Hindawi
Complexity
Volume 2022, Article ID 8213855, 10 pages
https://doi.org/10.1155/2022/8213855

Research Article

An Improved Multibranch Convolutional Neural Network with a Compensator for Crowd Counting

Zhiyun Zheng, Zhenhao Sun, Guanglei Zhu, Zhenfei Wang, and Junfeng Wang

School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China

Correspondence should be addressed to Junfeng Wang; iewangjf@zzu.edu.cn

Received 28 May 2021; Revised 25 February 2022; Accepted 14 March 2022; Published 28 April 2022

Academic Editor: Jenq-Haur Wang

Image-based crowd counting has extremely important applications in public safety issues. Most of the previous studies focused on extremely dense crowds. However, as the number of webcams increases, a crowd with extremely high density can obtain less error by summing the images of multiple close-range webcams, but there are still some problems such as heavy occlusions and large-scale variation. To solve the above problems, this paper proposes a new type of multibranch neural network with a compensator, in which features are extracted through multibranch subnetworks of different scales. The weights between the branches are adjusted by the compensator, and the captured features are distinguished among different branches. To avoid learning nearly the same features in each branch and reducing the training deviation, the dataset is labeled with head scale, and the adaptive grading loss function is used to calculate the estimated loss of the subregions. The experimental results show that the accuracy of the network proposed in this paper is about 10% higher than that of the comparison network.

1. Introduction

In recent years, the crowd counting based on computer vision has been widely used in video surveillance, traffic control, security services, etc., which has attracted great attention from people. At present, the solution of crowd counting has gradually developed from target detection to display density distribution, and the total number is given by integrating the density map. Inspired by the great success of convolutional neural networks in computer vision tasks, such as object detection[1, 2], image segmentation [3, 4], and object tracking [5, 6], people began to pay attention to the application of convolutional neural networks method in crowd counting.

Due to the challenges of heavy occlusions, large-scale variation, perspective effect, and large density differences in crowd counting [7], crowd counting faces great difficulties, especially for the overconcentration of crowd distribution caused by perspective effect, which seriously affects the crowd distribution in the image. The number of people owned in a smaller area accounts for a larger proportion of the number of people owned by the entire picture. As shown in Figure 1, human head targets are mainly concentrated in a small area in the middle of the image, while the total number of human head targets in the larger area below the image is significantly smaller than that in the middle area of the image. It is obvious that, compared with the crowd counting error produced in the area below the image, the error produced by the crowd counting in the middle area of the image has a more significant impact on the final evaluation index of the whole image. Therefore, the current focus of reducing crowd counting error is mainly to improve the accuracy of crowd counting in extremely dense areas.

An area with a large population density in the image also means that the head target is extremely small. In an image with a large number of people, it is often only necessary to successfully capture the head target features with medium and below target scales to get a good counting result, but this also leads to the neglect of large-scale human head targets. Moreover, the number of features that can be obtained from a single head target with a very small scale is relatively small, and overfitting of such datasets will reduce the generalization ability of the model, making it difficult to be applied in actual scenarios.

Figure 1: Image of people with large density differences.

This is also one of the problems in current population counting.

With the increasing demand in reality, especially for semi-enclosed areas with high crowd density and danger, such as stadiums and theatres, the number of webcams gradually increases; in the meantime, the distance to the crowd is closer; then, the corresponding scale of nearby human head target also follows. By increasing the webcams, the distance between the webcams and the crowd is shortened, and thus the scene of great crowd density is transformed into the sum of the crowd count in multiple regions, making the result more accurate. However, the problem of population scale variation still exists. Even a close-up of the camera will cause an increase in the nearby human head target, leading to the greater variation in the human head target. Therefore, crowd counting with large-scale variation is of great value in real applications.

The head-scale information of the image is not marked in the head data annotation. Since the convolutional neural network relies heavily on the information provided by the dataset during the training process, the lack of scale annotation information will also affect the results of network learning. Therefore, this article adds grading labeling of head scales in the dataset to increase the missing head-scale information in the dataset; then, according to the scale information, the adaptive grading loss function is used. According to the head size information in the image blocks, the final loss is obtained by calculating the local loss and accumulating the loss of all blocks. Moreover, the loss of all blocks is accumulated, and then the final loss is obtained.

Aiming at the large-scale change of crowd counting, this paper proposes a multibranch convolutional neural network with a compensator. The multibranch structure can effectively solve the multiscale feature problem, and the compensator can perform weight compensation on the outputs of different branches. Different branches use convolution kernels of different scales. For different scale features, branches of different receptive fields can be captured correspondingly, and the performance at different branches is optimized at the same time. The network also uses the adaptive grading loss function and adaptively uses the asynchronous loss function according to the target scale. The head-scale grading label is manually added to the dataset, which preserves the head-scale information in the image to a certain extent. Finally, a comparative experiment is conducted on the dataset with a large variation in feature scale. Compared with the comparison network, the population counting accuracy of this network in the dataset is improved.

2. Related Work

Crowd counting is mainly divided into two categories, namely, detection-based methods and regression-based methods. Detection-based methods [8, 9] have good results when the crowd is sparse and the occlusion is not heavy. However, it is often difficult to obtain good crowd counting based on the detection in complex actual scenes, such as heavy occlusion and complex background. The regression-based method calculates the number of people by learning the mapping relationship between image features and density maps. However, this method ignores the spatial information in the image and cannot intuitively feel where the crowd gathers. Recently, with the great success of convolutional neural networks (CNNs), researchers in this field have focused on training CNNs-based models to generate high-quality density maps and thus improving counting performance [10, 11]. The practice has proved that the use of convolutional neural networks for crowd counting is very effective, but early models are affected by scale variation, which usually leads to a decrease in inaccuracy. For this reason, some scholars have proposed a series of multibranch neural networks to extract multiscale features, aiming to extract features of different scales by using convolution kernels of different scales to deal with the problem of large-scale variation. For example, the multicolumn convolutional neural network (MCNN) proposed by Zhang et al. [12] used multi-size filters to extract features with different sizes and finally integrated these features into the same density map. Similarly, Sam et al. [13] proposed the Switch-CNN which used a switch classifier to select the best classifier from a pool of density generators. Subsequently, in order to make the network have certain specialties, scholars began to optimize the network in a targeted manner, so that the network has the characteristic ability to solve some certain problems. For example, Sindagi and Patel [14] proposed a multilevel bottom-top and top-bottom fusion network (MBTTBF), which was carefully designed to combine multiple shallow and deep features. Chen et al. [15] proposed a scale pyramid network (SPN), which extracted multiscale features in parallel by using the dilated convolution of different dilation rates in a shared single-column CNN. A scale-based attention model was proposed to

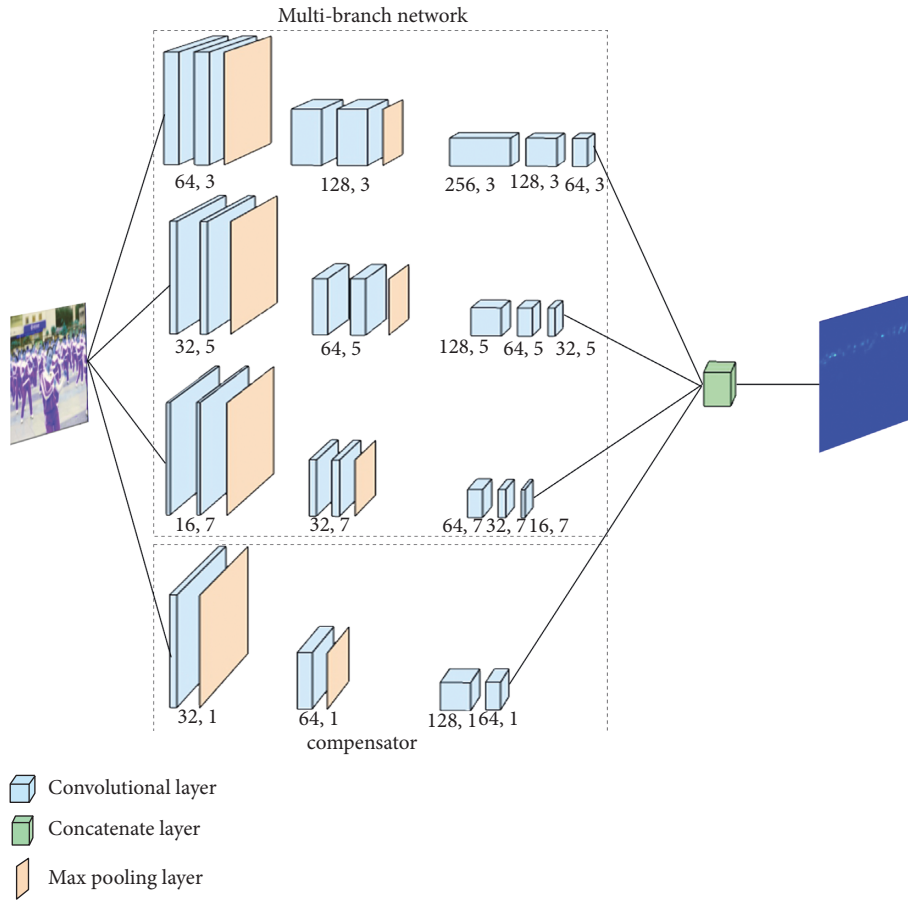


Figure 2: The overall architecture of the multibranch neural network with a compensator.

adaptively select the appropriate head size and shape [16, 17]. Wang [18] et al. proposed a combination of dilated convolutions with multiple dilation rates to process population characteristics of different scales using dilated convolutions to capture larger-scale targets, which has the advantage of fewer parameters.

Although these methods have made effective improvements, there are still some limitations:

(1) Multiple branches of exactly the same length will learn almost the same characteristics, which deviates from the original purpose of multibranch design.

(2) The dilated convolution itself has a strong ability to capture large-scale features, but it has a poor ability to capture small-scale features. Using dilated convolution requires a more complex network structure to ensure that the network can simultaneously capture features at different scales.

(3) The network structure is overly complex, and the network level is deep, so it is difficult to generalize well without a large enough dataset.

Aiming at the above problems, this paper proposes a multibranch neural network with a compensator.

(1) We used a multibranch network to solve multiscale problems; at the same time, we increased the compensator module, adjusted the weights between the multibranch structures, and increased the degree of difference in the learned features of each branch.

(2) The dataset is added with scale level information, and the head features are divided into 4 levels according to the scale to facilitate the network to learn.

(3) In view of the large difference in target scales, a hierarchical loss function based on target scales was proposed.

In response to the above problems, this paper proposes a multibranch neural network with a compensator. The network adjusts the weights among the branches through the compensator and increases the differences in learning characteristics of each branch. The shallow structure ensures the generalization ability of the network.

3. The Proposed Method

3.1. Multibranch Convolutional Neural Network. In order to solve the problems of the huge variation in the scale of crowd counting and similar extraction features of each branch in the multibranch structure, a multibranch neural network with a compensator is proposed. The overall architecture is shown in Figure 2. The network consists of an

input, an output, a multibranch neural network, and a compensator.

The input part can receive input images of different pixels, but in order to prevent the image from being too large, the image larger than 224 pixels is reduced. Because this dataset needs to be reduced in a small proportion, it will not cause serious loss of precision due to the reduction, which will affect the crowd counting precision. The output part is connected to the output of the multibranch neural network; the compensator module and the outputs of different branches enter the output part together; then, the output part generates a density map showing the distribution of the crowd. The multibranch neural network part is a multiscale feature extraction module composed of three branches. A network which is excessively deep may weaken the generalization ability. However, the 3×3 convolution kernel, which has a better effect in extracting image features, is difficult to perform well when extracting large-scale features in a shallower network due to insufficient receptive fields. Based on this consideration, this paper adds branches of large-scale convolution kernels while using a shallower network. Finally, a network structure composed of three 7-layer branches is used to extract features of the image. The three branches in Figure 2 use 3×3, 5×5, and 7×7 convolution kernels from the top to the bottom. In order to optimize the training efficiency, the number of convolution kernels decreases by half each time as the scale of the convolution kernel increases.

By calculating the receptive field and comparing the receptive field with the image scale, it can be seen that the large-scale convolution kernel plays an important role in capturing large-scale objects. The receptive field size of the first branch is calculated as follows:

$$RF_i = (RF_{i-1} - 1)\text{Stride}_i + \text{KSize}_i, \tag{1}$$

where RF is the receptive field, i is the number of layers, Stride is the step size, and KSize is the size of the convolution kernel.

By bringing in the common influence of the convolution kernel and the pooling layer on the receptive field, the receptive field of the first branch above the network is 40×40. Similarly, the receptive fields of the second and third branches are calculated to be 76×76 and 112×112, respectively. As can be seen from Figure 3, the receptive field of 112×112 can already read a quarter of the input image with pixels of 224×224, which is enough to cope with scale changes in most cases. Even if there are features that exceed this scale, the number of them is less than 4, which has little impact on the counting results, but a larger number of parameters of the convolution kernel have a negative impact on counting accuracy, network training, network generalization, etc. Therefore, larger convolution kernels are not used.

4. Branch Weight Compensation

The bottom of Figure 2 is the compensator module, which compensates for the results of other branches through image

Figure 3: Effect image of receptive field.

features and optimizes the final result. After experimental tests, the network itself shows the ability to handle complex scenes; the excessively complex compensator module makes it difficult for the model to converge and to achieve good results. The simpler compensator module can actually improve the network's ability to extract simple features.

The compensator module is implemented by using a single branch, which allows the network to adaptively learn the output weights of other branches by grasping the image input characteristics. After testing, the network will be used to learn feature information, even with a very small number of 3×3 convolution kernels, which defeats the original purpose of designing the compensator. Therefore, the compensator is only composed of a 1×1 filter and a max pooling layer, which is used to organize the feature information of the image and perform weight compensation on the first three branches, so as to optimize the weights of different branches under different scale features.

The three convolution operations ensure its nonlinear structure, and the two max pooling operations can first keep the same size as the output of other branches, that is, a quarter of the input image. At the same time, the receptive field of the 1×1 convolution kernel can be increased to 4×4, which helps the branch to better grasp the characteristics of the input image while keeping fewer parameters.

The outputs of the three branches of feature extraction are Y_i, Y_j, Y_k ($i \in \{1, 2, \ldots, 64\}$, $j \in \{1, 2, \ldots, 32\}$, $k \in \{1, 2, \ldots, 16\}$), ($i \in \{1, 2, \ldots, 64\}$, $j \in \{1, 2, \ldots, 32\}$, $k \in \{1, 2, \ldots, 16\}$); the output of the compensator is c_t ($t \in \{1, 2, \ldots, 128\}$). The weight when they perform the first convolution operation is ω_{mn} ($m \in \{1, 2, \ldots, 64\}$, $n \in \{1, 2, \ldots, 200\}$), $\sum \omega(c + Y_i + Y_j + Y_k)$. After the second convolution operation, the weight is δ_m, and the final weight compensation formula is calculated as follows:

$$M = \sum_{m=1}^{64} \delta_m \left[\sum \omega_{m,n} \left(c + Y_i + Y_j + Y_k \right) \right], \tag{2}$$

where i represents different branches, j represents the number of filters, Y represents the value of the branch output, and ω represents the weight corresponding to its Y.

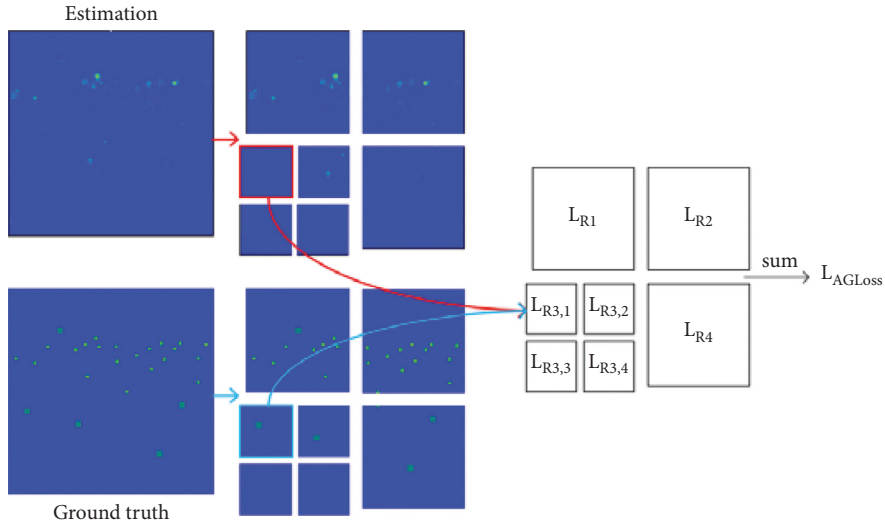


Figure 4: Demonstration of two-level adaptive classification loss.

By compensating for the weights, the output of each branch that had been originally fixed can have a certain choice space, which helps the multibranch structure to have more free choices when learning different features. To a certain extent, we avoid all branches trying to learn exactly the same characteristics, thus making differences between branches.

5. Adaptive Grading Loss Function

In the training phase, the previous CNN-based density estimation network usually uses the Euclidean distance between the entire estimated density map and the ground truth density map as the loss function [19], as shown in the following formula:

$$L(\theta) = \frac{1}{M}\sum_{k=1}^{M}\left\|D\left(X^k;\theta\right) - D^k\right\|_2^2, \tag{3}$$

where X^k is the k-th input image, D^k is its ground truth density map, θ is the parameter of the counting network, $D(X^k;\theta)$ is the estimated density map, and M is the size of the training set.

This loss function ignores the influence of feature information of different scales on the network training process. Since different scale features are often required to feed back different size steps, this loss function cannot meet the asynchronous requirements of backpropagation of different scale features, which weakens the learning ability of the network when training different scale features. In order to solve the above problems, the loss function is optimized by making full use of the dataset with scale information. For different crowd densities, an adaptive hierarchical loss function (AGLoss) [20] is proposed with reference to Adaptive Pyramid Loss (APLoss). AGLoss can adaptively divide the density map into subregions at different levels according to the real head scale. Then, the AGLoss computes the relative estimated loss for each part and adds it to get the final loss. The data is preprocessed hierarchically using the same structure as the 3×3 branch of the network so that subsequent networks use different loss functions.

Table 1: Specification of the employed dataset. Num: the number of images; max: the maximal crowd count; min: the minimal crowd count; ave: the average crowd count; total: the total number of labeled people.

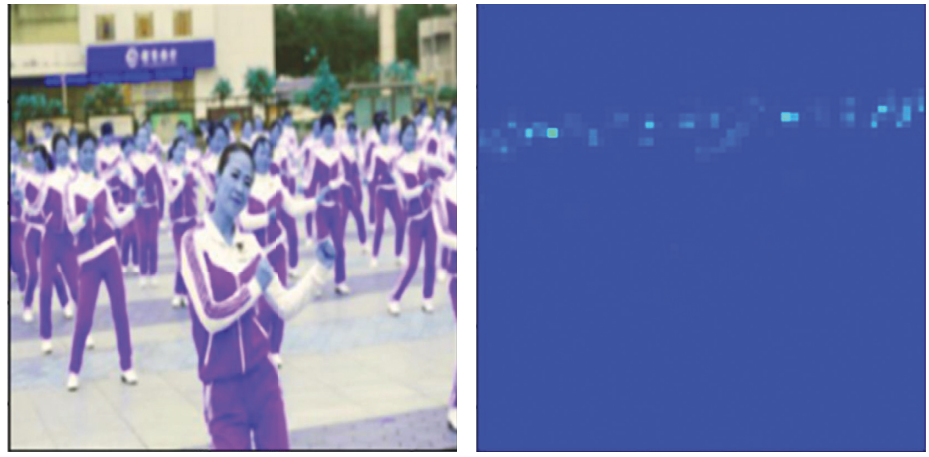
Num	Resolution	Max	Min	Ave	Total
503	Different	78	2	25.9	13028

Table 2: Performance comparison of different methods on dataset.

Method	MAE	MSE
MSCNN [21]	12.6	368.9
MCNN [12]	8.8	350.8
DSA-CNN [22]	8.2	306.8
DCN [23]	8.1	311.3
Ours	7.3	302.4

Specifically, the AGLoss is calculated in the following way. First, the real population density map D^k is divided into a 2×2 first-level grid, and R_{i_1} is used to represent the subregions, where $i_1 \in \{1,2,3,4\}$. If the local population feature scale of the subregion R_{i_1} is greater than the given threshold T_{i_1}, it is divided into a 2×2 secondary subgrid. Figure 4 shows the two-stage AGLoss calculation. The human head feature scales in the secondary subgrids are all greater than T_{i_2}, then secondary subgrids are iteratively divided into 2×2 three-level subgrids until the feature scale of this area is less than T_{i_n}. Let $R_{i_1,\ldots,i_n}$ denote the in-layer subregion, $i_n \in \{1,2,3,4\}$. After the segmentation is completed, an uneven grading grid can be obtained. Apply the obtained adaptive grading grid to the estimated density map $D(X^k;\theta)$ and calculate the local loss of each subregion based on the following equation:

Ground truth: 26 Estimation: 24



Ground truth: 25 Estimation: 28

Figure 5: People with large-scale variation.

$$l^k_{R_{i_1,\ldots,i_n}} = \begin{cases} \dfrac{\left\| D_{R_{i_1,\ldots,i_n}}\left(X^k;\theta\right) - D^k_{R_{i_1,\ldots,i_n}} \right\|_2^2}{\left(1/\max\left(D^k_{R_{i_1,\ldots,i_n}}\right)\right)+1}, & \max\left(D^k_{R_{i_1,\ldots,i_n}}\right) < T_{i_n}, \\ \sum_{i_n=1}^{4} l^k_{R_{i_1,\ldots,i_n}}, & \text{otherwise.} \end{cases} \tag{4}$$

Finally, sum up all local losses to obtain the final AGLoss according to the following formula:

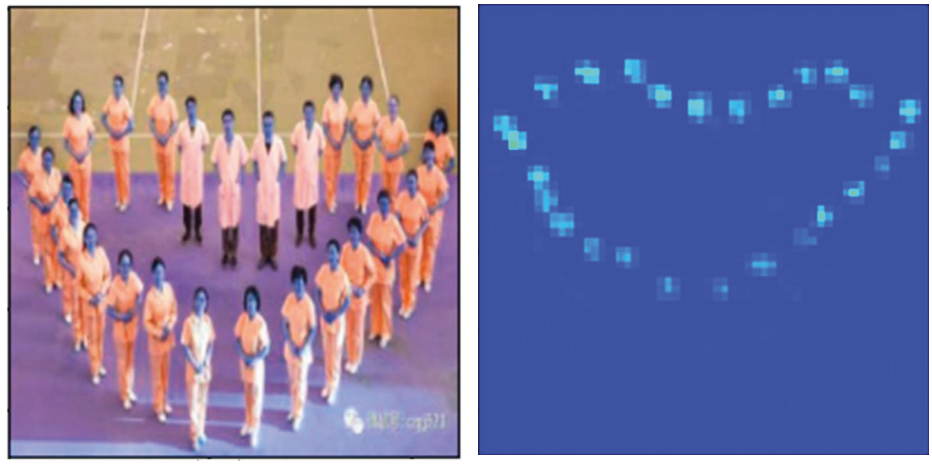
$$L_{\text{AGLoss}} = \frac{1}{M}\sum_{k=1}^{M}\sum_{i_n=1}^{4} l^k_{R_{i_n}}. \tag{5}$$

6. Case Study

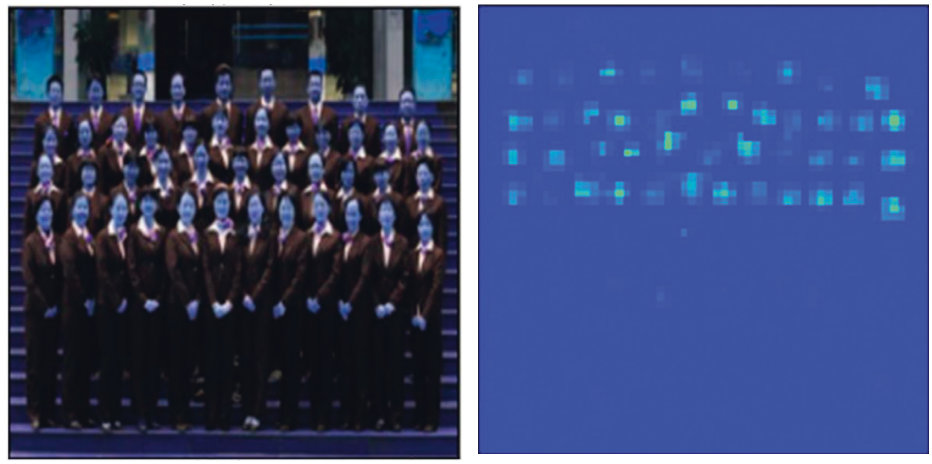
6.1. Data Description. In order to obtain data of scenes with moderate density and large variation in the crowd scale, a dataset with large variation in head size and different shooting distances was selected from the public crowd counting competition dataset. The usual data preprocessing method is Gaussian blur processing for each head feature, while all heads use the same size Gaussian kernel processing, and all head features will be treated as the same scale feature during crowd counting feedback, which makes the image originally containing the head size information completely lost during backpropagation. However, it takes too much manpower to scale markings for all the heads precisely. In order to solve the above problems, human head features are scaled, so the human head is divided into four levels by the length of the pixels: greater than 112, greater than 56 and less than 112, greater than 28 and less than 56, and less than 28. Different scale blur processing was used for them.

The scenes contained in the images of the dataset are quite different, the distortion degree of the images is different, and the scale of the human head varies obviously. The details of the data are shown in Table 1. In this case, it is already very challenging to extract head features, and the total number of marked heads is small, which puts forward high requirements for the generalization ability of the network. The dataset is divided into the training set, the validation set, and the test set on a scale of 6 : 2:2.

6.2. Performance Assessment. In this paper, two metrics, namely, mean absolute error (MAE) and mean square error (MSE), are used to evaluate the performance of all considered methods.



Ground truth: 26 Estimation: 25



Ground truth: 44 Estimation: 44

Figure 6: People with the same scale.

$$\mathrm{MAE} = \frac{1}{M}\sum_{i=1}^{M}\left|C_i - \widehat{C}_i\right|, \tag{6}$$

$$\mathrm{MSE} = \frac{1}{M}\sum_{i=1}^{M}\left(C_i - \widehat{C}_i\right)^2, \tag{7}$$

where M is the number of test images, and C_i and $\widehat{C}_i$ are the actual number of people and the estimated number of the i-th image, respectively.

6.3. Experimental Results and Discussion. First, three branches of extracted features are pretrained, and other parts of the entire network are pretrained without changing the pretraining weights, so as to avoid the deviation between the parameters of the pretrained branches and those generated by the other parts. The direction of feature extraction and analysis changes. Finally, the entire pretrained network is trained again to obtain the final training model. The batch size used is 8, the learning rate is 0.0003, and the weight attenuation is 1e-5 (Adam optimizer).

MSCNN [21], MCNN [12], DSA-CNN [22], and DCN [23], were reproduced on the dataset, compared with the multibranch neural network with the compensator to evaluate the experimental results. MSCNN repeatedly passes the input image through a single convolution kernel of different scales and then merges the output results. The interactive use of convolution kernels of different scales can combine many situations of receptive fields to solve the problem of scale change. MCNN also adopts a multibranch structure. Different branches use a larger convolution sum firstly and then use three relatively small convolution kernels. The convolution kernel scales between different branches differ by 2. MCNN is a classic method, and the model is simple and efficient. DSA-CNN adds a 1×1 convolution kernel before the convolution kernels of different scales to form a DSAM, which is used repeatedly in the network to read multiscale features. DCN uses a dual-branch structure in the first half of the network: the first half uses a large-scale convolution kernel for large-scale feature extraction, and the second half outputs the results. The calculation results of different methods are shown in Table 2.

As shown in Table 2, the multibranch neural network with the compensator achieves an accuracy of 7.3 MAE. Because of the perspective effect, the proportion of data with smaller head size in the dataset is larger. Crowd counting

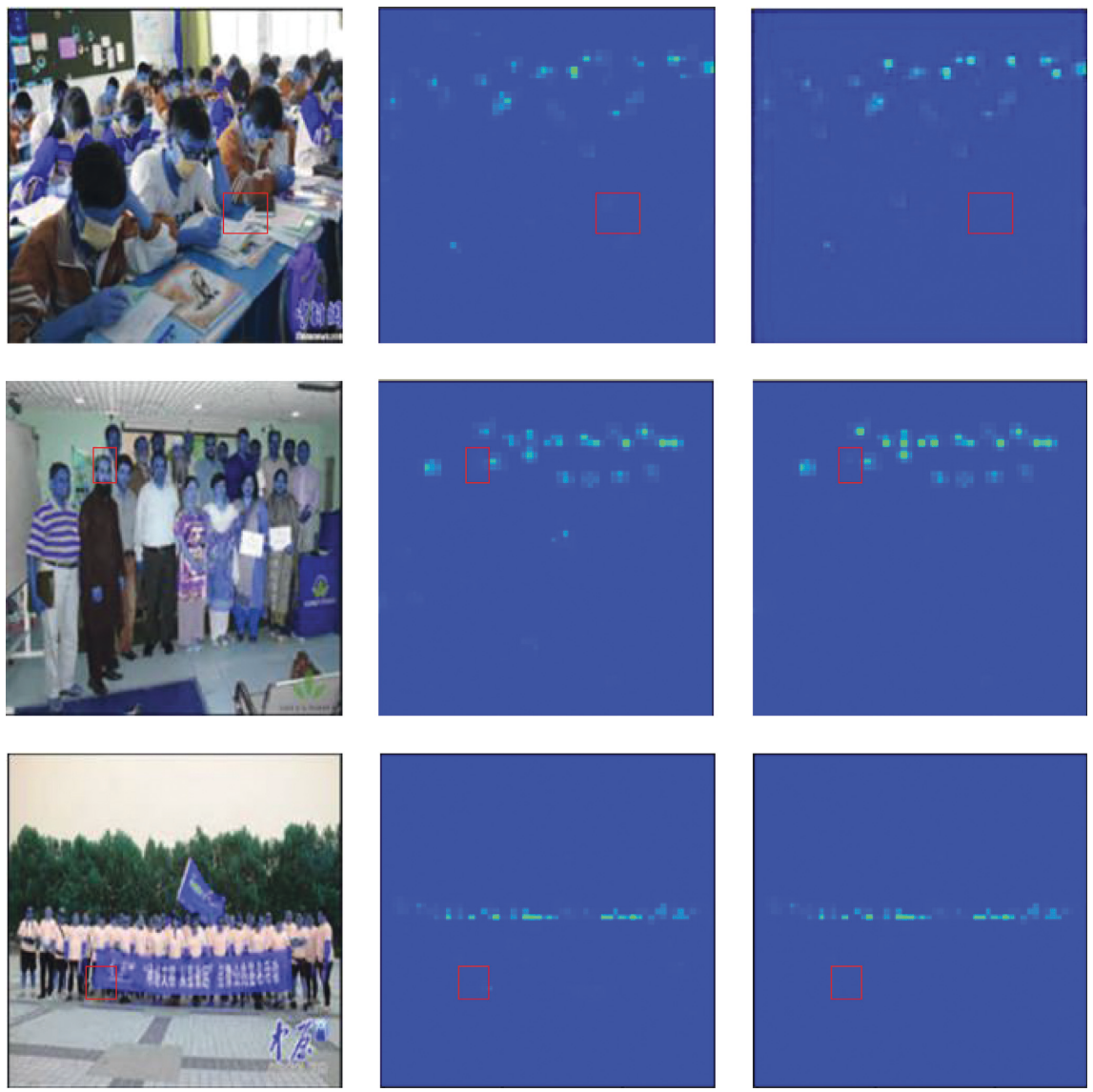


Figure 7: The effect of the compensator.

accuracy for small-scale features has a significant impact on the results. However, the dataset is aimed at people with different scale features, and there are also a certain number of features with larger scales. This requires the network to be able to count objects of different scales. Since MSCNN attaches equal importance to objects of all scales, its accuracy in counting small-scale objects with dense information is insufficient. MCNN is a classic method of crowd counting, and it also has better results for scale changes. The DSAM of DSA-CNN also has good results when counting crowds with large-scale variation. DCN also has excellent performance for two-branch networks of different scales. The network proposed in this paper performs crowd counting at different scales through different branches and pays more attention to small-scale targets with greater small information density in parameter allocation. The compensator module also has some optimization to the network. The hierarchical loss function uses different calculation methods for objects of different scales, which is more helpful for counting objects of different scales. For the detailed effect of each part of the network on the count, there is a detailed analysis later. The effect of counting images with large variation in feature scale is shown in Figure 5. The effect of counting images with similar feature scale sizes is shown in Figure 6. A multi-branch neural network with a compensator can successfully capture and count head features at different scales.

In order to analyze the effectiveness of the compensator and the adaptive grading function, we compare them with the networks without the compensator or the adaptive grading loss function and analyze the results. The MAE value of the network without the compensator module is 7.7, and that of the network with the compensation module is 7.3, which is relatively low. Through the comparison and the results displayed in Figure 7, it can be seen that the weight compensator has two main functions: one is to reduce the misjudgment of human head features as human heads, and

Table 3: AGLoss ablation of our ASNet on the dataset.

	1-level loss	2-level AGLoss	3-level AGLoss
MAE	7.9	7.3	7.5
MSE	306.3	302.4	305.8

Table 4: Branch comparison.

Configuration	MAE	MSE	Loss	RF	Filters	Parameters
3×3	7.6	313.6	0.00018	40	640	924225
5×5	8.1	328.4	0.00022	76	320	642849
7×7	14.2	417.6	0.00031	112	160	316177

the other is to improve the ability to recognize human head targets. Through these two aspects, the network has stronger results. The importance of the compensator is proved, which is helpful for the weight adjustment of the multibranch neural network. The adjustment of network weights also helps each branch of the multibranch neural network structure to better play different roles for targets of different scales.

The computational results of the proposed model with 2-level and 3-level AGLoss are presented in Table 3. It can be seen that AGLoss further improves the counting performance of the network with the compensator. The MAE of the network with the compensator and 2-level AGLoss reaches 7.3, which is better than the network with the compensator with MSE loss and 3-level AGLoss. The level 2 AGLoss has better generalization ability than the level 3 AGLoss. This may be caused by the over-grading of larger human head features, resulting in human head features being segmented in different regions. Therefore, the level 2 AGLoss was finally selected.

For further qualitative analysis, the capabilities of multiple branches are analyzed separately, which is also one of the reasons for the construction of branches. After using individual branches to make predictions and analyzing MAE, RMSE, and loss indicators, the results are shown in Table 4. Small-scale targets occupy a larger proportion in the dataset. The 3×3 branch is more effective than other branches. In addition, due to the large number of large-scale feature pixels, crowd counting only needs simple head information and does not require too much detailed information. Therefore, its information density is lower, and the use of fewer parameters of large-scale convolution kernels can also increase the efficiency of the network.

7. Conclusions

This paper proposed an improved multibranch convolution model for solving crowd counting in complex scenes. In the proposed model, multibranch structure was employed to capture head features of different scale, and an extra compensator was introduced to optimize weights of different branches according to different scale features. The feasibility of the proposed model was verified on a public crowd counting competition dataset. Experimental results showed that the proposed model accurately estimated the number of crowds with different head sizes and shooting distances. Meanwhile, compared with benchmarking methods, namely, MSCNN, MCNN, DSA-CNN, and DCN, the proposed model achieved the best evaluation performance. Furthermore, through the head-scale grading labeling, targets with different scales were optimized via adaptive grading loss function. Therefore, it is promising to utilize the proposed method to count the number of people in complex scenarios [24–27].

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was partially supported by National Key R&D Program of China (2018********01) and National Social Science Foundation Project (17BXW065).

References

[1] S. Ren, K. He, R. Girshick, J. Sun, and R.-C. N. N. Faster, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[3] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Computer Vision – ECCV 2018*, pp. 833–851, Springer Cham, New York, NY, USA, 2018.

[4] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, Venice, Italy, October 2017.

[5] T. Zhang, C. Xu, and M. H. Yang, "Robust structural sparse tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 473–486, 2019.

[6] T. Zhang, C. Xu, and M. H. Yang, "Learning multi-task correlation particle filters for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 365–378, 2019.

[7] Z. Wang, L. Wang, and C. Huang, "A Fast Abnormal Data Cleaning Algorithm for Performance Evaluation of Wind Turbine," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2020.

[8] W. Ge and R. T. Collins, "Marked point processes for crowd counting," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2913–2920, Miami, FL, USA, June 2009.

[9] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: an evaluation of the state of the art," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 4, pp. 743–761, 2012.

[10] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1299–1302, New York, NY, USA, October 2015.

[11] D. Oñoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *Proceedings of the Computer Vision - ECCV 2016*, pp. 615–629, Amsterdam, Netherlands, October 2016.

[12] 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 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 589–597, Las Vegas, NV, USA, June 2016.

[13] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4031–4039, Honolulu, HI, USA, July 2017.

[14] V. Sindagi and V. Patel, "Multi-level bottom-top and top-bottom feature fusion for crowd counting," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1002–1012, Seoul, Republic of Korea, October 2019.

[15] X. Chen, Y. Bin, N. Sang, and C. Gao, "Scale pyramid network for crowd counting," in *Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1941–1950, Waikoloa, HI, USA, January 2019.

[16] C. Xu, K. Qiu, J. Fu, S. Bai, Y. Xu, and X. Bai, "Learn to scale: generating multipolar normalized density maps for crowd counting," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8381–8389, Seoul, Republic of Korea, November 2019.

[17] M. Hossain, M. Hosseinzadeh, O. Chanda, and Y. Wang, "Crowd counting using scale-aware attention networks," in *Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1280–1288, Waikoloa, HI, USA, January 2019.

[18] M. Wang, H. Cai, J. Zhou, and M. Gong, "Stochastic multi-scale Aggregation network for crowd counting," in *Proceedings of the ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020.

[19] Z. Wang, L. Wang, C. Huang, Z. Zhang, and X. Luo, "Soil-moisture-sensor-based automated soil water content cycle classification with a hybrid symbolic aggregate approximation algorithm," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14003–14012, 2021.

[20] X. Jiang, L. Zhang, M. Xu et al., "Attention scaling for crowd counting," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4705–4714, Seattle, WA, USA, June 2020.

[21] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, "Multi-scale convolutional neural networks for crowd counting," in *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP)*, pp. 465–469, Beijing, China, September 2017.

[22] J. Wu, W. Qu, H. Yu, Y. Zhou, and Z. Cui, "A novel crowd counting method via deep convolutional neural network," in *Proceedings of the 2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 162–167, Tianjin, China, August 2019.

[23] W. Zhang, Y. Wang, Y. Liu, and J. Zhu, "Deep convolution network for dense crowd counting," *IET Image Processing*, vol. 14, no. 4, pp. 621–627, 2020.

[24] J. Cao, Y. Pang, and X. Li, "Learning multilayer channel features for pedestrian detection," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3210–3220, Jul. 2017.

[25] E. Walach and L. Wolf, "Learning to Count with CNN Boosting," in *Computer Vision – ECCV 2016*, pp. 660–676, Springer Verlag Cham, New York, NY, USA, 2016.

[26] C. Shang, H. Ai, and B. Bai, "End-to-end crowd counting via joint learning local and global count," in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, pp. 1215–1219, Phoenix, AZ, USA, September 2016.

[27] C. Cong Zhang, H. Hongsheng Li, X. Wang, X. Xiaokang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 833–841, Boston, MA, USA, June 2015.