Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 9769208, 1 page
https://doi.org/10.1155/2023/9769208

Retraction

Retracted: Research on Webcast Supervision Based on Convolutional Neural Network and Wireless Communication

Wireless Communications and Mobile Computing

Received 28 November 2023; Accepted 28 November 2023; Published 29 November 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

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

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

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

References

[1] Z. Sun, J. Sun, and X. Li, "Research on Webcast Supervision Based on Convolutional Neural Network and Wireless Communication," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 1191641, 7 pages, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 1191641, 7 pages
https://doi.org/10.1155/2021/1191641

Research Article

Research on Webcast Supervision Based on Convolutional Neural Network and Wireless Communication

Zhidong Sun,[1] **Jie Sun**,[2] **and Xueqing Li**[1]

[1]*School of Software, Shandong University, Shandong, China*
[2]*Affiliated Hospital of Qingdao Binhai University, Shandong, China*

Correspondence should be addressed to Xueqing Li; xqli@sdu.edu.cn

Received 3 August 2021; Accepted 24 August 2021; Published 16 September 2021

Academic Editor: Balakrishnan Nagaraj

Action recognition is the technology of understanding people's behavior and classification from video or image sequences. This thesis uses the deep learning approach for action recognition to realize webcast supervision. This paper uses the convolutional neural network (CNN) and the Gaussian Mixture Model (GMM) to establish the webcast supervision system. At the same time, streaming-based wireless communication network technology is adopted to ensure video transmission speed and quality. Results show that the average detection speed of the system can reach 11.86 frame/s, and the average recognition accuracy is 92.16%, and the missed detection rate is lower than 5%. The design of this system can fully meet the requirements of webcast supervision.

1. Introduction

As a new internet entertainment business model, the webcast is the product of the rapid development of the information age. The spread and rapid growth of this are mainly based on the progress of Internet technology and the rise of various live broadcast platforms. As a new product of the Internet age, there is nothing wrong with bringing social entertainment. However, if it challenges social value and standard social order, it must be effectively regulated. For example, in the current major live broadcast platforms on the market, there are often problems in the operation of the broadcast, like being vulgar and low threshold access of the host broadcast. These problems affect the development of the live broadcast platform and affect the environment of the Internet. However, current live broadcast supervision is mostly manual, which is far from efficient and also vulnerable to loopholes by illegal personnel. Therefore, it is urgent to establish an automatic live broadcast supervision system.

At present, there is relatively little research on the supervision of webcast, but there are more researches on video monitoring system [1, 2], which two share the same characteristics in some way. The so-called video monitoring system is the product of the comprehensive application of multimedia technology, computer network, industrial control, and artificial intelligence. It is developing towards the direction of video digitization, system network, and intelligent management. In the simulation era, video monitoring is mainly represented by an analog tape recorder. The system comprises an analog camera, special cable, video switching matrix, analog monitor, analog video equipment, and videocassette. In the digital era, digital video recorder has begun to appear due to the development of digital video compression and coding technology. DVR enables users to digitize analog video signals and store them on a computer hard disk instead of a videocassette. Digital storage greatly improves the user's ability to process video information. In the network era, with the further development of the whole digital and networked video monitoring system, the role of video monitoring is becoming more and more important. However, it is weighty work for the staff who continuously monitor activities in the monitoring scene, day and night. Therefore, video monitoring needs to be more intelligent, active, and effective, so computer vision and application researchers timely put forward the concept of a video monitoring of new generation.

Intelligent video monitoring is a new subject direction and application field by combining computer vision technology with multimedia communication technology, which is also a new challenging research content in the field of computer vision. By applying the method of computer vision and video analysis, it realizes the positioning, identification, and tracking of the target in the monitored scene by automatic analysis of the image sequence recorded by the camera without human intervention. On this basis, it will analyze and judge the related targets' behavior to realize 24-hour all-weather monitoring, accurate alarm, and high response speed. The introduction of intelligent video monitoring technology into the supervision of live broadcast networks can monitor the environment of live broadcasts and alarm the bad behavior of the host in time, which significantly improves the working efficiency of the staff.

In recent years, CNN has been widely used in human behavior recognition. As a representative deep learning network, CNN has a great improvement over the traditional neural network recognition effect [3–9]. Moreover, this method is an end-to-end recognition method, which does not need to be designed manually, and is of translation invariance and scale invariance. Its calculation way is very similar with the mammal visual system.

In this paper, the supervision of network live broadcast is a security supervision system which is established based on CNN. It can monitor the behavior of anchors in real time so as to guarantee the safety and health of network live broadcast.

2. Methods

2.1. Neural Network. A neural network is a kind of machine learning model employed for data classification or data prediction. The model structure is constructed based on data and learning rules. A neural network regression model is trained with data based on a training algorithm to predict a subsequent set of data.

As shown in Figure 1, a neural network model consists of some nodes/neurons, set at multiple layers: the input layer, one or more hidden layers, and the output layer. Each node/neuron has an activation function, which calculates how much neuron is "stimulated." At each layer, the collections of nodes/neurons transform the input parameters; these parameters are distributed to the next layer, which is described as

$$z_j^n = \sum\left(w_{ji}^{(1)} x^{n-1} i + w_{j0}^{(1)}\right), \tag{1}$$

$$a_1^n = \sum\left(w_{ij}^{(2)} z^{n-1} i + w_{10}^{(2)}\right), \tag{2}$$

$$y_1^n = F(a_1^n), \tag{3}$$

where x represents the input to the first layer; z represents the first layer's output; i, j represents the neural network node index; ${w_{ji}}^{(1)}$ represents the weight between the jth node in the ith layer and the ith node in the $(i+1)$th layer; $F({a_i}^n)$ represents the output value of the ith node in the $(n+1)$th layer after being activated by the activation function; w and w_0 represent the weight and bias between the neurons, which measures the significance of the data passed along the link (synapse). $F(a)$ employs the activation function, which employes the hidden layer's aggregated output to calculate output y.

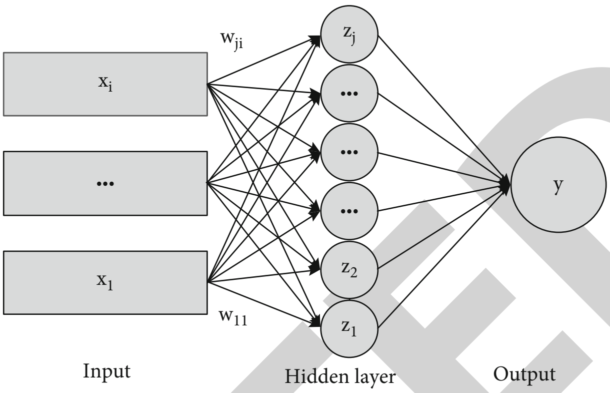

Figure 1: Scheme of neural network.

The initial weights and biases are randomly assigned, and the training process continues until the desired output is obtained, which is evaluated by the cost function

$$E(w) = \frac{1}{2}\sum_{k=1}^{K} y(x_k, w_k) - t_k^2, \tag{4}$$

where y represents the output; t represents the desired output. The Levenberg-Marquardt (LM) algorithm is utilized in the neural network training process, which is a variation of gradient descent. The weight and bias of the neural network model are changed during the training process to minimize the error, which is described as

$$w^n = w^{n-1} - \left(J^T J + \mu I\right)^{-1} J e^{n-1}, \tag{5}$$

where $J = \partial E/\partial w$ represents the full-scale Jacobian matrix related to w; I represents the identity matrix; m represents a combination coefficient; e represents the prediction error.

The Levenberg-Marquardt algorithm starts with a forward computation by (1), (2), and (3). The prediction errors of the output layer and the hidden layer are calculated by

$$\begin{aligned} e^{(3)} &= y_1 - t, \\ \delta_1^{(3)} &= e_1^{(3)}, \\ \delta_j^{(2)} &= w_{1j}\delta_1^{(3)}. \end{aligned} \tag{6}$$

As shown in Equations (7) and (8), the Jacobian is calculated by a back-propagation process:

$$\frac{\partial E}{\partial w_{ji}} = \delta_j^{(2)} x_i, \tag{7}$$

$$\frac{\partial E}{\partial w_{1j}} = \delta_1^{(3)} z_j. \tag{8}$$

In the training process of the sample, the learning sample should be processed to make it fluctuate in a certain range. The normalization method is adopted in this paper to process the data to ensure that the data is between 0 and 1, which is written as

$$x_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \tag{9}$$

where x_i represents the average value, $x_{\max}$ represents the maximum value, and $x_{\min}$ represents the minimum value.

2.2. Convolutional Neural Network. Figure 2 is the structure of a typical CNN, which consists of an input layer, a convolution layer, a downsampling layer (pooling layer), a fully connected layer, and an output layer.

The input of a CNN is usually the raw image X. If H_i is the characteristic graph of the convolutional neural network layer $i(H_0 = X)$, the production process of H_i is as follows:

$$H_i = f(H_{i-1} \otimes W_i + b_i), \tag{10}$$

where W_i is the weight vector of the convolution kernel at the first level i. The sign $\otimes$ represents the convolution kernel to convolve with the image of layer $i-1$ or feature graph, and the output of the convolution is added to the offset vector b_i at level i. Finally, the characteristic graph H_i of the layer i is obtained through the nonlinear excitation function $f(x)$.

The subsampling layer is usually behind the convolution layer, and the subsampling rule is as follows:

$$L_i = \text{subsampling}(H_{i-1}). \tag{11}$$

The CNN classifies the extracted features through the alternating transfer of multiple convolutional layers and lower sampling layer, and then, the probability distribution Y based on the input is got.

$$Y(i) = P(L = l_i | H_0 ; (W, b)). \tag{12}$$

The training objective of CNN is to minimize the loss function $L(W, b)$ of the network. The difference of the input H_0 and the value of expectation (residual error) is calculated by the loss function after the forward conduction. In this paper, the Levenberg-Marquardt is used. The Levenberg-Marquardt back propagation is employed to enhance the model training rate related to pure error back propagation or steepest descent, and this algorithm maintains the accuracy of the trained model. The neural network regression model is trained with the designed model structure, input parameters, and the number of nodes. The accuracy of both the training and the prediction/estimation is evaluated by mean absolute error (MSE), which is written as

$$e_{\text{avg}} = \frac{1}{n} \sum_{k=1}^{n} \left| \text{Output}_k - \text{Output}_{r,k} \right|, \tag{13}$$

where e_{avg} represents the average absolute error; n represents the number of data points, Output_k represents the kth estimated output parameter, and $\text{Output}_{r,k}$ represents the k th reference output parameter.

In the training process, the CNN is a commonly used gradient descent method. The residual error is propagated back through gradient descent, and the trainable parameters of each layer of CNN are updated layer by layer (W and b). The learning rate parameter η is used to control the intensity of the normal propagation of residuals:

$$\begin{aligned} W_i &= W_i - \eta \frac{\partial E(W, b)}{\partial W_i}, \\ b_i &= b_i - \eta \frac{\partial E(W, b)}{\partial b_i}. \end{aligned} \tag{14}$$

2.3. Extracting the Foreground of the Person Video. The common character behavior video foreground extraction tools are mixed Gaussian background modeling (GMM) [10], codebook algorithm [11], self-organizing background checks [12], vibe algorithm [13], and so on. In this paper, the GMM method is used to extract anchors' behaviors. The GMM is used to conduct statistics on pixel sample information, and statistical information such as probability density of a large number of sample values of pixel points over a long period of time is used to represent the background. Generally, this statistical information includes the number of patterns, the mean of each pattern, and standard deviations. Then, the method of statistical difference (such as 3σ principle) is used to distinguish the target pixel, which also has a good modeling effect on the more complex dynamic background. In the GMM, 3 to 5 Gaussian models are generally used to represent each pixel's features in the image. Moreover, each pixel in the current image is matched with the mixed Gaussian model so as to update the model after a new frame is obtained. In addition, if the match is successful, it is the background point; otherwise, it is the foreground point.

In this model, it is general that the color information between pixels is not related. Therefore, the pixels are handled independently of each other. For each pixel on the video frame image, the change of the pixel value on the sequence image is treated as a random process that continuously generates the pixel value. It means that the Gaussian distribution is used to describe the color rules of each pixel.

For the multimodal (multimodal Gaussian distribution) models, each pixel on an image frame is viewed as a superposition of multiple Gaussian distributions with different weights. The weights and distribution parameters of each Gaussian distribution are updated over time, and each Gaussian distribution corresponds to a state that may produce the color of the pixel. In the process of color image processing, it is assumed that the three color channels of pixel point are independent of each other and have the same variance. For

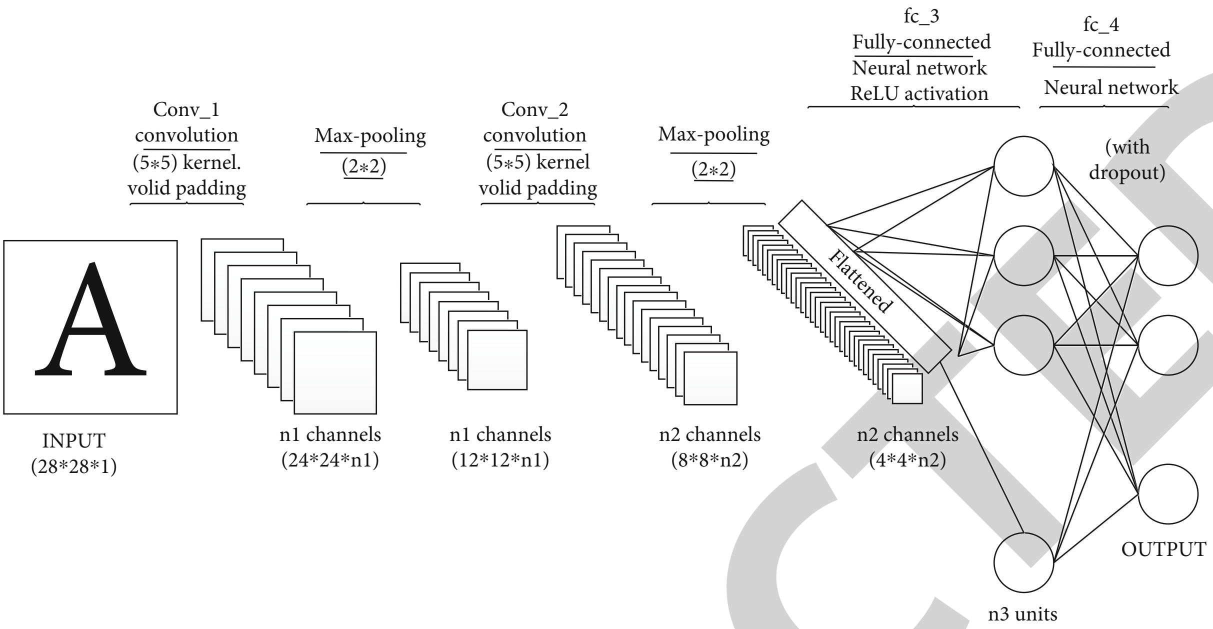

Figure 2: Fundamentals of CNN.

the observation dataset $\{x_1, x_2, \cdots, x_N\}$ of the random variable $X, x_t = (r_t, g_t, b_t)$ is the sample of the pixel in the t moment, so the single sampling point x_t obeys the probability density function of the mixed Gaussian distribution:

$$p(x_t) = \sum_{i=1}^{k} w_{i,t} \times \eta\left(x_t, \mu_{i,t}, \tau_{i,t}\right), \eta\left(x_t, \mu_{i,t}, \tau_{i,t}\right) = \frac{1}{\left|\tau_{i,t}\right|^{1/2}} e^{-(1/2)\left(x_t-\mu_{i,t}\right)^T \tau_{i,t}{}^{-1}\left(x_t-\mu_{i,t}\right)}, \tau_{i,t} = \delta_{i,t}^2 I, \quad (15)$$

where k is the total number of distribution patterns, $\eta(x_t, \mu_{i,t}, \tau_{i,t})$ is the i Gaussian distribution at t moment, $\mu_{i,t}$ is the average of the Gaussian distribution, $\tau_{i,t}$ is the covariance matrix for it, $\delta_{i,t}$ is the variance, I is the three-digit identity matrix, and $w_{i,t}$ is the weight of the i Gaussian distribution at the t moment.

2.4. Extraction of Character Features. Feature extraction is an important part of character behavior detection. For network broadcast, sample selection plays an important role in the detection of character behavior. Currently, there are many methods of character feature extraction, such as SIFT [14], Hear [15], HOG [16], and LBP [17]. HOG feature extraction method is used in this paper. The specific steps of HOG are as follows.

2.4.1. Graying. In view of the fact that the color information of the image does not play a significant role in the live broadcast monitoring, it is necessary to convert the image to grayscale first in order to facilitate the later operation.

2.4.2. Standardize the Gamma Space and Color Space. This step mainly reduces the impact of illumination on the recognition effect and normalizes the image before the subsequent steps. The gamma compression formula is shown in

$$I(x, y) = I(x, y)^{\text{gamma}} \left(\text{gamma} = \frac{1}{2}\right). \quad (16)$$

2.4.3. Calculating Image Gradient. In this step, it needs to calculate the horizontal gradient and vertical gradient of the image and calculate the gradient direction value of each pixel. The main effect of this step is to further reduce the effect of illumination (light). The computing method of the gradient is shown in

$$G_x(x, y) = H(x+1, y) - H(x-1, y), G_y(x, y) = H(x, y+1) - H(x, y-1), \quad (17)$$

where $G_x(x, y), G_y(x, y), H(x, y)$ is the horizontal and vertical gradients and pixel values at pixel points, respectively; furthermore, the gradient value and the gradient direction of the pixel (x, y) can be written by

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}, \alpha(x, y) = \tan^{-1}\left(\frac{G_y(x, y)}{G_x(x, y)}\right). \quad (18)$$

2.4.4. Component Gradient Histogram. In this step, the image is divided into a number of cell units $(a * a)$. The histogram of n bins is used to calculate the gradient information of the $a * a$ pixels. As is shown in Figure 3, if the gradient of a pixel is within a^2 degrees, the $i+1$ bin count in the histogram is incremented by 1. In this way, the histogram of the cell's gradient direction can be obtained.

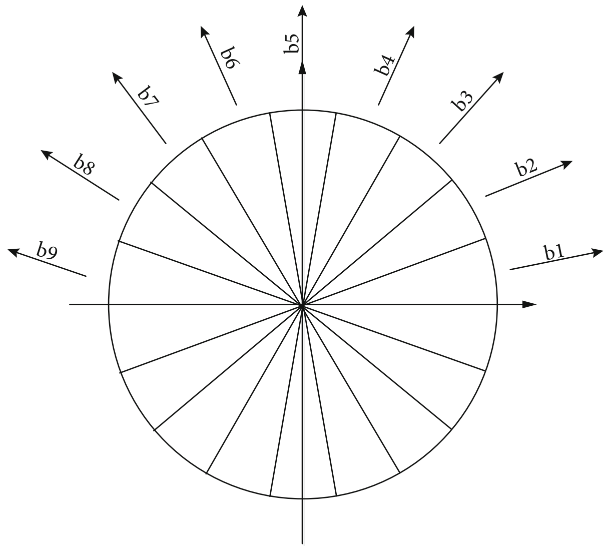

Figure 3: Graph of gradient direction projection.

2.4.5. Combination of Block. Variations in the foreground-background contrast and local illumination greatly increase the range of gradient intensity. In order to avoid this effect, it is usually necessary to do a local normalization of the gradient intensity. Normalization can compress the shadows, lighting, and edges. A common approach is to combine individual cell units into a block to get the HOG feature of this block by connecting the eigenvectors for all cell units in a partnership. Consequently, the overlap between the different blocks can effectively improve the problems such as local lighting.

3. Results and Discussions

3.1. Design of Webcast Supervision System. As is shown in Figure 4, the framework of the webcast supervision system is based on the GMM and the CNN; the specific process is as follows:

Firstly, a dataset is established to collect a large number of anchors' behavior samples and the size unified to the height is 100 and the width is 50. Then, the training set and the test set are found, respectively. Before the neural network model training, a large-scale training set should be established first. In order to detect the effect of the network, a test set should be built to consider the effect of the neural network model. 1241 images of anchors' behavior are collected in this paper, of which the training set and the test set were 80% and 20%, respectively. In terms of data sources, live-broadcast shots of anchors on various live broadcast platforms were selected, including waving, clapping, walking, and running. In order to monitor anchors' behaviors, this paper selected bad behaviors (smoking) as cases of violations.

Secondly, the classifier was trained and the CNN structure was built, and the dataset was trained to obtain the model.

In this paper, the structure of the CNN network includes 5 layers.

(1) Convolutional layer C1: the input is the 3-channel RGB image with a size of 100 ∗ 5, which is convolved with a 5 ∗ 5 convolution check to get 16 feature maps of 96 ∗ 46

(2) Sampling layer S2: S2 is a lower sampling layer, which uses the principle of local correlation of images to sample images. This method can reduce the amount of data processing when retaining effective information. The subsampling is carried out for the data of the C1 layer. For the 3 ∗ 3 region of each C1 feature map, we sum and add bias to the 9 pixels and then store the results calculated by using the Sigmoid activation function in the new feature map. Finally, we get a feature diagram of 16 ∗ 32 ∗ 16 in S2 layer S2

(3) Convolutional layer C3: after step 2, the S2 layer is convoluted by the 16 3 ∗ 3 convolution checks and a 30 ∗ 14 feature map is got

(4) Sampling layer S4: the function of S4 is the same as the S2; it has 16 15 ∗ 7 feature maps, which are connected with the feature maps of the C3 layer

(5) F5: the connection between the F5 layer and the S4 layer is the standard full connection, which is a standard MLP neural network transfer mode. Finally, F5 is connected to the classifier to complete the last part of the training

Thirdly, enter the video stream. In the paper, we used the HikVision webcam and input the video stream in the RTSP format.

Fourthly, Gaussian background was mixed for modeling, and foreground pixels were extracted after modeling. The process of the Gaussian background modeling algorithm is as follows:

(1) Each new image's prime value x_t is kept comparing (15) until a distribution model matches the new pixel values. In other words, the mean deviation of the same model is below 2.5σ:

$$\left|X_t - u_{i,t-1}\right| \le 2.5\sigma_{i,t-1} \tag{19}$$

(2) The weight of each pattern is updated according to Equation (20), where α is the learning rate. For the matched pattern, $M_{k,t} = 1$; otherwise, however, $Mk = 0$. Then, the weight of each pattern is normalized:

$$\omega_{k,t} = (1-\alpha) \times \omega_{k,t-1} + \alpha \times M_{k,t} \tag{20}$$

(3) The mean value and variance of the unmatched pattern remain unchanged, and the parameters of the matched pattern are updated according to

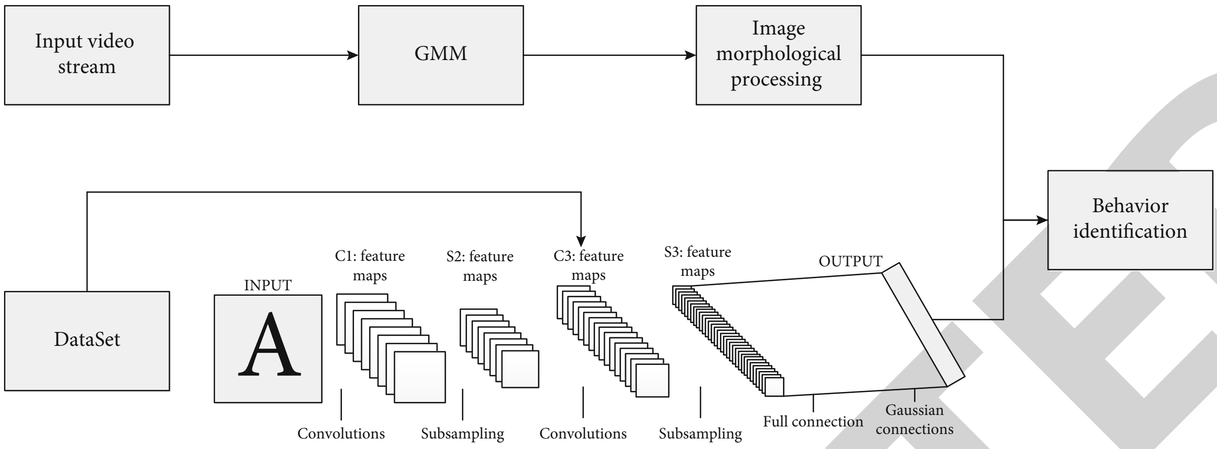

Figure 4: Webcast supervision system.

Table 1: The accuracy of identification.

Category		Accuracy (%)	Detection speed (frame/s)	Missed detection rate (%)
Waving		91.7	12.2	4.3
Clapping		92.6	11.7	5.4
Walking		97.2	10.9	3.8
Running		85.9	13.4	3.7
Smoking		93.4	11.1	4.7
	GMM+CNN	92.16	11.86	4.38
Average	Faster+R-CNN	88.71	0.87	41.56
	HOG+SVM	90.70	4.72	24.51

$$\sigma_t^2 = (1-\rho)\times\sigma_{t-1}^2 + \rho\times(X_t-\mu_t)^{T+1}, \rho = \alpha\times\eta\times(X_t|\mu_k,\sigma_k), \mu_t = (1-\rho)\times\mu_{t-1} + \rho\times X_t \tag{21}$$

(4) If no relevant patterns are matched in step 1, the pattern with the least weight will be replaced. The mean value of this pattern is the current pixel value, the standard deviation is the initial larger value, and the weight is the smaller value

(5) Each pattern is arranged in descending order according to the value of ω/α^2

(6) The B module is the back view according to (22), where T representation of the proportion:

$$B = \arg\left(\min\left(\sum_{k=1}^{b} w_k > T\right)\right) \tag{22}$$

Fifthly, the extracted foreground point is binarized, and the image morphology is processed, including filtering, expansion, and etching, and the edge contour is extracted to get the processed image.

Finally, the processed graphics and the model trained by CNN were compared and analyzed to judge the anchor's behavior.

3.2. The Accuracy of Identification. Each test sample will be tested in the testing stage for each kind of behavior, and then, the classification result will be obtained. The classification result will be compared with the label of the test sample. In Table 1, the accuracy rate of each class is, respectively, counted. Finally, the average accuracy rate is obtained by using the accuracy rate of each class. In addition, the detection speed and missed detection rate are also given in Table 1. Furthermore, comparing with the recognition accuracy of different methods, it is seen that the recognition method of GMM+CNN is higher than the other algorithms, reflecting its accuracy in live broadcast behavior detection.

4. Conclusion

In this paper, an anchor behavior monitoring system based on a CNN+GMM is designed. The deep neural network can autonomously and fully learn behavior features, which avoids explicit feature extraction and makes the algorithm more robust, by effectively eliminating the influence of illumination, angle, form, and other factors on the final detection results.

The system can meet the requirements of real-time performance. There is no lag phenomenon in visual measurement in real-time video. The average detection speed can reach 11.86 frame/s, and the average recognition accuracy is 92.16%, and the missed detection rate is lower than 5%. The design of this paper can fully meet the requirements of webcast supervision.

Data Availability

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Conflicts of Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

[1] Z. Wei, S. Cheung, and M. Chen, “Hiding privacy information in video surveillance system,” in *IEEE International Conference on Image Processing 2005*, Genova, Italy, 2005.

[2] K. A. Joshi and D. G. Thakore, “A survey on moving object detection and tracking in video surveillance system,” *International Journal of Soft Computing & Engineering*, vol. 2, no. 3, pp. 44–48, 2012.

[3] Y. C. Su, T. H. Chiu, C. Y. Yeh, H. F. Huang, and W. H. Hsu, “Transfer learning for video recognition with scarce training data for deep convolutional neural network,” 2014, https://arxiv.org/abs/1409.4127.

[4] J. Liu, Z. Li, Y. Tang et al., “3D convolutional neural network based on memristor for video recognition,” *Pattern Recognition Letters*, vol. 130, pp. 116–124, 2020.

[5] Y. Ji, S. Kim, and K. B. Lee, “Sign language learning system with image sampling and convolutional neural network,” in *2017 First IEEE International Conference on Robotic Computing (IRC)*, Taichung, Taiwan, 2017.

[6] I. C. Duta, L. Liu, F. Zhu, and L. Shao, “Improved residual networks for image and video recognition,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 9415–9422, Milan, Italy, 2020.

[7] L. Cong, M. Longhua, and L. Feng, “Multi-timescale gated neural network for video recognition,” *Recent Patents on Computer Science*, vol. 10, no. 999, pp. 1–1, 2017.

[8] X. Qi, C. Liu, and S. Schuckers, “CNN based key frame extraction for face in video recognition,” in *IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, Singapore, 2018.

[9] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, “Atention transfer from web images for video recognition,” in *Proceedings of the 25th ACM international conference on multimedia*, Mountain View, California, USA, 2017.

[10] J. Kan, K. Li, J. Tang, and X. Du, “Background modeling method based on improved multi-Gaussian distribution,” in *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, Taiyuan, China, 2010.

[11] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, “Real-time foreground-background segmentation using codebook model,” *Real Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[12] D. Avola, M. Bernardi, L. Cinque, C. Massaroni, and G. L. Foresti, “Fusing self-organized neural network and keypoint clustering for localized real-time background subtraction,” *International Journal of Neural Systems*, vol. 30, no. 4, p. 2050016, 2020.

[13] O. Barnich and M. van Droogenbroeck, “ViBe: a universal background subtraction algorithm for video sequences,” *IEEE Transactions on Image processing*, vol. 20, no. 6, pp. 1709–1724, 2011.

[14] J. Luo and G. Oubong, “A comparison of sift, pca-sift and surf,” *International Journal of Image Processing (IJIP)*, vol. 3, no. 4, pp. 143–152, 2009.

[15] Z. Fei and P. D. With, “Fast facial feature extraction using a deformable shape model with Haar-wavelet based local texture attributes,” in *2004 International Conference on Image Processing, 2004. ICIP'04*, pp. 1425–1428, Singapore, 2004.

[16] A. J. Newell and L. D. Griffin, “Multiscale histogram of oriented gradient descriptors for robust character recognition,” in *2011 International Conference on Document Analysis and Recognition*, pp. 1085–1089, Beijing, China, 2011.

[17] Guoying Zhao, T. Ahonen, J. Matas, and M. Pietikainen, “Rotation-invariant image and video description with local binary pattern features,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1465–1477, 2012.