset achieved 0.1/0.172 and 0.016/0.118 improvements, the UCSD dataset MAE and MSE indicators achieved 0.05/0.09 and 0.15/0.26 improvements, and the CrowdFlow dataset MAE and MSE indicators achieved 4.3/11 and 1.3/0.5 boost. RPN is a method of density regression to guide target detection. It has to be said that this method performs best in UCSD datasets. The M1 and M2 method can improve the crowd counting accuracy and obtain the detection frame position, but the M1+M2 method relies on a large number of bounding box annotations, because the expensive annotation cost limits the further application of detection tasks.

(5) ConvLSTM Improvement Methods (M3). Although these CNN-based annotations can capture strong semantic annotations, they lack contextual temporal correlation information, which may lead to many false positive results. ConvLSTM can provide contextual information related to consecutive frames and time. As shown in Table 2, the M3 method makes full use of temporal information by designing the network structure of ConvLSTM to improve the performance of this dataset. Among these structures, the density map generated by the most complex BiConvLSTM method is closest to the ground truth, achieving the MAE = 1.356 /MSE = 2.105 on the MICC dataset and the MAE = 1.13 /MSE = 1.43 on the UCSD dataset, the Mall dataset achieves the MAE = 2.1/MSE = 7.6, and the FDST dataset achieves the MAE = 4.12/MSE = 4.48. From the results of FCN-rLSTM, ConvLSTM-nt, unidirectional ConvLSTM, and bidirectional ConvLSTM, we can conclude that the video crowd counting performance is BiConvLSTM>ConvLSTM >FCN-rLSTM>ConvLSTM-nt in the order from strong to weak, and DRC-ConvLSTM has the smallest error. But the M3 method relies on strong semantic annotation information.

(6) Our Proposed Method (M4). In order to further improve the network performance and improve the problem that the M3 method relies on the lack of strong semantic annotation information, and at the same time improve the ability of the M1+M2 method to correlate context information in time, this paper proposes a potential way to further improve the M3 method by incorporating both depth-aware and adaptive Gaussian kernels in the M3 method to estimate the number of people in a video. Table 2 compares our model variants with state-of-the-art methods. It was found that the DRC-ConvLSTM method of M4 performed well on the RGB image dataset, such as the UCSD dataset MAE and MSE indicators achieved a 0.11/0.08 improvement, the Mall dataset MAE and MSE indicators achieved a 0.18/0.12 improvement, and in the FDST dataset, MAE and MSE errors decreased by 0.07/0.09. It is found that the DRC-ConvLSTM method adds a depth-aware kernel and an adaptive Gaussian kernel to the accuracy compared to BiConvLSTM on the RGBD image dataset.

4.2.2. Crowd Counting in Image. This paper compares the errors of the models used in the most representative references [74–76] in UCF_QNRF image dataset and UCF-CC-50 image dataset. The conclusion after comparison is that the performance of the proposed method in the UCF_QNRF dataset is better than that of the DFN model, but the error is worse than that of SS-CNN. The performance of the UCF-CC-50 dataset is better than that of the DFN model, but the error is worse than that of the SS-CNN and SD-CNN models, as shown in Table 3. The reason is that SS-CNN and SD-CNN have made a lot of contributions in the multi-scale sensing mechanism, but the method in this paper only uses the adaptive Gaussian kernel to judge the head size of small targets which has certain limitations. In addition, the population density of the depth sample MICC dataset used in this paper is low, so the detection and recognition error of the high-density dataset UCF-CC-50 is large.

4.2.3. Comparison of Complexity and Time Consumption of Advanced Models. In the UCSD dataset, the error of this method is the lowest among many improved methods of ConvLSTM. In order to verify the real-time performance of video real-time counting network, the most advanced real-time counting network is compared in model parameter quantity (Params) and frame rate (FPS). Model parameter quantity (Params) is used to measure model complexity, and frame rate (FPS) is used to measure model real-time performance. Through comparison, it is found that the method described in this paper adds a double reverse conduction mechanism to the BiConvLSTM model, so the parameters of DRC-ConvLSTM are twice as large as those of BiConvLSTM. However, too many model parameters increase the processing time of video images, so some real-time performance is sacrificed. FPS = 0.43 for DRC-ConvLSTM, as shown in Table 4.

Analyzing the qualitative results shows that our method performs well in datasets with varying degrees of crowding. The main reason is that our proposed network learns more spatiotemporal contextual information, which is consistent with our original motivation. The results verify the effectiveness of our method.

In Figure 7, we present one fixed scene in five datasets, including four references in each scene (black: ground truth, red: our method, blue: BiConvLSTM, AND green: RDNet). From Figure 7, we can see that our method outperforms

Table 4: Detailed information comparison of the error, complexity, and time consumption of the state of the art on the UCSD dataset.

Method	MAE (UCSD)	MSE (UCSD)	Params	Frames/s (fps)
MCNN [69] (M2)	1.07	1.35	0.13×10^6	45.81
Switch-CNN [70] (M2)	1.62	2.1	1.543×10^6	3.86
CSRNet [71] (M2)	1.16	1.47	16.26×10^6	0.37
MCNN-adaptive [63] (M1 & M2)	1.02	1.26	0.13×10^6	45.81
CSRNet-adaptive [63] (M1 & M2)	1.01	1.19	16.26×10^6	0.37
RPNs [73] (M1 & M2)	0.97	1.12	2.15×10^6	2.77
FCN-rLSTM [63] (M3)	1.54	3.02	5.74×10^6	1.18
ConvLSTM-nt [63] (M3)	1.73	3.52	4.68×10^6	1.27
ConvLSTM [63] (M3)	1.3	1.79	3.43×10^6	1.74
BiConvLSTM [63] (M3)	1.13	1.43	6.86×10^6	0.87
DRC-ConvLSTM (ours) (M4)	1.02	1.35	13.72×10^6	0.43

BiConvLSTM, BiConvLSTM outperforms ConvLSTM, and ConvLSTM outperforms RDNet in all scenarios, which proves that crowd counting improves after including temporal information and depth perceptron. For scenario 5, a careful observation of the green count curve RDNet reveals that there is an obvious underestimation problem in this scenario. In fact, many people will have a large area of shadow when they walk to the right channel, so many pedestrians are blocked by shadows. This large area of occlusion will seriously affect the detection effect of RDNet under visible light. However, our method and THE BiConvLSTM method are able to discover moving target pedestrians under the continuous learning of time series frames, so the red and blue curves are closer to the black ground truth baseline. Since our method adds depth perceptron and adaptive Gaussian kernel, our method is closer to the baseline. Our spatiotemporal model tends to count them because detection is more accurate with feature information from time frames.

To confirm the effectiveness of our method in density warning, we complete counting experiments under three different density videos. We can use this network to directly regress the density of a video stream, which requires consecutive images as input. Figure 8 shows the results on three real datasets. For MICC, FDST, and CrowdFlow, since the MICC datasets we use have adequate frame annotations, this enables us to pretrain depth perceptrons and regressors. From the density map results generated on MICC, FDST, and CrowdFlow, the density map of our method is very close to the ground truth density map, which also improves the counting performance to a certain extent. When the crowd becomes very sparse (as in scenario 1), the overcorrelation of temporal features does not lead to overlearning, thus demonstrating the effectiveness of this method at different densities.

4.2.4. Crowd Detection. At present, it is difficult to separately count people of different scales in videos, and most counting methods can only complete the global population estimation. Although the crowd density map can estimate the total number of crowds in dense crowds, it cannot locate the most crowded crowds in the video. If the crowds of different scales can be counted separately, the number of crowds in different areas can be more comprehensively evaluated. Since the dense crowd dataset gives the annotation of the real location of the crowd head, first, we extract the head center from the annotation, then we extract the head location points in the density map with the help of adaptive Gaussian kernel, RGBD depth perceptron, and DRC-ConvLSTM. By evaluating the precision, recall, and F-measure between the coordinates of the estimated head location points in this paper and the ground truth annotation location point coordinates, we further verify the localization performance of our method in videos of different scales. Due to the existence of various density clustering phenomena in the density of the location points, the number of people of different scales is estimated by means of the improved DBSCAN clustering algorithm. From left to right are the original image, global human head count, high-density human head count, medium-density human head count, and low-density human head count, as shown in Figure 9. Compared with the current more complex feature extraction detection framework, DBSCAN density clustering can effectively cluster multiscale crowds, thereby improving the effectiveness of local crowd counting.

4.3. Ablation Study

4.3.1. Effectiveness of Density Map Regression-Guided Detection. We conduct ablation experiments on density regression-guided detection. As shown in Table 5, three different variables are selected for qualitative analysis, namely, depth perception model DAM, depth-adaptive Gaussian kernel DAGK, and DRC-ConvLSTM, in which DAM can locate the layer of each target in the RGBD image; DAGK generates an adaptive Gaussian kernel based on the RGBD head image annotation points, and we can use these Gaussian kernel density functions $H(x)$ to generate the

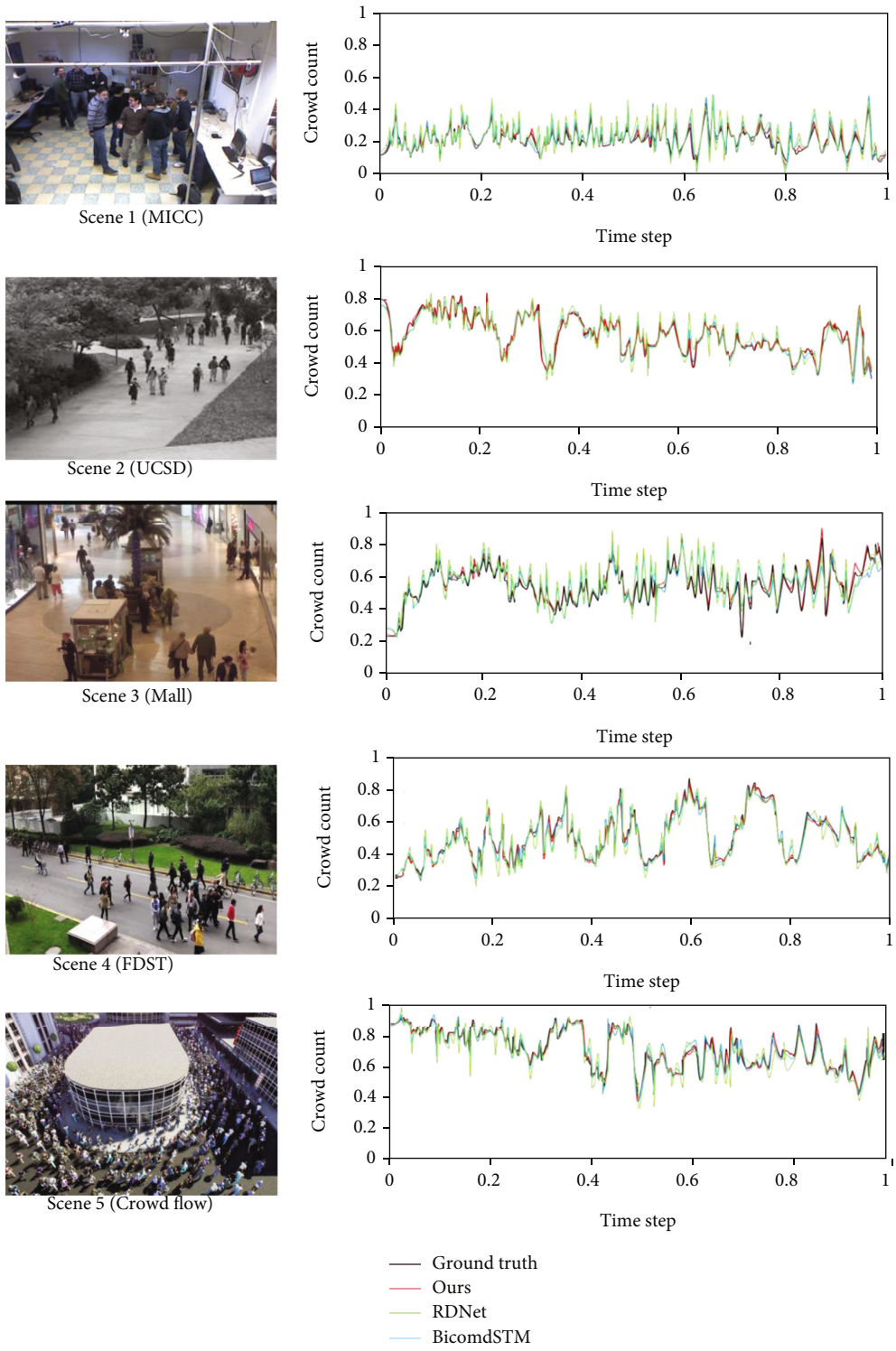

Figure 7: Crowd count of time steps in prediction vs. ground truth on 5 scenes. Black: ground truth; red: our method; blue: BiConvLSTM; green: RDNet.

corresponding density map; DRC-ConvLSTM can learn continuous features in image sequence frames and then construct and time associated joint features.

On the MICC and UCSD datasets, we compare five combinations and finally find the best combination. From the results, we can see that using DAGK alone in the MICC deep crowd counting dataset has a large error in the counting results. We believe that the reason is that DAGK cannot fully obtain the spatial boundary constraints of crowd features from the depth layer. It is worth noting that the method using only DAM does not converge, because the density function $H(X)$ cannot be constructed by only locating the

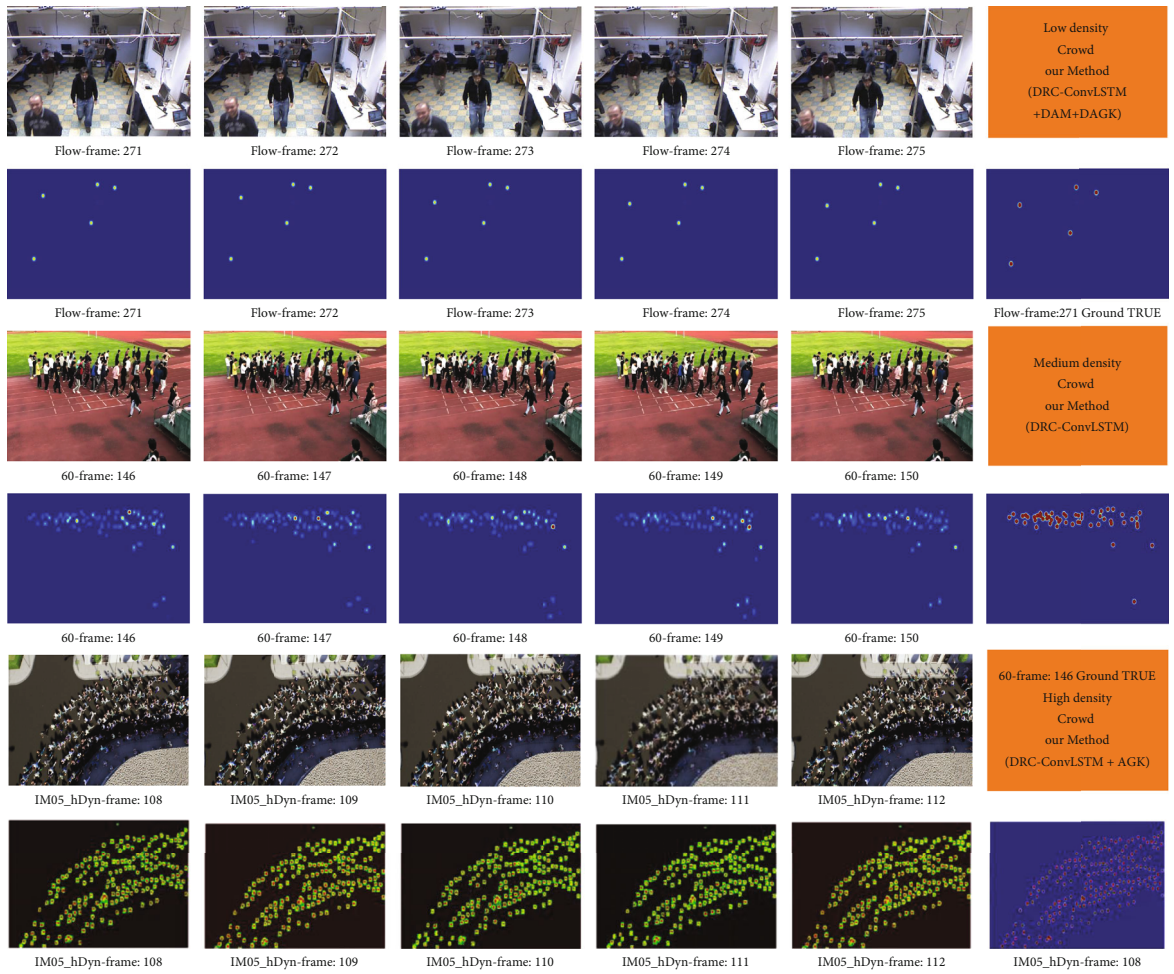

Figure 8: Five consecutive frame sequences are shown in three datasets in a specific scene with low, medium and high population densities, and the density map sequence of the corresponding frames is generated by the method in this paper.

depth layer without finding the Gaussian kernel corresponding to the target. Using DAM and DAGK at the same time is more suitable for processing deep features, especially for denser crowds with higher accuracy. From the experimental results, it can be seen that the accuracy of using only DRC-ConvLSTM is higher than that of using DAM and DAGK at the same time. The reason is that DRC-ConvLSTM can learn the deep temporal correlation features of the target in the deep time series of MICC, so it is more efficient in image counting. Finally, DAM and DAGK are added to BiConvLSTM to further increase the depth layer information extraction of the model and the depth information matching of the Gaussian kernel. The result exceeds the accuracy of using DRC-ConvLSTM alone, because the depth perception kernel will produce better association constraints. DRC-ConvLSTM has performed well in the UCSD dataset and has surpassed BiConvLSTM, and the use of multivariate Gaussians will shorten the convergence time of the Gaussian kernel. Therefore, DRC-ConvLSTM is equally effective for crowd counting in the MICC dataset and UCSD dataset.

4.3.2. Effectiveness of the DRC-ConvLSTM Method. We performed ablation experiments on BiConvLSTM and its three variants. As shown in Table 6, the video crowd counting performance of BiConvLSTM and its three different variants is explored. RC stands for BiConvLSTM, and the design of BiConvLSTM includes two reverse sequence features passed to the channel; SDT is a variant of BiConvLSTM, and BiConvLSTM's design includes two direct sequence feature transmission channels; DRC is a variant of BiConvLSTM, and the design of BiConvLSTM includes two layers of parallel reverse sequence feature transmission channels; and DCC is a variant of BiConvLSTM, and the design of BiConvLSTM includes two layers with parallel isotropic sequential characteristic conduction channels. From the results, we can see that the accuracy of SDT's counting results is not as high as that of RC, because the same-direction feature conduction can only strengthen the strength of the same feature but cannot increase the diversity of time-series features, which is not conducive to sequence feature feedback. In the same way, the accuracy of the counting result of DCC is not as high as that of DRC. However, the accuracy of the count results of DCC is higher than that of DRC, because the two-layer reverse feature can strengthen the characteristics of each column of reverse features more than one-layer reverse features. Finally, DRC also has the highest head detection

Figure 9: Comparison of detection results of different densities and scales of people: (a) global detection; (b) high scale detection; (c) medium density detection. (d) low scale detection.

Table 5: Ablation studies on our dataset.

Component	C.1	C.2	C.3	C.4	C.5
DAM		√	√		√
DAGK	√		√		√
DRC-ConvLSTM				√	√
MAE (MICC)	1.456	—	1.354	1.251	1.226
MSE (MICC)	2.468	—	2.249	2.226	2.001
MAE (UCSD)	—	—	—	1.02	—
MSE (UCSD)	—	—	—	1.35	—

DAM: depth-aware model; DAGK: depth-adaptive Gaussian kernel.

Table 6: Comparison of different feature counting methods on ShanghaiTechPartA.

Component	C.1	C.2	C.3	C.4	Ours
BiConvLSTM	√	√	√	√	—
SDT	√				—
RC			√		—
DCC		√			—
DRC				√	—
MAE	1.28	1.24	1.18	1.12	1.02
MSE	1.53	1.48	1.41	1.38	1.35
Precision	0.746	0.826	0.909	0.928	0.931
Recall	0.716	0.722	0.747	0.795	0.801

RC: reverse conduction; SDT: same direction transmission; DRC: double reverse conduction; DCC: double coconduction.

accuracy, which is mainly due to the fact that DRC can obtain the contextual feature compensation of annotation points, so precision and recall are the highest.

Precision and recall are compared on UCSD for different variants of ConvLSTM and BiConvLSTM in Figure 10. In the images, we see that ConvLSTM performs worse than different variants of BiConvLSTM in both precision and recall, which confirms that changes in the number of feature layers and conduction directions further affect the performance of ConvLSTM. Like SDT, it brings slight improvement to ConvLSTM but not as good as BiConvLSTM. However, using stronger constraints, i.e., using the inverse DRC-ConvLSTM with two layers, lead to a larger improvement, confirming the importance of correctly constructing the feature flow. As expected, the dual-layer DRC performed optimally among all variants, and the performance of DRC and RC was further improved. This confirms that the use of two-layer inverse DRC-ConvLSTM feature flow is an effective improvement on feature augmentation theory. But not all layers increase, such as DCC is not as accurate as RC, so we confirm the important contribution of reverse feature conduction used in multilayer ConvLSTM.

4.3.3. Effectiveness of Counting Different Dense Crowds Based on Density Clustering. Using the improved DBSCAN in this paper can achieve clustering of crowd points at different scales, because the Gaussian mixture of dense crowds has the clustering effectiveness of similar density regions. Using a Gaussian mixture model can divide groups of different densities, while the improved DBSCAN clusters people of different scales according to different density levels to

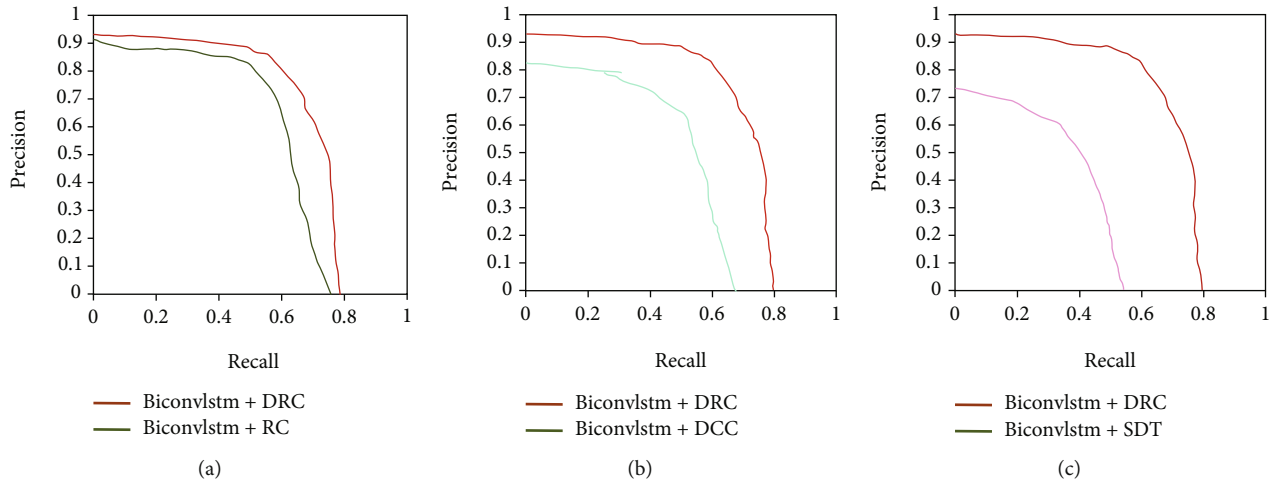

Figure 10: Traditional method density map and detection results: (a) input image; (b) YOLO V3 crowd head detection results; (c) MCNN ground truth density map; (d) MCNN estimated density. Precision-recall curves for all object classes. (a) Average precision-recall curves of DRC and RC; (b) average precision-recall curves of DRC and DCC; (c) average precision-recall curves of DRC and SDT.

Table 7: Comparison of different classification detection results on ShanghaiTechPartA dataset.

Component	C.1	C.2	Ours(local area count)
DRC-ConvLSTM	√	√	—
DBSCAN	√		—
DBSCAN+		√	—
MAE	—	1.02	1.02
MSE	—	1.35	1.35

DRC: double reverse conduction.

Figure 11: Crowd detection in CrowdFlow dataset: (a) before detection; (b) after detection.

complete the individual counting of local groups, while the traditional DBSCAN requires a fixed number of parameters MinPts and distance parameters Eps. Therefore, the clustering of populations with large scale changes cannot be completed.

The DBSCAN algorithm is verified by the experimental study of ablation, so as to guide more accurate local population counting and head detection of different scale populations. Therefore, two different clustering combination schemes were carried out, and the results are shown in Table 7. From the results, we find that using DBSCAN on the DRC-ConvLSTM method cannot achieve estimation with less error; the reason is that DBSCAN cannot find crowds of different scales, because traditional DBSCAN is only effective for single density clustering. However, using the improved DBSCAN on the DRC-ConvLSTM method can identify people with large differences in density, so the combination of the improved DBSCAN and the DRC-ConvLSTM method in the above four cases can achieve separate counting of regions at different scales. The MAE and MSE accuracies of the method used in this paper are the highest.

5. Conclusion

In this paper, a network DRC-ConvLSTM for video crowd counting and multicrowding scale separate counting is implemented by the final design. Our proposed DRC-ConvLSTM is trained and tested on RGB and RGBD datasets. On the RGB dataset, the design of the DRC-ConvLSTM network can further extract the crowd features associated with time. On the RGBD dataset, the DRC-ConvLSTM network combined with the depth perception model and the depth-adaptive Gaussian kernel can extract the depth space edge constraint features and deep layer matching, which ultimately achieves better crowd density estimation results than using DRC-ConvLSTM alone. The depth modality effectiveness of this method is tested on the MICC depth annotation dataset. The counting effectiveness under visible light is tested and evaluated on four labeled datasets, CrowdFlow dataset, FDST dataset, Mall dataset, and UCSD dataset. Observing the experimental results, our method achieves superior results than other state-of-the-art techniques in video real-time crowd counting. The improved DBSCAN clustering algorithm has good clustering results for uneven datasets and performs well in denoising, multi-density clustering, and merged clustering, which enables DRC-ConvLSTM to count individuals at different scales.

6. Discussion

DRC-ConvLSTM has certain limitations in counting the crowd in the video. When the light is very weak, there will be large errors in crowd detection and counting. For example, Figure 11 is the crowd detection results in CrowdFlow

dataset, Figure 11(a) is the original picture of the crowd, and Figure 11(b) is the detection results. From Figure 11(b), it can be clearly seen that when the crowd passes through the shadow area, head detection can only detect a small number of heads with obvious characteristics. However, the detection rate of the head with no obvious features in the shadow area is very low. The detection rate is high in areas with sufficient light. Although the depth sensing module included in the method used in this paper can improve some detection accuracy, CrowdFlow and most current real scenes do not contain the depth information of crowd images. Therefore, in real applications, this method is greatly constrained by the difficulty in obtaining the illumination and depth information. These problems need to be solved in the future.

In addition, crowd clusters of different densities are estimated, and then, the population cluster counts in different density areas are accumulated to obtain an overall population count estimate. This is more efficient than estimating the population of the entire graph directly. Doing so allows us to impose density clustering constraints and thus estimate which part of the population in the same densely populated area is more necessary for the warning of danger of overcrowding. A promising future application is people counting using drones that can shoot over crowds to overcome excessive occlusion and scale changes.

Data Availability

The MICC dataset, CrowdFlow dataset, FDST dataset, Mall dataset, and UCSD dataset used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (Nos. 61179019 and 81571753), CERNET Innovation Project of China (No. NGII20170705), and Baotou Youth Innovative Talent Project of China (No. 0701011904).

References

[1] L. D. Huang and J. J. Jia, "Crowd disaster risk identification in large sport venues," in *Applied mechanics and materials. Vol. 584*, vol. 584, pp. 2125–2128, Trans Tech Publications Ltd, 2014.

[2] M. Yin, Z. Li, and A. Dila, "Risk assessment of safety accidents in small and medium gyms," in *IOP Conference Series: Earth and Environmental Science*, vol. 474no. 7, p. 072044, Atlanta, GA, US, 2020.

[3] X. Wang and X. Ma, "Risk control analysis of safety accident in hydrogen refueling station based on PHAST software," in *IOP Conference Series: Earth and Environmental Science*, vol. 680no. 1, p. 012119, Atlanta, GA, US, 2021.

[4] K. Gkountakos, K. Ioannidis, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "A crowd analysis framework for detecting violence scenes," in *Proceedings of the 2020 International Conference on Multimedia Retrieval*, vol. 2020, pp. 276–280, June 2020.

[5] C. A. Pouw, F. Toschi, F. van Schadewijk, and A. Corbetta, "Monitoring physical distancing for crowd management: real-time trajectory and group analysis," *PLoS One*, vol. 15, no. 10, pp. 737-738, 2020.

[6] L. Bai, C. Wu, F. Xie, and Y. Wang, "Crowd density detection method based on crowd gathering mode and multi-column convolutional neural network," *Image and Vision Computing*, vol. 105, p. 104084, 2021.

[7] C. Xu, D. Liang, Y. Xu et al., "Autoscale: learning to scale for crowd counting," *International Journal of Computer Vision*, vol. 130, no. 2, pp. 405–434, 2022.

[8] J. Wan, Z. Liu, and A. B. Chan, "A generalized loss function for crowd counting and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1974–1983, 2021.

[9] Z. Ma, X. Hong, X. Wei, Y. Qiu, and Y. Gong, "Towards a universal model for cross-dataset crowd counting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3205–3214, 2021.

[10] P. Yugendar and K. V. R. Ravishankar, "Crowd behavioural analysis at a mass gathering event," *Journal of KONBiN*, vol. 46, no. 1, pp. 5–20, 2018.

[11] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: integrated recognition, localization and detection using convolutional networks," 2013, http://arxiv.org/abs/1312.6229.

[12] R. Laroca, E. Severo, L. A. Zanlorensi et al., "A robust real-time automatic license plate recognition based on the YOLO detector," *2018 international joint conference on neural networks (IJCNN)*, 2018, pp. 1–10, Rio de Janeiro, Brazil, 2018.

[13] W. Liu, D. Anguelov, D. Erhan et al., "SSD: Single shot multibox detector," in *European conference on computer vision*, Computer Vision – ECCV 2016, pp. 21–37, Springer, Cham, 2016.

[14] X. Wei, J. Du, M. Liang, and L. Ye, "Boosting deep attribute learning via support vector regression for fast moving crowd counting," *Pattern Recognition Letters*, vol. 119, pp. 12–23, 2019.

[15] C. Liu, Y. Huang, Y. Mu, and X. Yu, "DRENet: giving full scope to detection and regression-based estimation for video crowd counting," in *International Conference on Artificial Neural Networks*, Artificial Neural Networks and Machine Learning – ICANN 2021, pp. 15–27, Springer, Cham, 2021.

[16] X. Jiang, Z. Xiao, B. Zhang et al., "Crowd counting and density estimation by trellis encoder-decoder networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6133–6142, Long Beach, CA, 2019.

[17] X. Liu, J. Yang, W. Ding, T. Wang, Z. Wang, and J. Xiong, "Adaptive mixture regression network with local counting map for crowd counting," in *European Conference on Computer Vision*, Computer Vision – ECCV 2020, pp. 241–257, Springer, Cham, 2020.

[18] K. J. Almalki, B. Y. Choi, Y. Chen, and S. Song, "Characterizing scattered occlusions for effective dense-mode crowd counting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3840–3849, 2021.

[19] J. Ma, Y. Dai, K. Hirota, and School of Automation, Beijing Institute of Technology, "A survey of video-based crowd anomaly detection in dense scenes," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 21, no. 2, pp. 235–246, 2017.

[20] E. B. Varghese and S. M. Thampi, "Application of cognitive computing for smart crowd management," *IT Professional*, vol. 22, no. 4, pp. 43–50, 2020.

[21] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor, "Fully convolutional crowd counting on highly congested scenes," 2016, http://arxiv.org/abs/1612.00220.

[22] E. B. Varghese and S. M. Thampi, "Towards the cognitive and psychological perspectives of crowd behaviour: a vision-based analysis," *Connection Science*, vol. 33, no. 2, pp. 380–405, 2021.

[23] J. A. T. Olivero, C. M. B. Anillo, J. P. G. Barrios, E. M. Morales, E. J. Gachancipá, and C. A. Z. de la Torre, "Comparing state-of-the-art methods of detection and tracking people on security cameras video," in *2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*, pp. 1–5, Bucaramanga, Colombia, April 2019.

[24] S. Suzuki, Y. Amemiya, and M. Sato, "Enhancement of gross-motor action recognition for children by CNN with OpenPose," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, pp. 5382–5387, Lisbon, Portugal, October 2019.

[25] M. Bansal, M. Kumar, and M. Kumar, "2D object recognition: a comparative analysis of SIFT, SURF and ORB feature descriptors," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 18839–18857, 2021.

[26] T. Zhang, X. Zhang, X. Ke et al., "HOG-ShipCLSNet: a novel deep learning network with hog feature fusion for SAR ship classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–22, 2022.

[27] S. Karanwal and M. Diwakar, "OD-LBP: orthogonal difference-local binary pattern for face recognition," *Digital Signal Processing*, vol. 110, p. 102948, 2021.

[28] W. Rahmaniar, W. J. Wang, C. W. E. Chiu, and N. L. L. Hakim, "Real-time bi-directional people counting using an RGBD camera," *Sensor Review*, 2021.

[29] L. Zhou, H. Yuan, and C. Ge, "ConvLSTM-based neural network for video semantic segmentation," in *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pp. 1–5, Munich, Germany, December 2021.

[30] M. Shahriari, D. Feiz, A. Zarei, and E. Kashi, "Simulating pedestrian in public places and provide a solution to improve the LOS," *Innovative Infrastructure Solutions*, vol. 7, no. 1, pp. 1–10, 2022.

[31] Z. Zuo, W. Yin, G. Yang, Y. Zhang, J. Yin, and H. Ge, "Determination of bus crowding coefficient based on passenger flow forecasting," *Journal of Advanced Transportation*, vol. 2019, 12 pages, 2019.

[32] A. Alhadhira, M. S. Molloy, M. Casasola et al., "Use of dimensional analysis in the X-, Y-, and Z-axis to predict occurrence of injury in human stampede," *Disaster Medicine and Public Health Preparedness*, vol. 14, no. 2, pp. 248–255, 2020.

[33] Z. Chen, R. Wang, Z. Zhang, H. Wang, and L. Xu, "Background-foreground interaction for moving object detection in dynamic scenes," *Information Sciences*, vol. 483, pp. 65–81, 2019.

[34] H. Ullah, I. U. Islam, M. Ullah, M. Afaq, S. D. Khan, and J. Iqbal, "Multi-feature-based crowd video modeling for visual event detection," *Multimedia Systems*, vol. 27, no. 4, pp. 589–597, 2021.

[35] H. Yu and L. Zhang, "Partial feature aggregation network for real-time object counting," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2405–2409, Toronto, ON, Canada, June 2021.

[36] C. Li, X. Li, X. Li et al., "A state-of-the-art survey of artificial neural networks for whole-slide image analysis: from popular convolutional neural networks to potential visual transformers," 2021, http://arxiv.org/abs/2104.06243.

[37] J. Zhao, W. Lei, Z. Li, D. Zhao, M. Han, and X. Hou, "Detection of crowdedness in bus compartments based on ResNet algorithm and video images," *Multimedia Tools and Applications*, vol. 81, no. 4, pp. 4753–4780, 2021.

[38] S. V. Shivapuja, M. P. Khamkar, D. Bajaj, G. Ramakrishnan, and R. K. Sarvadevabhatla, "Wisdom of (binned) crowds: a Bayesian stratification paradigm for crowd counting," in *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3574–3582, NY, US, October 2021.

[39] H. Yin, J. Wu, Z. Liu, X. Yang, Y. Qu, and H. Sun, "Optimizing the release of passenger flow guidance information in urban rail transit network via agent-based simulation," *Applied Mathematical Modelling*, vol. 72, pp. 337–355, 2019.

[40] J. Sekasi and M. L. Martens, "Assessing the contributions of urban light rail transit to the sustainable development of Addis Ababa," *Sustainability*, vol. 13, no. 10, p. 5667, 2021.

[41] Y. Liang, W. Song, and X. Dong, "Evaluating the space use of large railway hub station areas in Beijing toward integrated station-city development," *Land*, vol. 10, no. 11, p. 1267, 2021.

[42] N. Mahmoud, M. Abdel-Aty, Q. Cai, and J. Yuan, "Estimating cycle-level real-time traffic movements at signalized intersections," *Journal of Intelligent Transportation Systems*, vol. 26, no. 4, pp. 400–419, 2021.

[43] Y. Gao, J. Li, Z. Xu, Z. Liu, X. Zhao, and J. Chen, "A novel image-based convolutional neural network approach for traffic congestion estimation," *Expert Systems with Applications*, vol. 180, p. 115037, 2021.

[44] M. Luckner, I. Krzemińska, P. Wawrzyniak, and J. Legierski, "Estimating population density without contravening citizen's privacy: Warsaw use case," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4494–4506, 2022.

[45] N. Khan, A. Ullah, I. U. Haq, V. G. Menon, and S. W. Baik, "SD-net: understanding overcrowded scenes in real-time via an efficient dilated convolutional neural network," *Journal of Real-Time Image Processing*, vol. 18, no. 5, pp. 1729–1743, 2021.

[46] J. Li, Y. Wang, C. Wang et al., "DSFD: dual shot face detector," in *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5060–5069, Long Beach,CA, 2019.

[47] J. Bai, J. Ren, Y. Yang et al., "Object detection in large-scale remote-sensing images based on time-frequency analysis and feature optimization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[48] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer visionar*, p. 01355, Seoul, Korea, 2019.

[49] A. K. Pai, A. K. Karunakar, and U. Raghavendra, "A novel crowd density estimation technique using local binary pattern and Gabor features," in *2017 14th IEEE International*

Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6, Lecce, Italy, September 2017.

[50] Y. Gao and H. Yang, "Crowd counting via multi-level regression with latent Gaussian maps," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1970–1974, Toronto, ON, Canada, June 2021.

[51] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris, and Y. Aksoy, "Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9685–9694, 2021.

[52] J. Zhang, S. Chen, S. Tian, W. Gong, G. Cai, and Y. Wang, "A crowd counting framework combining with crowd location," *Journal of Advanced Transportation*, vol. 2021, Article ID 6664281, 14 pages, 2021.

[53] Y. Xue, Y. Li, S. Liu, X. Zhang, and X. Qian, "Crowd scene analysis encounters high density and scale variation," *IEEE Transactions on Image Processing*, vol. 30, pp. 2745–2757, 2021.

[54] S. Zhang, H. Li, and W. Kong, "A cross-modal fusion based approach with scale-aware deep representation for RGBD crowd counting and density estimation," *Expert Systems with Applications*, vol. 180, p. 115071, 2021.

[55] D. Lian, X. Chen, J. Li, W. Luo, and S. Gao, "Locating and counting heads in crowds with a depth prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, p. 1, 2021.

[56] C. Luo, J. Zhang, J. Yu, C. W. Chen, and S. Wang, "Real-time head pose estimation and face modeling from a depth image," *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2473–2481, 2019.

[57] Z. Wu, "Human motion tracking algorithm based on image segmentation algorithm and Kinect depth information," *Mathematical Problems in Engineering*, vol. 2021, 10 pages, 2021.

[58] D. Liciotti, M. Paolanti, R. Pietrini, E. Frontoni, and P. Zingaretti, "Convolutional networks for semantic heads segmentation using top-view depth data in crowded environment," in *2018 24th international conference on pattern recognition (ICPR)*, pp. 1384–1389, Beijing, China, August 2018.

[59] S. Kim, S. Hong, M. Joh, and S. K. Song, "DeepRain: ConvLSTM network for precipitation prediction using multi-channel radar data," 2017, http://arxiv.org/abs/1711.02316.

[60] Y. Chang and B. Luo, "Bidirectional convolutional LSTM neural network for remote sensing image super-resolution," *Remote Sensing*, vol. 11, no. 20, p. 2333, 2019.

[61] D. Chen, J. Wang, and C. Xiong, "Research on origin-destination travel demand prediction method of inter-regional online taxi based on SpatialOD-BiConvLSTM," *IET Intelligent Transport Systems*, vol. 15, no. 12, pp. 1533–1547, 2021.

[62] Y. Chen, L. Zhou, N. Bouguila, C. Wang, Y. Chen, and J. Du, "BLOCK-DBSCAN: fast clustering for large scale data," *Pattern Recognition*, vol. 109, p. 107624, 2021.

[63] F. Xiong, X. Shi, and D. Y. Yeung, "Spatiotemporal modeling for crowd counting in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5151–5159, Venice, ltaly, 2017.

[64] M. Asad, H. Jiang, J. Yang, E. Tu, and A. A. Malik, "Multi-level two-stream fusion-based spatio-temporal attention model for violence detection and localization," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 36, no. 1, p. 2255002, 2021.

[65] H. Zhang, H. Chang, B. Ma, S. Shan, and X. Chen, "Cascade RetinaNet: maintaining consistency for single-stage object detection," 2019, http://arxiv.org/abs/1907.06881.

[66] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "DetNet: a backbone network for object detection," 2018, http://arxiv.org/abs/1804.06215.

[67] A. G. Wilson, D. A. Knowles, and Z. Ghahramani, "Gaussian process regression networks," 2011, http://arxiv.org/abs/1110.4411.

[68] Y. B. Liu, R. S. Jia, Q. M. Liu, X. L. Zhang, and H. M. Sun, "Crowd counting method based on the self-attention residual network," *Applied Intelligence*, vol. 51, no. 1, pp. 427–440, 2021.

[69] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 589–597, LAS VEGAS, US, 2016.

[70] D. Sam Babu, S. Surya, and R. Venkatesh Babu, "Switching convolutional neural network for crowd counting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5744–5752, Honolulu, Hawaii, 2017.

[71] Y. Li, X. Zhang, and D. Chen, "CSRNet: dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1091–1100, SALT LAKE CITY US, 2018.

[72] F. Hong, C. Lu, W. Jiang, W. Ju, and T. Wang, "RDNet: regression dense and attention for object detection in traffic symbols," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 25372–25378, 2021.

[73] K. Sultan Daud and B. Saleh, "Scale and density invariant head detection deep model for crowd counting in pedestrian crowds," *The Visual Computer*, vol. 37, no. 8, pp. 2127–2137, 2021.

[74] S. D. Khan, Y. Salih, B. Zafar, and A. Noorwali, "A deep-fusion network for crowd counting in high-density crowded scenes," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 1–12, 2021.

[75] K. Sultan Daud and B. Saleh, "Sparse to dense scale prediction for crowd couting in high density crowds," *Arabian Journal for Science and Engineering*, vol. 46, no. 4, pp. 3051–3065, 2021.

[76] B. Saleh, K. Sultan Daud, and U. Habib, "Scale driven convolutional neural network model for people counting and localization in crowd scenes," *IEEE Access*, vol. 7, pp. 71576–71584, 2019.