Hindawi
Scientific Programming
Volume 2022, Article ID 1990951, 9 pages
https://doi.org/10.1155/2022/1990951

Research Article

Crowd Density Estimation Method Using Deep Learning for Passenger Flow Detection System in Exhibition Center

Jun Xiang [1] and Na Liu [2]

[1] *School of Economics and Trade, Chognqing College of Finance and Economic, Yongchuan, Chongqing 402160, China*
[2] *Organization United Front Work Department, Chognqing College of Finance and Economic, Yongchuan, Chongqing 402160, China*

Correspondence should be addressed to Jun Xiang; jun_xiangevent@126.com

Received 10 December 2021; Accepted 20 January 2022; Published 18 February 2022

Academic Editor: Ahmed Farouk

Aiming at the problems of crowd distribution, scale feature, and crowd feature extraction difficulties in exhibition centers, this paper proposes a crowd density estimation method using deep learning for passenger flow detection systems in exhibition centers. Firstly, based on the pixel difference symbol feature, the difference amplitude feature and gray feature of the central pixel are extracted to form the CLBP feature to obtain more crowd group description information. Secondly, use the LR activation function to add nonlinear factors to the convolution neural network (CNN) and use dense blocks derived from crowd density estimation to train the LR-CNN crowd density estimation model. Finally, experimental results show that the mean absolute error (MAE) and mean square error (MSE) of the proposed method in the UCF_CC_50 dataset are 325.6 and 369.4, respectively. Besides, MAE and MSE in part_A of the Shanghai Tech dataset are 213.5 and 247.1, respectively, and they in part_B are 85.3 and 99.7, respectively. The proposed method effectively improves the accuracy of crowd density estimation in exhibition centers.

1. Introduction

The foreign exhibition service industry has developed into a relatively mature industry, and the domestic exhibition service industry is also developing rapidly. At present, most exhibition service companies still focus on whether the exhibition can be successfully held and provide postshow analysis reports for exhibition organizers. However, there is a lack of research on realtime exhibition hall analysis services, especially in terms of passenger flow detection [1]. The exhibition service industry based on location services has gradually emerged, and various crowd density estimation solutions have emerged [2–4].

Population counting and density estimation have great practical significance [5–7], which can be extended to the following three applications:

(1) Public safety supervision: in places with dense crowds in the real scene, the staff monitors the crowd's dynamic information in realtime through electronic camera equipment, analyzes potential safety hazards, and tries to avoid them [8, 9].

(2) Intelligence collection and analysis: as far as China is concerned, residents' travel and tourism have become normal during the annual holidays. Statistics and analysis of crowd flow of major tourist venues in China are beneficial to road traffic management and arrangements. At the same time, the overall tourism policy can be adjusted according to the travel preferences and interests of the crowd in each time period obtained in the past [10, 11].

(3) Virtual model construction: it provides a reliable mathematical model for the transformation between virtual reality and reality [12].

Crowd counting and density estimation research cannot only provide important guarantees for the safety of people's lives and property but also aid in promoting the maximization of social and economic benefits. It has a wide range of application prospects and important practical significance [13–15]. Therefore, crowd counting and density estimation have gradually become a common research hotspot in academia and industry.

In the early research, scholars used the Haar wavelet transform, shape feature, directional gradient histogram, and texture feature to manually extract the detection. Counting was completed by detecting head, body, or wholebody features in crowd images [16–18]. With the improvement of hardware technology and the advancement of deep learning technology, the performance of many computer vision tasks has been greatly improved, and CNN has played an important role in tasks such as target detection, image classification, and semantic segmentation. Therefore, CNN was widely used in counting tasks, and the related performance was greatly improved [19, 20]. Reference [21] designed a multitask framework based on CNN to simultaneously estimate the density level and the number of target crowds. It used the former to provide additional information to assist the latter to improve the counting performance of the model. Reference [22] established a multicolumn CNN, using different sizes of receptive fields to obtain target features of different scales. The crowd density map was generated by fusion with a 1×1 size convolution kernel. Reference [23] used the same network to process and generate crowd density maps for input images at different resolutions, and at the same time, output attention maps to supervise the generation of crowd scale predictions. However, this method needed to reason about multiple pictures of different scales at the same time, which greatly increases the number of network calculations. Reference [24] introduced an attention mechanism to fuse features based on detection and regression, but this method did not perform well in high-density areas and could not achieve realtime prediction. In order to enhance the perception of crowd density areas, reference [25] established a series of attention modules and regression modules. It used deformable convolution to establish an attention module to detect crowd areas and improve the perception of density maps for crowds of different densities. Reference [26] proposed a self-supervised counting algorithm that uses the rule that there are always more people in large image blocks than in small image blocks in unlabeled data to establish a self-supervised learning task to improve the counting performance of the algorithm. Reference [5] proposed an end-to-end population density estimation network to generate a high-quality population density map, which can obtain high-quality map estimation. Reference [27] proposed a crowd counting method based on crossconfrontation loss and global features for high-density scenes of different scales. The cross-countermeasure loss was used to generate the residual map, and the uniformity problem of the fusion density map was solved through the consistency between different scales, extracted a wide range of contextual information and focused on the key information in the global spatial features to generate a residual map. In reference [28], a multilevel neural network is constructed to estimate population density, and good results are achieved. Reference [29] proposed a multiscale context learning module called the multiscale context aggregation module. The module first extracted information on different scales, and then adaptively aggregated it to capture the fullscale of the crowd. However, most research is still focused on traditional shallow models. The fitting ability of shallow models is limited, and the effect is better in simple image processing of crowd scenes. But when the background is more complex, crowd density estimation is more difficult, and the extraction of scale features and crowd features is not sufficient.

Based on the above analysis, this paper proposes a crowd density estimation method using deep learning for passenger flow detection systems in exhibition centers in order to solve the problems of crowd distribution, scale feature and crowd feature extraction difficulty in the exhibition center scene. Firstly, extract the difference between the amplitude feature and gray feature of the center pixel to form the CLBP feature together to obtain more descriptive information about the crowd density. Then use the LR activation function to add nonlinear factors to CNN and use the dense blocks obtained by crowd density estimation to train the LR-CNN crowd density estimation model.

2. Proposed Model Framework

The primary problem of crowd behavior analysis is to detect an area where a large crowd gathers and perform corresponding crowd behavior analysis in this area. Based on the traditional algorithm, this paper uses the complete local binary pattern (CLBP) to extract the characteristics of crowd aggregation. On this basis, the deep learning model is used to construct the detection of crowd gathering. CNN is applied to crowd group detection, and the CLBP feature is trained by operations such as convolution and pooling. After extracting the fundamental features, the prediction result of crowd gathering is obtained. Comparing with the prediction results given by actual experts, five density results are obtained: sparse, normal, low-density, medium-density, and high-density. The steps of the crowd density estimation algorithm are shown in Figure 1.

3. Image Preprocessing and CLBP Feature Extraction

The local binary patterns (LBP) feature is one of the most commonly used texture feature detection methods. However, the LBP feature is not compatible with the density detection of any level of the crowd. The real-time performance and accuracy in complex scenes are not enough. Thus, this paper proposes a CLBP feature extraction method. Traditional LBP features use rectangular neighborhoods, which are not rotation invariant. In order to realize the texture feature of rotation invariance, a circular neighborhood is added. The schematic diagram of the circular neighborhood is shown in Figure 2.

In the circular neighborhood, the neighborhood of the center pixel has a larger selection range. When certain values cannot be read directly from the pixel, the bilinear interpolation method is used to give the calculation result and the pixel is read. For the same radius, when there is a rotation,

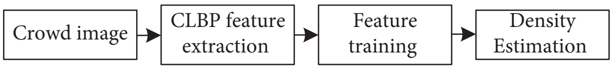


Figure 1: Crowd density estimation algorithm.

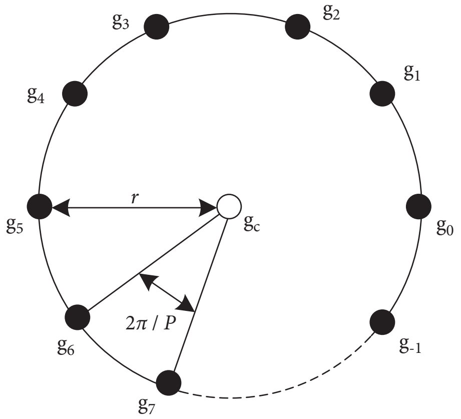


Figure 2: Circular neighborhood diagram.

the LBP value is different. In order to obtain the same LBP value, the smallest LBP value should be selected from all the results after rotation as the LBP value of the neighborhood. That is, a result that satisfies all the rotations should be selected. Figure 3 below shows the rotation result, where black represents “1” and white represents “0.” The calculated LBP value results are given in parentheses.

The introduction of the circle makes the calculation object more complicated, and the “uniform mode” calculation method should be adopted at this time. This method only performs two change calculations of 0–1 or 1–0, and the following formula is given to calculate the rotation-invariant LBP of the circular neighborhood:

$$U\left(\mathrm{LBP}_{P,R}\right) = \sum_{i=0,j=0}^{P-1} \left| s\left(g_i - g_c\right) - s\left(g_j - g_c\right) \right|, \qquad s(x) = \begin{cases} 0, x \geq 0, \\ 1, x < 0. \end{cases} \tag{1}$$

Figure 4 shows the circular neighborhood rotation-invariant LBP method for uniform mode calculation.where the number in the center of the neighborhood represents the uniform mode LBP value of LBP, and the neighborhood value is the number of “1.” The LBP value of the neighborhood of nonuniform mode ($U > 2$) is $P + 1$, and the calculation formula is as follows:

$$U\left(\mathrm{LBP}_{P,R}^{r2}\right) = \begin{cases} \sum_{i=0}^{P-1} s\left(g_i - g_c\right) \\ P+1 \end{cases}, s(x) = \begin{cases} 0, & x \geq 0, \\ 1, & x < 0. \end{cases} \tag{2}$$

Traditional LBP only extracts the difference between the pixel value of the neighborhood and the pixel value of the center point, and the characteristics that can describe the crowd are limited. In order to better express the local features of the crowd, this paper also extracts the amplitude feature of difference and the gray feature of the center pixel on top of the symbol feature of pixel difference, forming a CLBP feature. This feature can give more descriptive information about the group of people. The extraction process of CLBP is shown in Figure 5.

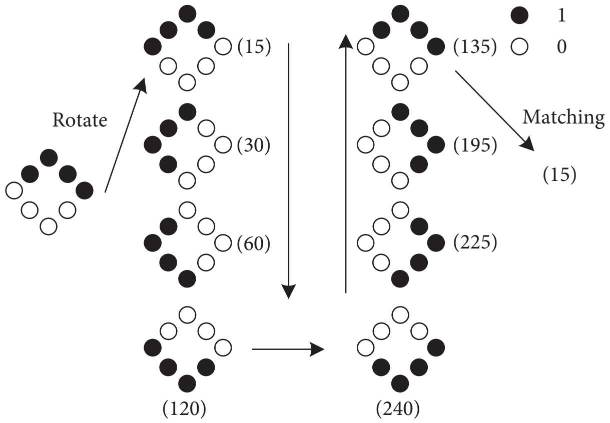


Figure 3: Calculation process of rotation invariant LBP in the circular 8 neighborhood.

The matrix (a) gives the center pixel and its 8 neighboring pixels. First, calculate the difference between the neighboring pixels and the center pixel to get matrix (b). Then generate the sign of each difference to get matrix (c). Finally, take the absolute value of all the differences of the matrix (b), obtain the magnitude of the difference, and get matrix (d). After the preprocessing is complete, the following steps are taken:

(1) The symbol matrix (c) is binarized to obtain a matrix composed of “0” and “1.” Then use the above formula (2) to calculate the characteristic S of symbols describing the difference;

(2) The global average of elements of the difference magnitude matrix (d), denoted as m_p is calculated. All elements in the matrix (d) are used to make the difference with the global average value. If the result is negative, it is recorded as “0,” and if it is nonnegative, it is recorded as “1” to generate a binary matrix. Equation (2) is used again to calculate the characteristic M that describes the magnitude of the difference;

(3) The average gray value of the center pixel is calculated, denoted as c_p. In the same way, c_p used to binarize the central pixel, and then equation (2) is used to calculate and describe the grayscale characteristics of the central pixel.

4. CNN Framework Construction of Crowd Density Estimation

4.1. Network Training Framework. After extracting the stable CLBP feature from the original crowd video sequence, it is necessary to predict the crowd group through a classifier, find the crowd group that exceeds the threshold range, and define it as a large crowd gathering situation. In this link,

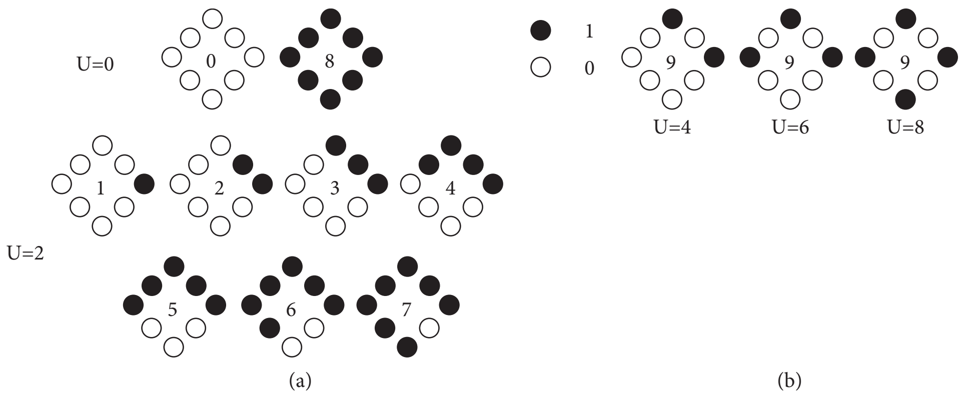
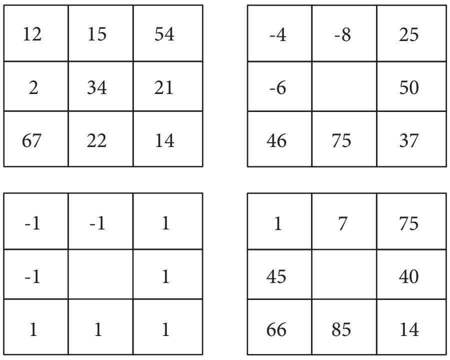


Figure 4: Calculation process of uniform mode circular neighborhood rotation invariant LBP; (a) uniform mode $P = 8$ and (b) nonuniform mode $P = 8$.



12	15	54
2	34	21
67	22	14

-4	-8	25
-6		50
46	75	37

-1	-1	1
-1		1
1	1	1

1	7	75
45		40
66	85	14

Figure 5: LBP calculation process of the uniform mode circular neighborhood rotation invariant.

traditional methods use shallow learning models for prediction and tracking, including common BP neural networks and SVM classifiers. The shallow model has achieved good results in learning and predicting a small number of samples. However, when the scene of crowd group detection is more complicated, and there are occlusions and overlaps, the limited learning ability of shallow models will gradually reduce the effect of crowd group detection, and gradually lose a certain degree of robustness. In recent years, there have been relatively few studies on the detection of crowd groups in deep learning. But deep learning has made good progress in the fields of image processing and pattern recognition. Therefore, this paper intends to use the deep learning model to predict and track the CLBP feature to obtain the clustering of a crowd.

CNN is a feedforward neural network. CNN is based on the biological vision system. It simplifies the fully connected neural network into CNN, and the connections of neurons between the upper and lower layers of adjacent layers are no longer all related. From a mathematical point of view, the weight between the two fully connected network layers is overwhelmingly zero. For example, in image processing, each pixel is only related to the local area around it. By simplifying the number of connections of neurons, the neural network can be simplified without affecting the characteristics of the image itself, reducing network complexity and reducing calculation time.

When the input $x_t\,(t = 1, 2, \ldots, n)$ and the filter $f_t\,(t = 1, 2, \ldots, m)$ are given, the input signal length n is much greater than the filter length m, and the output of one-dimensional convolution is

$$y_t = \sum_{k=1}^{n} f_k \cdot x_{t-k+1}. \tag{3}$$

One-dimensional convolution can be used in signal processing. When the filter is $f_t = 1/n$, the convolution is equivalent to the moving average of the signal sequence. Two-dimensional convolution is often used in image processing. Given an image $x_{ij}, 1 \le i \le M, 1 \le j \le N$ and filter $f_{ij}, 1 \le i \le m, 1 \le j \le n$, generally $m \ll M, n \ll N$. The output of a convolution is

$$y_{i,j} = \sum_{u=1}^{m} \sum_{v=1}^{n} f_{uv} \cdot x_{i-u+1,j-v+1}. \tag{4}$$

Figure 6(a) is the fully connected layer of the network. If there are $n^{(l-1)}$ neurons in the l layer, there are n^l neurons in the $l-1$ layer, and there are $n^{(l)} \times n^{(l-1)}$ connected edges. That is, the weight matrix has $n^{(l)} \times n^{(l-1)}$ parameters. When the number of neurons increases, the parameters increase, and the time complexity of calculation increases, which greatly reduces the efficiency of training. As shown in Figure 6(b), the fully connected layer is replaced with a convolutional connection. At this time, each neuron in the l layer is only connected to a neuron in a local area window of the $l-1$ layer, forming a local connection network. The input of i neuron of l layer is defined as

$$\begin{aligned} a_i^l &= f\left(\sum_{j=1}^{m} w_j^{(l)} \cdot a_{i-j+m}^{(l-1)} + b^{(l)}\right) \\ &= f\left(w^{(l)} \cdot a_{(i-j+m)i}^{(l-1)} + b_i\right), \end{aligned} \tag{5}$$

where $w^{(l)} \in R^m$ is an M-dimensional filter.

The above formula can be simplified to:

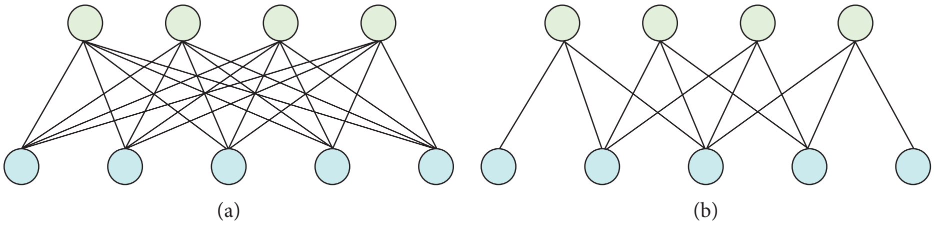


Figure 6: Full connection layer and convolution layer. (a) Full connection layer. (b) Convolution layer.

$$a^{l} = f\left(w^{l} \otimes a^{(l-1)} + b^{(l)}\right). \tag{6}$$

It can be seen from formula (6) that w^l is the same for all neurons. This reveals another extremely important feature of CNN: weight sharing. That is, for two adjacent layers of networks, the weight matrix w^l is the same. Only a few parameters are needed to describe the output from the l network to the $l+1$ layer, and the number of neurons in the $l+1$ layer is determined, which is $n^{(l+1)} = n^{(l)} - m + 1$.

When processing images, the computer cannot directly recognize the surface features of an image, like a human brain, and the computer can only accept and process the data. Therefore, a digital image can be converted into a two-dimensional matrix, and the position of each pixel is used to describe the entire image. The two-dimensional matrix of image conversion is used as the input of the neural network, and two-dimensional convolution is required at this time. Assume that $x^{(l)} \in R^{(w_1 \times h_1)}$ and $x^{(l-1)} \in R^{(w_{l-1} \times h_{l-1})}$ are the neuronal activity of l and $l+1$ layers, respectively. Each element of $X^{(l)}$ is

$$X_{s,t}^{(l)} = f\left(\sum_{i=1}^{u}\sum_{j=1}^{v} W_{i,j}^{(l)} \cdot X_{s-i+u,t-j+v}^{(l-1)} + b^{(l)}\right). \tag{7}$$

After a filter is processed, the characteristics of an image can be obtained. By increasing the number of filters used, a number of different features can be obtained, thus enhancing the ability of the convolutional layer to represent images. The filter is essentially a feature extractor. Due to the weight sharing, each set of output uses the same filter, which is the feature extractor. The output of the image processed by the filter is a feature of the image. This process can also be called feature mapping. Assume that the number of filters used in the $l-1$ layer is n_{l-1}, and the size of each group of feature maps is $m_{l-1} = w_{l-1} \times h_{l-1}$. The total number of neurons in the $l-1$ layer is $n_{l-1} \times m_{l-1}$. The number of feature mapping groups in the l layer is n_l. If it is assumed that the input of each feature map $X^{(l,k)}$ of l layer is all the feature maps of $l-1$ layer,

then k feature map $X^{(l,k)}$ of l layer is

$$X^{(l,k)} = f\left(\sum_{p=1}^{n_{l-1}} W^{(l,k,p)} \otimes X^{(l-1,p)} + b^{(l,k)}\right), \tag{8}$$

where $W(l,k,p)$ represents the filter required from the p feature vector of $l-1$ layer to the k feature vector of l layer. It can be found from the above formula that the neurons in the entire layer of the l layer get the input of the next layer, the $l+1$ layer, through filter convolution and bias adjustment. Different filters can get different inputs. The connection relationship between feature maps can be defined as a connection table T. The number of features is adjusted by setting the number of "0"s in the connection table to ensure that the desired features can be extracted and the computational complexity is reduced.

The convolutional layer is locally connected, which significantly reduces the number of connections compared to the fully connected layer, but the number of neurons does not change much. If the output is followed by a classifier, the input dimension of the classifier is still too high, overfitting will still occur, and the input image cannot be accurately classified. Pooling operation is introduced to reduce the dimensionality of features and avoid overfitting problems. The feature map $X^{(l)}$ obtained by convolution of the upper convolution layer through the filter can be divided into several regions $R_k, k = 1, \ldots, K$. To perform pooling operations on these regions, a subsampling function sub is defined as

$$X_k^{(l+1)} = f\left(w^{(l+1)} \cdot \mathrm{sub}\left(R_k\right) + b^{(l+1)}\right). \tag{9}$$

where $\overset{(l+1)}{w}$ and $b^{(l+1)}$ are trainable weight and bias parameters, respectively.

As shown in Figure 7, the LR-CNN model designed in this paper is

(1) Input layer: the input data is a $32 * 32$ image block.

(2) Con1 : the first convolutional layer, using 8 $5 * 5$ filters, through convolution operation to obtain 8 $28 * 28$ feature maps.

(3) Pool2: the first subsampling layer uses the maximum pool sampling method, that is, one point is collected for every adjacent $2 * 2$ pixel area. Its value is the function value with the largest gray value among the four pixels.

(4) Con3 : the second convolution layer, using 15 $5 * 5$ filters, after convolution operation, $8 * 15$ $10 * 10$ feature maps are obtained.

(5) Pool4: the second subsampling layer uses the same subsampling as the Pool2 layer.

(6) Con5 : the last convolutional layer, using 5 $5 * 5$ filters.

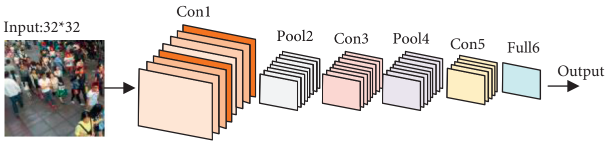


Figure 7: LR-CNN counting model.

(7) Ful6: it is a fully connected layer that converts 600 $1 * 1$ neurons in Cons into a feature vector.

(8) Output layer: input the feature vector obtained by CNN into the activation function to obtain the counting result. When training the model, it is also necessary to add the counting accuracy rate to estimate the accuracy rate of the counting and loss function layers.

The LR-CNN model proposed in this paper reduces the number of neurons when extracting features based on the same filter in the convolutional layer. However, since each convolutional layer requires multiple filters, different features of the image need to be extracted. Therefore, the total number of neurons is significantly increased after the convolution operation of the convolutional layer, and the purpose of convolution is to reduce the dimension of features. However, additional burdens are generated in this process. The layer-by-layer increase in the number of neurons and parameters will eventually cause the algorithm to crash and the computer will stop working. Thus, the subsampling layer is necessary, and it is an effective means to reduce the number of neurons and the number of parameters. Therefore, the subsampling layer must exist intermittently or uninterruptedly throughout the entire network. At the same time, it is considered that the influence of the subsampling layer on the feature is negative. Therefore, the alternate appearance of the convolutional layer and subsampling layer is the best design obtained by combining various factors.

4.2. Loss Function. This paper uses two loss functions to optimize the model. One is the Euclidean loss function, and the other is the cross-entropy loss function. Let $X = \{X_1, \ldots, X_N\}$ denote training samples and N denote the total number of training samples.

Euclidean loss function is used for density estimation

$$L_e = \frac{1}{2N}\sum_{i=1}^{N}\left\| F_h\left(X_i;\theta\right) - D_i \right\|_2^2, \tag{10}$$

where $F_h(\ ;\theta)$ represents the estimated density map. θ is the weight parameter of the counting model and D_i represents the true density map.

The cross-entropy loss function is

$$L_c = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}\left[\left(y^i = j\right)F_C\left(X_i;\theta_c\right)\right]. \tag{11}$$

The total loss function is a linear combination of L_e and L_c, and the formula is as follows:

$$L = L_e + \alpha L_c, \tag{12}$$

where the parameter α is a scale factor, which is used to control the proportion of cross-entropy loss.

5. Experiment and Analysis

5.1. Dataset. The experiment uses two commonly used datasets, namely the Shanghai Tech dataset and the UCF_CC_50 dataset.

The Shanghai Tech dataset consists of two parts, part_A_final and part_B_final. The picture of part_A is a crowd image randomly selected from the Internet, and the data picture of part_B is taken by a camera on the streets of Shanghai. Compared with the part_A dataset, part_B has a sparse distribution, but the scene is relatively fixed, while the scene of part_A changes greatly. part_A training set: 300 pictures, test set: 182 pictures. part_A training set: 400 pictures, test set: 316 pictures, a total of 1198 pictures, 330,165 annotation headers.

The UCF_CC_50 dataset pictures are all grayscale images downloaded from the Internet. They have extremely dense crowds and smallscale changes. Large amounts of data only have head features and are severely blocked by pedestrians. The sample size of this dataset is small, but the number of people varies greatly. In the experiment, a 5-fold crossvalidation method was used to evaluate the performance of different counting models. The specific method is to randomly divide the picture into 5 parts, with 4 parts for training and 1 part for testing. Five sets of experiments are carried out, and the average value is taken as the final result.

5.2. Evaluation Index. This paper uses MAE and MSE as two indicators to evaluate the performance of the algorithm. MAE and MSE are the most commonly used standards to measure the performance of the algorithm. The calculation formula of MAE and MSE is as follows:

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{N}\sum_{1}^{N}\left|y_i - \tilde{y}_i\right|, \\ \mathrm{MSE} &= \sqrt{\frac{1}{N}\sum_{1}^{N}\left|y_i - \tilde{y}_i\right|^2}, \end{aligned} \tag{13}$$

where N represents the total number of test images, y_i is the actual number of people in the i image, and $\tilde{y}_i$ is the number of people estimated by the i algorithm.

5.3. Analysis and Comparison. In the use of the CLBP feature extraction algorithm and the CNN depth model for crowd density estimation and group detection, this paper has carried out 2000 iterations of training. In order to visualize the results, when the CNN network becomes stable, the 200 verified samples are extracted from the CLBP feature and then input to the trained CNN network. Figure 8 shows the comparison between the real predicted value and the CNN predicted value.

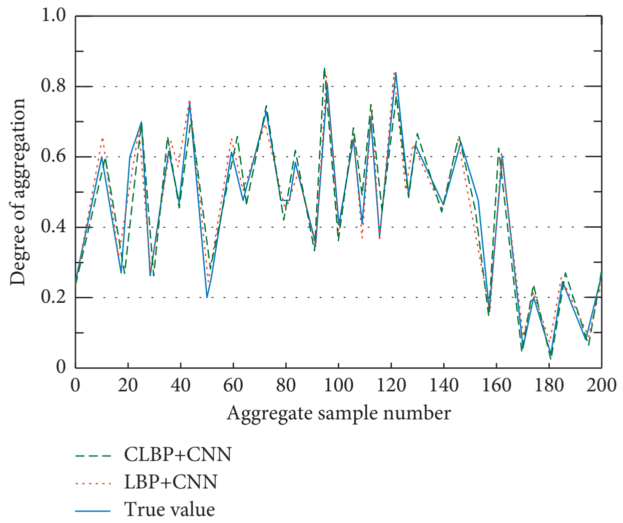


FIGURE 8: Comparison between output values and actual output values of different methods.

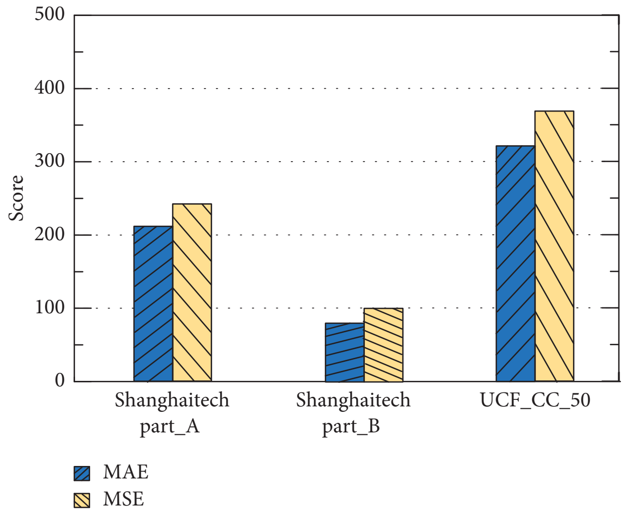


FIGURE 9: Test results of counting model on each dataset.

After deep neural network training, the predicted value of the degree of aggregation of each pixel position is obtained. In the actual prediction, the mask value of prediction results 200 ∗ 200 is tested in the range of 10 ∗ 10, and the average value in the range is calculated, and the threshold Th = 0.5 is set as the criterion. When the predicted average value of a certain detection area reaches or exceeds the set threshold, the area is regarded as an area where people gather. And through the inverse process of the compression process, the position is projected into the original RGB image, and the corresponding area is standardized in the figure. This paper tests CLBP + CNN and LBP + CNN. It can be seen from the results that CNN can do most of the correct detection of crowd gathering groups. The comparison between the predicted value and the actual predicted value is almost the same. In actual use, there is a strong result presentation that can ensure the robustness and accuracy of the data. However, the CNN network requires a lot of training time to obtain better weights to predict complex scenes.

The CNN counting model was tested on the UCF_CC_50 and Shanghai Tech datasets, and the results obtained are shown in Figure 9.

The method in this paper is compared with the methods in reference [5, 27, and 29] in the UCF_CC_50 dataset and the Shanghai Tech dataset. The experimental results are shown in Table 1 and Table 2. The MAE and MSE of the proposed method in the UCF_CC_50 dataset is 325.6 and 369.4, respectively. The MAE and MSE in the part_A part of the Shanghai Tech dataset are 213.5 and 247.1, respectively, and the MAE and MSE in the part_B part are 85.3 and 99.7 respectively. The experimental results show that the method proposed in this paper can solve the problem of counting dense crowds within the allowable error range. The comparison results show that the proposed method is better than the comparison method in counting accuracy under high crowd density scenarios. This is because the proposed model extracts the difference between the amplitude feature and the gray feature of the central pixel to form the CLBP feature, which obtains more detailed information about the population density. However, the lack of effective feature extraction methods in comparison methods makes MAE and MSE much higher than the proposed methods. Besides, using the dense block in the image as a training set instead of the entire image provides a feasible method to solve the counting problems caused by crowd image congestion and scene distortion.

TABLE 1: Comparison with other algorithms on the UCF_ CC_ 50 dataset.

Method	MAE	MSE
Reference [5]	456.5	489.7
Reference [29]	403.7	455.9
Reference [27]	357.4	378.1
The proposed method	325.6	369.4

TABLE 2: Comparison with other algorithms on the Shanghai Tech dataset.

Method	Part_A		Part_B	
	MAE	MSE	MAE	MSE
Reference [5]	335.4	387.9	157.3	187.9
Reference [29]	289.6	325.4	102.8	125.6
Reference [27]	256.3	289.7	95.4	108.5
The proposed method	213.5	247.1	85.3	99.7

6. Conclusion

Aiming at the problems of crowd distribution, scale feature, and crowd feature extraction difficulty in exhibition centers, this paper proposes a crowd density estimation method using deep learning for passenger flow detection systems in the exhibition center. The difference between the amplitude feature and the gray level feature of the center pixel are extracted to form the CLBP feature together to obtain more descriptive information about the crowd density. The LR activation function is used to add nonlinear factors to CNN and use dense blocks obtained by crowd density estimation to train the LR-CNN crowd density estimation model. Finally, the experimental results show that the proposed method can achieve the lowest MAE and MSE on the tested datasets. This shows that by extracting the difference between the amplitude feature and the gray feature of the center pixel, using the CLBP feature for feature extraction, you can extract more effective information.

However, deep learning has a complex network structure and requires a large amount of calculation, which requires faster hardware support. In the future, GPUs can be introduced to increase the speed of computer processing data, or the concept of parallel computing can be introduced into CNN, and the execution speed of algorithms can be accelerated by shunting.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

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

Acknowledgments

This work was supported by the Yongchuan District Science and Technology Fund (no. Ycstc, 2019rb1202).

References

[1] S. A. M. Saleh, S. A. Suandi, and H. Ibrahim, "Recent survey on crowd density estimation and counting for visual surveillance," *Engineering Applications of Artificial Intelligence*, vol. 41, no. 5, pp. 103–114, 2015.

[2] V. A. Sindagi and V. M. Patel, "A survey of recent advances in CNN-based single image crowd counting and density estimation," *Pattern Recognition Letters*, vol. 107, no. 7, pp. 3–16, 2018.

[3] S. Elbishlawi, M. H. Abdelpakey, A. Eltantawy, M. S. Shehata, and M. M. Mohamed, "Deep learning-based crowd scene analysis survey," *Journal of Imaging*, vol. 6, no. 9, pp. 95–104, 2020.

[4] X. Zhang, Q. Yu, and H. Yu, "Physics inspired methods for crowd video surveillance and analysis: a survey," *IEEE Access*, vol. 6, no. 2, pp. 66816–66830, 2018.

[5] Z. Fan, Y. Zhu, Y. Song, and Z. Liu, "Generating high quality crowd density map based on perceptual loss," *Applied Intelligence*, vol. 50, no. 4, pp. 1073–1085, 2020.

[6] S. Pu, T. Song, Y. Zhang, and D. Xie, "Estimation of crowd density in surveillance scenes based on deep convolutional neural network," *Procedia Computer Science*, vol. 111, no. 5, pp. 154–159, 2017.

[7] V. J. Kok and C. S. Chan, "Granular-based dense crowd density estimation," *Multimedia Tools and Applications*, vol. 77, no. 15, Article ID 20227, 2018.

[8] H. Fradi and J. L. Dugelay, "Crowd density map estimation based on feature tracks," in *Proceedings of the 2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 040–045, IEEE, Pula, Italy, September 2013.

[9] M. V. Anees and S. G. Kumar, "Deep learning framework for density estimation of crowd videos," in *Proceedings of the 2018 8th International Symposium on Embedded Computing and System Design (ISED)*, pp. 16–20, IEEE, Cochin, India, December 2018.

[10] B. Yılmaz, S. N. H. S. Abdullah, and V. J. Kok, "Vanishing region loss for crowd density estimation," *Pattern Recognition Letters*, vol. 138, no. 7, pp. 336–345, 2020.

[11] H. Zheng, Z. Lin, J. Cen, Z. Wu, and Y. Zhao, "Cross-line pedestrian counting based on spatially-consistent two-stage local crowd density estimation and accumulation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 3, pp. 787–799, 2018.

[12] W. Yanqin, Y. Zujun, W. Yao, and L. Xingxin, "Crowd density estimation based on conditional random field and convolutional neural networks," in *Proceedings of the 2019 14th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, pp. 1814–1819, IEEE, Changsha, China, November 2019.

[13] J. Guo, X. Wu, T. Cao, S. Yu, and Y. Xu, "Crowd density estimation via markov random field (MRF," in *Proceedings of the 2010 8th World Congress on Intelligent Control and Automation*, pp. 258–263, IEEE, Jinan, China, July 2010.

[14] S. Peng, B. Yin, X. Hao, Q. Yang, A. Kumar, and L. Wang, "Depth and edge auxiliary learning for still image crowd density estimation," *Pattern Analysis & Applications*, vol. 24, no. 4, pp. 1777–1792, 2021.

[15] Y. Li and B. Zhou, "A hybrid approach to crowd density estimation using statistical leaning and texture classification," in *Proceedings of the 2013 International Conference on Optical Instruments and Technology: Optoelectronic Imaging and Processing Technology*, pp. 9045–9550, International Society for Optics and Photonics, Beijing, China, November 2013.

[16] H. Jiang and W. Jin, "Effective use of convolutional neural networks and diverse deep supervision for better crowd counting," *Applied Intelligence*, vol. 49, no. 7, pp. 2415–2433, 2019.

[17] H. Fradi and J.-L. Dugelay, "Towards crowd density-aware video surveillance applications," *Information Fusion*, vol. 24, no. 3, pp. 3–15, 2015.

[18] Z. Duan, Y. Xie, and J. Deng, "HAGN: hierarchical attention guided network for crowd counting," *IEEE Access*, vol. 8, no. 9, Article ID 36376, 2020.

[19] M. Fu, P. Xu, X. Li, Q. Liu, M. Ye, and C. Zhu, "Fast crowd density estimation with convolutional neural networks," *Engineering Applications of Artificial Intelligence*, vol. 43, no. 5, pp. 81–88, 2015.

[20] L. G. Zhu, H. Zhang, S. Ali, B. L. Yang, and C. Y. Li, "Crowd counting via multi-scale Adversarial convolutional neural networks," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 705–715, 2020.

[21] Y. Hu, H. Chang, F. Nian, Y. Wang, and T. Li, "Dense crowd counting from still images with convolutional neural networks," *Journal of Visual Communication and Image Representation*, vol. 38, no. 6, pp. 530–539, 2016.

[22] 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, NV, USA, June 2016.

[23] D. Kang and A. Chan, "Crowd counting by adaptively fusing predictions from an image pyramid," pp. 569–578, 2018, https://www.researchgate.net/publication/325191555_Crowd_Counting_by_Adaptively_Fusing_Predictions_from_an_Image_Pyramid.

[24] J. Liu, C. Gao, D. Meng, and A. G. Hauptmann, “Decidenet: counting varying density crowds through attention guided detection and density estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5197–5206, Salt Lake City, UT, USA, June 2018.

[25] X. H. Jiang, L. Zhang, M. L. Xu et al., “Attention scaling for crowd counting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4706–4715, Seattle, WA, USA, June 2020.

[26] X. Liu, J. Van De Weijer, and A. D. Bagdanov, “Leveraging unlabeled data for crowd counting by learning to rank,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7661–7669, Salt Lake City, UT, USA, June 2018.

[27] S. Li, Z. Hu, M. Zhao, and Z. Sun, “Crowd counting using cross-adversarial loss and global feature,” *Journal of Electronic Imaging*, vol. 29, no. 5, pp. 53–59, 2020.

[28] H. Nan, M. Tong, L. Fan, and L. I. Min, “Multi-task multi-level convolutional neural network for application of crowd counting,” *Computer Engineering and Applications*, vol. 5, no. 9, pp. 506–514, 2019.

[29] Y. Zhang, H. Zhao, Z. Duan, L. Huang, J. Deng, and Q. Zhang, “Congested crowd counting via adaptive multi-scale context learning,” *Sensors*, vol. 21, no. 11, pp. 3777–3785, 2021.