WSLCC: A Weakly Supervised CNN-Transformer Model for Crowd Counting and Its Application in Sports Venue Management

Lizhou Pu[1*], JunLu[2]
[1]Department of Public Physical Education, Wenhua*College*, Wuhan 430074, China
[2]Physical education teaching and Research section, Wuhan donghu college, Wuhan 430212, China
E-mail: 18062028585@163.com
[*]Corresponding author

Keywords: weakly supervised learning, crowd counting, transformer, encoder, CNN; sports venues

Received: April 13, 2025

Aiming at the problem of low accuracy and poor adaptability of current crowd counting methods in sports venue management, an innovative crowd counting model based on weakly supervised learning (WSLCC) is proposed and a corresponding management platform is designed. In terms of model construction, this work combines weakly supervised learning ideas to deeply integrate traditional Convolutional Neural Networks (CNN) with Transformers. Firstly, advanced Convolutional Feature Module (CFM) is utilized to accurately capture and extract high-level semantic information of the crowd in video frames. Subsequently, this information is fed into an efficient Transformer Feature Module (TFM), which utilizes its powerful modeling capabilities to comprehensively construct global contextual information and long-range dependencies. Weakly supervised learning is reflected in using a small amount of labeled data to guide model learning and reduce dependence on a large amount of accurately labeled data. To validate the performance of the model, experiments are conducted on multiple datasets. The model ablation experiment shows that on the UCF_CC_50 dataset, the mean absolute error (MAE) of the WSLCC model is 62.8, and the root mean square error (RMSE) is 95.4, which is 2.5% and 0.2% lower than that of the LSC-CNN model, respectively. With the gradual addition of CFM and TFM modules, the model performance significantly improves, and the combined MAE and RMSE index values are the lowest. In practical applications, the sports venue management platform based on the WSLCC model achieves significant results, with an accuracy rate of 95.1% in crowd statistics, a venue utilization rate of 85.4%, a satisfaction score of 4.5 for resource allocation, and a management response time shortened to 5.3 minutes. This study effectively improves the adaptability and accuracy of crowd counting methods in complex environments, promoting the improvement of sports venue management efficiency.

Povzetek: Za področje upravljanja športnih prizorišč in množic je predstavljen model WSLCC, ki v šibko nadzorovanem okviru združi konvolucijsko semantiko (CFM) z izboljšanim transformerjem (TFM, na osnovi Swin z zamaknjenimi okni) za hkratno zajemanje lokalnih značilk in globalnega konteksta iz video slik. Jedro novosti je v rabi točkovnih, delnih anotacij za učenje gostot brez natančnih tabel označb ter v pretočnem kodirniku, kjer vektorsko sploščanje, lahka hierarhična pozornost in rezidualno zlivanje omogočijo robustno štetje glav v heterogenih, okludiranih prizorih.

1 Introduction

As large-scale sports events and mass-participation sports activities gain increasing popularity, the management of crowd safety at sports venues and public spaces is facing substantial challenges. Traditional crowdcounting (CC) methods typically rely on either manual inspections or fully supervised deep learning models that necessitate extensive data labeling. Manual inspections, however, are not only time-consuming but also susceptible to subjective biases. On the other hand, fully supervised deep learning models fall short in practical applications because of the exorbitant costs involved in data labeling and their limited ability to generalize across diverse scenarios [1-3]. In this context, weakly supervised learning has gradually emerged as a focal point of research within the domain of CC due to its ability to utilize partial or coarse-grained labels for model training [4]. By decreasing reliance on pixel-level annotation, weakly supervised methods can markedly cut the costs associated with data collection and annotation, and simultaneously enhance the algorithm's adaptability to complex sports scenes. Presently, the application of CC technology in sports management primarily concentrates on scenarios such as passenger flow statistics, safety alerts, and resource allocation. Nonetheless, the unique challenges posed by dynamic lighting changes, dense occlusion issues, and the spatial heterogeneity of audience distribution within sports venues—such as the varying densities between grandstand areas and evacuation routes—impose greater demands on traditional regression or detection-based counting methods [5-6]. In addition, the spatiotemporal characteristics of crowd flow during sports events, such as pre match gathering, mid match retention, and post match

evacuation, further require algorithms to have dynamic adaptability, which poses a dual challenge of multi-modal fusion and temporal modeling for the design of CC algorithms [7-8]. Many scholars have conducted research on this issue.

The dataset currently released is relatively small in scale and cannot fulfill the requirements of supervised learning-based CNN algorithms. To overcome the shortcomings of traditional manual counting methods, He X et al. proposed a method that combined CNN and Transformer networks for object counting in complex scenes. The results showed that this method reduced the error rate by 13.4%, indicating that the fusion of CNN and Transformer networks was effective in object counting in computer vision tasks [9]. To promote the application of CC methods in disaster management systems, public activities, safety monitoring, and other fields, Khan et al. proposed an end-to-end semantic segmentation framework for CC in dense and crowded images. As a result, it was found that the multi-scale features extracted by the algorithm from the image overcame the scale variation of crowded images [10]. In an effort to boost the accuracy of CC, Sindagi et al. introduced a novel CC network. This network progressively generated a crowd density map by means of residual error estimation. The proposed approach employed VGG16 as its backbone network. It took the density map produced by the final layer as an initial, rough prediction. Leveraging residual learning, the network then iteratively refined this prediction, ultimately generating increasingly detailed and accurate density maps. The results demonstrated that this method brought about a substantial reduction in counting errors, marking a significant improvement over previous techniques. [11]. To improve the accuracy of intelligent monitoring for crowd monitoring, Pang et al. developed a horizontal federated learning framework to train CC models while protecting privacy. This framework enabled intelligent monitoring systems to acquire knowledge through model integration while keeping the private data on local devices inaccessible [12]. To address the uncertainty issues in CC methods, Oh et al. proposed an extensible neural network framework that used bootstrap ensembles to quantify the decomposed uncertainty. The results showed that the suggested uncertainty quantification approach provided additional insights for CC problems and was easy to implement [13]. To improve the application effect of CC calculation method in image/video analysis, Vivekanandam et al. proposed a fast image CC method utilizing lightweight Convolutional Neural Network (CNN). The results showed that this method could easily classify head counts in some fields of view and is more accurate than other pre trained neural network models [14]. Oh M focused on the uncertainty of side weighting, while Vivekanandam [14] emphasized lightweight and rapid counting. The CC model based on weakly supervised learning (WSLCC), leveraging a CNN-Transformer architecture, accurately captures high-level semantic information under weak supervision, constructs a global context, and significantly enhances CC performance. Its advantage lies in its adaptability to weakly supervised scenarios and its more precise and reliable counting. Deep learning techniques are employed to establish a correlation between the visual content of images and the corresponding distribution of crowd density. Despite the notable advancements achieved thus far, accurately identifying pedestrians who are positioned at a considerable distance from the camera still poses a formidable challenge. Moreover, these challenging cases frequently account for a significant proportion of the dataset. Therefore, Chen et al. proposed a difficult sample focusing algorithm for CC regression tasks. This algorithm reduced the contribution of easy samples, enabling the model to promptly direct its attention towards challenging instances. Then, higher importance was given to difficult samples with erroneous estimations, and it was found that the introduced approach surpassed the leading-edge techniques [15]. Related research work is shown in Table 1.

Table 1: Related research worksheets

Author	Types of models	Types of models	MAE/RMSE	Superiority	Limit
HeX et al. [9]	CNN and Transformer fusion target counting model	UCF-QNRF data set	MAE:80.0	Increase complexity scenarios Target count accuracy	Poor versatility and lack of multi-data set verification
KhanK et al. [10]	End-to-end semantic segmentation framework	ShanghaiTech PartA data set	RMSE: 110.0	Extract multi-scale features to overcome the scale of crowded images change	The computational complexity is high and the real-time performance may be limited
SindagiVA et al. [11]	A CC network based on residual error estimation (VGG16 backbone)	UCF_CC_50 data set	MAE:75.0	Significantly improve the error and improve the counting accuracy	Depending on the VGG16 backbone network, there may be parameters More problems with large amount of computation
PangY et al. [12]	Level federal learning framework	WorldExpo'10 data set	/	Protect privacy and improve the accuracy of intelligent crowd monitoring nature	Model aggregation process may increase communication overhead and training Efficiency may be affected
OhM et al. [13]	Scalable neural network framework (bootstrap set Quantitative uncertainty)	Mall data set	/	Quantify the problem of counting people Uncertainty, easy to achieve	Specific quantitative indicators are indicated

Vivekananda mB. et al [14]	Lightweight CNN model	UCSD data set	MAE:9 0.0	Lightweight, can quickly count the number of images	Accuracy can be counted in complex backgrounds or occlusions It can go down
Chen et al. [15]	Difficult sample focusing algorithm (CC regression task)	Fudan-ShanghaiTech data set	MAE:7 0.0	Improve the detection of difficult samples ability	It is not mentioned whether the processing of easy samples may affect it The overall generalization ability of the model

In summary, research on intelligent algorithms for CC has yielded certain results and holds considerable importance in various aspects of daily life. Nonetheless, challenges such as occlusion in complex scenes, variations in target size due to perspective effects, and the difficulty in accurately identifying individuals within dense crowds persist. These problems complicate the accurate segmentation and counting tasks, particularly in crowded environments where the error rate tends to rise substantially. Moreover, the algorithms lack robustness against interference factors like lighting variations and background clutter, which can also impair counting accuracy. Based on this, the study innovatively proposes the WSLCC. The model first utilizes an advanced CNN Feature Extraction Module (CFM) to accurately capture and extract high-level semantic information of the crowd in video frames. Subsequently, this information is fed into an efficient Transformer Feature Module (TFM), which can fully utilize its powerful modeling capabilities to comprehensively construct global contextual information and long-range dependencies. Through this process, the model can significantly improve its CC performance under weakly supervised conditions, achieving more accurate and reliable headcount statistics. Then, based on the WSLCC model, a sports venue resource information management platform is designed to improve the adaptability and accuracy of CC methods in sports venue management, and promote the improvement of sports venue management efficiency. To address the challenges of difficult detection of pedestrians far from the camera and the abundance of challenging samples in current research, the proposed WSLCC method uses a CNN feature extraction module to accurately capture high-level semantic information about people. It then employs a Transformer feature extraction module to construct global context and long-distance dependencies, which helps to more comprehensively identify targets, focus on challenging samples, enhance the detection capability for complex crowd scenarios, and ultimately improve the accuracy of CC.

2 Methods and materials

2.1 Design of CC model based on weakly supervised learning

In sports venue management, CC is a crucial task. With the increase in sports activities and the expansion of audience size, accurately counting the number of attendees is of great significance for ensuring safety, optimizing resource allocation, and improving venue management efficiency [16]. However, traditional CC methods based on fully supervised learning heavily depend on a substantial quantity of accurately labeled data for model training, and obtaining such high-quality data is extremely difficult in complex and ever-changing scenarios such as sports venues. In addition, changes in crowd density, lighting conditions, occlusion, and other factors may have a significant impact on the counting results, leading to limited generalization ability of the model. Meanwhile, fully supervised learning methods have poor adaptability to new scenarios or abnormal situations, making it difficult to adjust and optimize in real time, thus failing to meet the high-precision and real-time requirements for CC in sports venue management [17-19]. A WSLCC model is proposed to address the above issues. In this study, weakly supervised learning employs point annotations. Specifically, for each image containing a group of people, the annotators only mark the positions of some members, rather than precisely counting or annotating the positions of all individuals in the image. The annotation strategy involves randomly selecting a certain percentage of the group members for position marking in each image, which serves as weakly supervised information to guide the model training. Figure 1 illustrates the architecture of the WSLCC model.

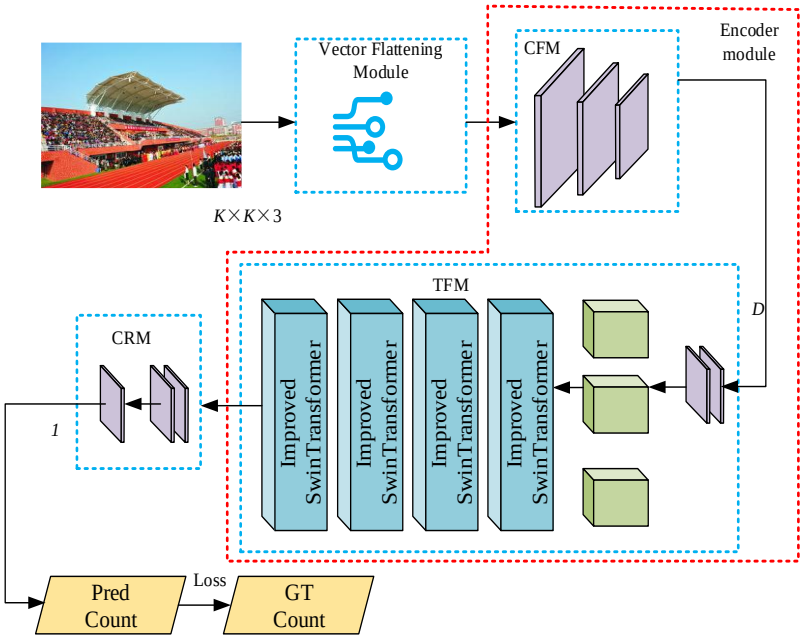

Figure 1: Illustrative representation of WSLCC model structure

In Figure 1, the WSLCC model is a highly integrated system consisting of three core components. The first is the vector flattening module. The primary function of this module is to divide the input image into multiple smaller segments and then transform these image segments into vector-based representations that can be subsequently processed by the model. This step serves as the initial phase of the model's processing pipeline and lays the groundwork for guaranteeing the precision of subsequent analyses. Following this is the encoder module, which holds a pivotal position in conducting in-depth analysis and processing of the vectors. The encoder module is further divided into two sub modules: CFM and TFM. These two sub modules work together to fully utilize their respective advantages and conduct in-depth vector mining, aiming to extract key information and features from images and provide strong support for subsequent CC predictions. Finally, there is the Regression Counting Module (CRM), which is the stage where the model outputs the predicted results. It is based on the features extracted by the encoder module and uses advanced algorithms and techniques to generate corresponding CC predictions. The task of CRM is to ensure the accuracy and reliability of prediction outcomes, thereby providing valuable reference information to users.

The model consists of two primary steps: training and inference. During the training phase, input images are first preprocessed by a vector flattening module, which divides the image into smaller modules and converts them into vector form. These vectors then pass through the CFM and TFM to extract key information and features from the image. The regression counting module, combined with weakly supervised point annotations, calculates the loss between the model's predictions and the actual labels. By continuously optimizing the loss function and adjusting the model parameters, the model gradually learns to accurately count people.

This study employed the Smooth L1 loss function. Compared to the traditional L1 loss function, the Smooth L1 loss function exhibits better differentiability near zero points, allowing the model to adjust parameters more smoothly during training. Additionally, it demonstrates greater robustness against outliers, thereby significantly enhancing the model's prediction accuracy and stability in complex scenarios. The loss function is described by equation (1).

$$\mathrm{L}=\frac{1}{N}\sum_{i=1}^{N}\begin{cases}0.5*\left(C_i^{ES}-C_i^{GT}\right)^2, \text{ if } \left|C_i^{ES}-C_i^{GT}\right|<1\\ \left|C_i^{ES}-C_i^{GT}\right|-0.5, \text{ otherwise}\end{cases} \quad (1)$$

In formula (1), N represents the number of training pictures, C_i^{ES} represents the predicted number of people by the model, and C_i^{GT} represents the real number of people by the model. In the reasoning step: In the reasoning stage, the input image is also processed by vector flattening module, CFM and TFM, and finally the CC prediction result is generated by regression counting module.

In the front-end architecture of the research, in the vector flattening module of the WSLCC model, the input image will first undergo a preprocessing step of flattening, in which the input image blocks are converted into one-dimensional vector sequences. This conversion step is crucial as it not only efficiently retains the spatial details of the image, but also establishes a robust groundwork for the ensuing encoding procedure. The result obtained during the block embedding stage is shown in equation (2).

$$\left\{x_p^i \in R^{K^2\times 3} \mid i=1,2,3\ldots,N\right\} \quad (2)$$

In equation (2), P denotes positional embedding, N means the number of image blocks, and x_p^i is the segmented image block. Subsequently, each segmented

image block undergoes a specific mapping process to be accurately transformed into a latent embedding vector with D-dimensional features for subsequent processing and analysis. The mapping process is achieved by applying a specially designed and trainable linear projection layer, which can transform data from the original space to a new feature space, as shown in equation (3).

$$E \in R^{N \times D} \tag{3}$$

In equation (3), E represents the mapping matrix. To ensure that each segmentation block in the image can fully preserve its original spatial position information, a learnable position embedding mechanism is proposed. The core of this mechanism is to dynamically inject spatial position encoding into the feature vectors of each image sub block. This encoding is not a static parameter generated by traditional fixed equations, but a dynamic vector learned autonomously by neural networks. During the model training process, the position embedding layer will automatically generate embedding values with spatial representation capabilities based on the 2D coordinates of the image blocks. This design allows the model to grasp the relative positional relationships between blocks and recognize their absolute positional information when analyzing the global context through self attention mechanism, as presented in equation (4).

$$e\{n\}=x\{n\}+P\{n\} \tag{4}$$

In equation (4), $x\{n\}$ represents the original feature embedding and $P\{n\}$ represents the positional embedding. $e\{n\}$ represents the final block embedding vector, which is used as input to the encoding layer of the WSLCC model. The pseudocode of the vector flat module is shown in Figure 2.

```
class VectorFlattenModule:
    def __init__(self, d_model: int, patch_size: int):
        # Learnable linear projection layer
        self.projection = Linear(in_dim=patch_size^2 * 3, out_dim=d_model)  # E_proj matrix in Eq(2)

        # Learnable position embeddings (Eq1 & Eq3)
        self.position_emb = nn.Parameter(torch.randn(1, num_patches, d_model))  # E_pos

    def forward(self, x: Tensor) -> Tensor:
        """
        x: input image tensor [B, C, H, W]
        returns: patch embeddings [B, M, d_model]
        """
        # Step 1: Image Patching (Eq1)
        patches = split_into_patches(x, patch_size)  # [B, M, patch_size^2 * C]

        # Step 2: Linear Projection (Eq2)
        patch_emb = self.projection(patches)  # [B, M, d_model]

        # Step 3: Add Position Embedding (Eq3)
        final_emb = patch_emb + self.position_emb  # [B, M, d_model]

        return final_emb

# Helper function implementation
def split_into_patches(x, patch_size):
    B, C, H, W = x.shape
    M = (H * W) // (patch_size^2)  # Number of patches (Eq1)
    x = x.reshape(B, C, H//patch_size, patch_size, W//patch_size, patch_size)
    x = x.permute(0, 2, 4, 1, 3, 5).contiguous()
    x = x.view(B, M, -1)
    return x
```

Figure 2: Pseudo-code diagram of vector flat module

2.2 Encoder module design of WSLCC model

Once the block embedding vectors of the input image have been successfully acquired, the encoder within the WSLCC model steps up to the plate, assuming the crucial task of performing in-depth analysis and processing on these vectors. Leveraging a neural network architecture, it meticulously sifts through the image data, extracting pivotal information and core features. The WSLCC model encoder designed this time consists of two sub modules, CFM and TFM. In the entire sports venue CC, the encoder first uses the CFM to accurately capture and extract high-level crowd semantic information from video frames. Subsequently, this information is fed into an efficient TFM, which can fully utilize its powerful modeling capabilities to comprehensively construct global contextual information and long-range dependencies. The encoder structure is shown in Figure 3.

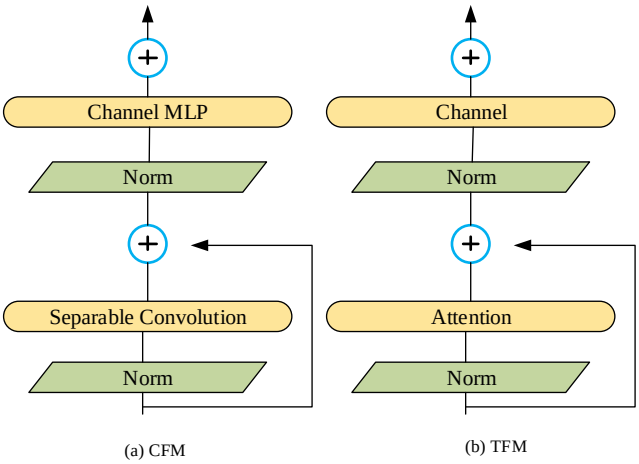

Figure 3: Illustrative representation of encoder structure

In Figure 3, the CFM submodule selects the first ten layers of the VGG16-BN network as feature extractors, leveraging the network's exceptional performance and stability in numerous computer vision tasks. To effectively manage the number of model parameters and extend the network's receptive field, the last two pooling layers of the VGG16-BN network were omitted. This alteration allows the network to capture more detailed image information while preserving efficiency. Consequently, the output feature map of the CFM submodule remains at 1/8 the resolution of the original image, offering more precise and detailed feature information for subsequent CC tasks.

In the CFM sub-module, the first ten layers of the VGG16-BN network were selected as the feature extractor. VGG16-BN is a variant of the VGG16 network that incorporates a Batch Normalization (BN) layer. The VGG16 network consists of multiple convolutional layers, pooling layers, and fully connected layers. The convolutional layers use small kernels (such as 3x3) to progressively extract image features. In the CFM, these first ten layers include multiple convolutional and pooling layers, which work together to perform the feature extraction task. The convolutional layers use convolutional kernels to slide over the image and apply weighted sums, initially extracting low-level local features such as edges and textures. As the number of layers increases, they gradually capture higher-level semantic features like crowd contours and poses. By selecting the first ten layers of VGG16-BN, the CFM controls the number of parameters, expands the receptive field, avoids overfitting, and enhances generalization capabilities. This approach also allows for the capture of rich image details, which is crucial for accurate CC. Ultimately, the CFM outputs feature maps with a resolution one-eighth of the original image, providing precise and rich features to the TFM, which helps it construct global context and long-distance dependencies. The output of the CFM obtained from this is shown in equation (5).

$$C_f = \mathrm{F}_{vgg}(I) \tag{5}$$

In equation (5), C_f represents the output of the CFM and F_{vgg} represents the first ten layers of the VGG16-BN network. Next, the CC model uses the output of CFM as the input of the TFM. Specifically, C_f is first transformed into one-dimensional sequences, and then the sequence is sent to TFM.

In the TFM, the crowd semantic features obtained from CFM are first transformed into one-dimensional sequences. Then, the sequence is sent to TFM for image segmentation and modeling of global context and long-range dependencies. However, traditional Transformer models suffer from problems such as high computational complexity and lack of hierarchical scale feature modeling ability due to calculating self attention at the original image resolution [20-21]. To this end, the study first introduces Shift Window Multi-Head Self Attention (SW-MSA) to improve the traditional Transformer and obtain a Swin Transformer network, whose overall structure is shown in Figure 4.

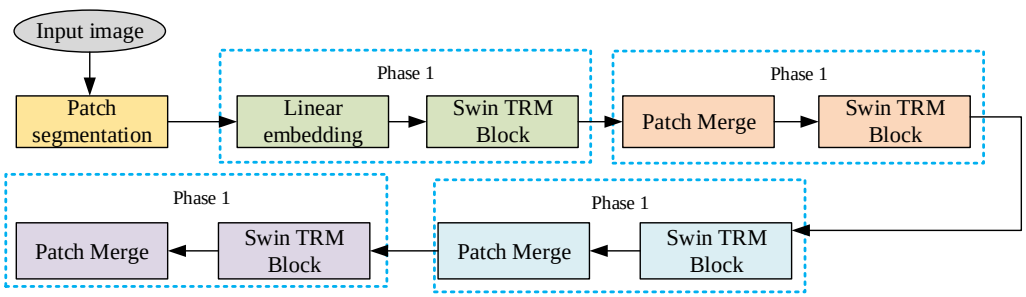

Figure 4: Illustrative representation of Swin Transformer structure

In Figure 4, Swin Transformer is a hierarchical visual Transformer architecture that consists of four progressive stages in its complete structure. At each stage, the image is divided into local regions through patch partitioning, and after linear embedding is mapped into feature vectors, feature extraction is performed by SwinTRM blocks. This structure has the ability to maintain global context modeling while improving computational efficiency. Next, to further enhance the efficiency and performance of the Transformer in handling image data, an investigation is conducted into employing a combination of two layers of Swin Transformer and a convolutional layer to construct an enhanced Swin Transformer block, which served as the primary feature extractor for the Transformer within the TFM.

In the TFM module, the improved SwinTransformer block uses SW-MSA to compute self-attention in its SwinTransformer layer. Specifically, each SwinTransformer layer includes a self-attention module based on SW-MSA, followed by a Multi-Layer Perceptron (MLP). Before the self-attention module and the MLP, there is a LayerNormalization (LN) layer. The structure follows this sequence: LN → SW-MSA → LN → MLP. This design enhances the model's ability to extract features by stabilizing the training process and improving feature extraction capabilities. The improved Swin Transformer block structure is presented in Figure 5.

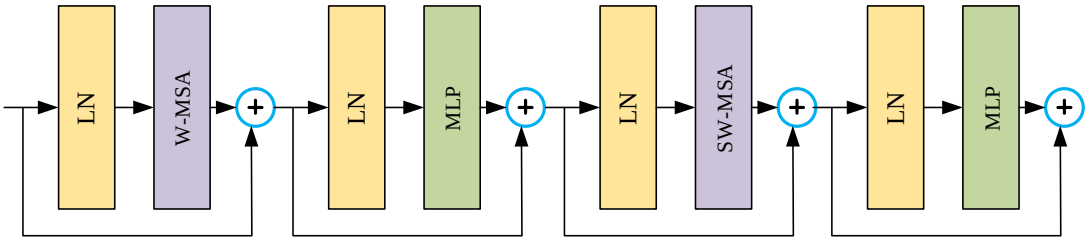

Figure 5: Illustrative representation of improved Swin Transformer block structure

In Figure 5, the improved Swin Transformer block is an enhanced deep learning model component, and its workflow is shown in equation (6).

$$\begin{cases} T_{F_i} = \mathrm{F}_{M_i}\left(T_{F_{i-1}}\right), i = 1,2,3,4 \\ T_{F_{i,j}} = \mathrm{Conv}\left(\mathrm{F}_{STL_{i,j}}\left(\mathrm{F}_{STL_{i,j-1}}\left(T_{F_{i,j-2}}\right)\right)\right), j = 2,4,6,8,\cdots,L \end{cases} \quad (6)$$

In equation (6), F_{M_i} represents the improved Swin Transformer block, and T_{F_i} represents its output features. $\mathrm{F}_{STL_{i,j}}$ represents the Swin Transformer layer at position (i, j), and $T_{F_{i,j}}$ represents its intermediate features. From this, the output feature T_F of the TFM can be obtained as shown in equation (7).

$$T_F = \mathrm{F}_{TFM}\left(\mathrm{Conv}\left(C_f\right)\right) \quad (7)$$

In equation (7), TFM represents the operations of the Transformer module, and $\mathrm{Conv}\left(C_f\right)$ is the input to the TFM after convolution operation on C_f. Conv represents the convolutional layer and F_{TFM} represents the TFM. In the aggregation of feature maps, a simple element-wise addition method is employed. For convolution operations, C_f in $\mathrm{Conv}\left(C_f\right)$ represents the number of channels in the input feature map. The size of the convolution kernel, stride, and padding are determined based on the specific model design and the dimensions of the input feature map, ensuring that the size and number of channels of the feature map after convolution meet the model's requirements. For example, in some cases, the convolution kernel might be set to 3x3, with a stride of 1 and padding of 1, to maintain the spatial dimensions of the feature map.

To prevent overfitting, the study introduced Dropout layers into the model. Specifically, Dropout layers were added after the convolutional layers in the CNN section and after the attention mechanism in the Transformer section. Through experimental optimization, the Dropout rate after the CNN convolutional layers was set to 0.3, and after the Transformer attention mechanism, it was set to 0.2. The introduction of Dropout layers enhanced the model's generalization ability, ensuring better performance across various scenarios. The pseudocode of the encoder module is shown in Figure 6.

```
class WSLCCEncoder:
    def __init__(self):
        self.cfm = VGG16BN_First10Layers()  # Pre-
modified CFM backbone
        self.tfm = SwinBlock(num_layers=2)  # 2-
layer Swin Transformer
        self.conv = Conv2d(256, 512, 3, padding=1)
        self.drop = [Dropout(0.3), Dropout(0.2)]

    def forward(self, x):
        # CFM path (Eq4)
        c = self.drop(self.cfm(x))  # 1/8 res

        # TFM path (Eq5-6)
        t = self.conv(c).flatten(2)
        t = self.drop(self.tfm(t))

        return c + t.view_as(c)  # Feature fusion

class SwinBlock:
    def __init__(self, num_layers=2):
        self.layers = nn.ModuleList([
            nn.Sequential(
                LayerNorm(),
                ShiftedWindowMSA(8),
                LayerNorm(),
                MLP(512)
            ) for _ in range(num_layers)
        ])

    def forward(self, x):
        for layer in self.layers:
            x = layer(x)
        return x
```

Figure 6: Pseudo-code diagram of the encoder module

After constructing the WSLCC model, to promote the efficiency of sports venue management, a sports venue resource information management platform based on a CC model is designed based on this model. The platform uses camera equipment to calculate the crowd density of each sports event area through a CC model, and uploads it to the database. Users can obtain the location of the shared sports equipment cabinet through the client, borrow the equipment, and return it within the specified time. Figure7 presents the comprehensive structure of the platform.

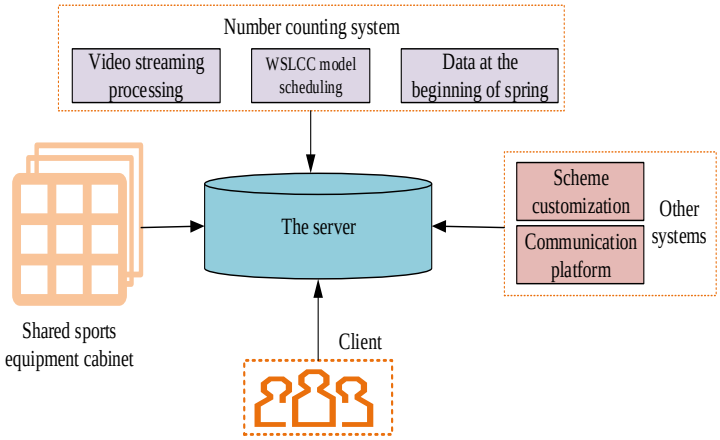

Figure 7: Illustrative representation of sports venue management platform architecture

In Figure 7, the platform mainly consists of two parts: a CC system and a shared sports equipment cabinet. The database architecture of the platform is shown in Table 2.

Table 2: Database architecture of sports venue management platform

Table name	Field name	Data type
Video_Info	video_id	int
	video_path	varchar
	start_time	datetime
	end_time	datetime

Crowd_Density_Data	density_id	int
	video_id	int
	region_id	int
	density_value	float
	timestamp	datetime
Equipment_Cabinet_Info	cabinet_id	int
	location	varchar
	equipment_list	text
	qr_code	varchar
User_Info	user_id	int
	username	varchar

The platform focuses on the management of sports venues, processing video streams, CC, and equipment borrowing and returning data. For video streams, cameras capture venue footage, which is then decoded and frame extracted by the processing module. The image frames and video information are stored in the video information table. For CC, the server receives the image frames and uses the WSLCC model to calculate the crowd density in each sports area. The results, along with the video ID, are stored in the crowd density data table, allowing the client to request real-time displays. When users borrow or return equipment, they scan a code, and the server returns the equipment cabinet information. After the user completes the operation, the server updates the equipment status and logs the user's operation history. Among them, the CC system mainly includes three modules: video stream processing, counting model scheduling, and data storage. In this system, video streams are captured by camera equipment, images are captured and uploaded to the server. The server calls the WSLCC model for calculation, saves the crowd density data to the database, and the client is able to acquire up-to-date data regarding the headcount. In the shared sports equipment cabinet section, each equipment cabinet is equipped with a unique QR code. After users scan the QR code through the client, the server retrieves the equipment information from the database based on the QR code information and returns it to the client for users to choose and use.

3 Results and discussion

3.1 Performance testing of CC model

To confirm the capability of the proposed WSLCC model, two commonly-used public datasets in the area of CC, ShanghaiTech and UCF_CC_50, were selected for model testing in the experiment. The ShanghaiTech dataset contains numerous crowd images with different scenes and densities, which are appropriate for training and assessing the effectiveness ofCC models.UCF_CC_50 is a challenging small-scale dataset with extremely high and variable crowd density in its images, commonly used to test the counting ability of models in high-density crowd scenes.

In this study, a series of preprocessing steps were employed to effectively utilize datasets such as ShanghaiTech for training and testing the WSLCC model. Before inputting the images into the model, the study first normalized all the images, scaling the pixel values from [0255] to the range of [0,1] to accelerate model training and improve numerical stability. Meanwhile, in order to meet the input size requirements of the model, the study uniformly adjusted all images to 224×224 pixels. For the ShanghaiTechPartA dataset, due to the high density of people and complex scenes in its images, data augmentation techniques were used to increase the diversity of the data. Specifically, it included operations such as random cropping, random flipping, and random rotation. Random cropping can randomly select a sub region in the image with a cropping ratio of [0.8,1.0], which allows the model to learn the crowd characteristics of different local regions. Random flipping performs horizontal flipping with a 50% probability, with a random rotation angle range of [-15°, 15°], further enriching the perspective and variation of the training data. The image resolution and crowd density of the ShanghaiTechPartB dataset differed significantly from those of PartA, with relatively open scenes and lower crowd density. In response to this characteristic, in addition to the general operations mentioned above, the study also adjusted the brightness and enhanced the contrast of the image during preprocessing. The brightness adjustment varied randomly within the range of [0.8, 1.2], and the contrast enhancement used histogram equalization method to highlight the crowd characteristics in the image, enabling

the model to better learn crowd patterns under different densities and scenes.

In this study, the Adam optimizer was used to train the WSLCC model. The initial learning rate was set to 1e-5, and the weight decay coefficient was set to 0.0001. To further enhance the model's performance, a stepwise learning rate scheduling strategy was implemented. After every 10 epochs, the learning rate would be reduced by γ=0.1 times. To prevent overfitting, the study adopted an early stopping strategy. During training, the study monitored the Mean Absolute Error (MAE) on the validation set. If the MAE on the validation set does not decrease for five consecutive epochs, the training is halted. At this point, the model with the lowest MAE on the validation set was selected as the final model. To enhance the model's generalization ability, various data augmentation techniques were applied to the input images during training. These techniques included random cropping, random horizontal flipping, and random rotation. The cropping ratio was randomly selected between 0.8 and 1.0, resulting in sub-images of size 224x224. The probability of random horizontal flipping was 50%, and the angle of random rotation ranged from −15° to 15°, also with a 50% probability.

The current mainstream CC models were selected as the comparison models for the experiment, namely Contextual Scale Regression Network (CSRNet), Multi-column CNN (MCNN), and Locally Scale Aware CNN (LSC-CNN). MAEand Root Mean Square Error (RMSE) were selected as evaluation metrics for model performance. Table 3 displays the experimental operating conditions and parameter configurations.

Table 3: Experimental operating conditions and parameter configurations

Experimental environment		Set the item
JRE	CUDA edition	11.4.0
	CPU	NVIDIARTX4090Ti
	Internal memory	16.00GB
	Batch size	8
	Learning rate initial value	1e-5
	Optimizer	Adam
	Software environment	MatlabR2018a
	Weight decay	1e-4
	Iterations	3000
TFM hyperparameter	SEmbed_dim	4
	Window_size	0.125
	Depths	[8,8,8,8]
	Num_heads	[8,8,8,8]

This study chose Matlab R2018a over TensorFlow or PyTorch because Matlab has a rich set of built-in functions and toolboxes, which offer significant advantages in data processing and visualization. It can efficiently perform data preprocessing and result presentation. Moreover, its concise syntax makes it more convenient for implementing specific algorithms, thus meeting the project's needs for rapid development. Firstly, the evaluation metrics used in the study included accuracy, recall, and Mean Average Precision (mAP). The research model was trained on the training set (TS) and validation set (VS) of ShanghaiTech's SHTech Part_Section, as shown in Figure 8.

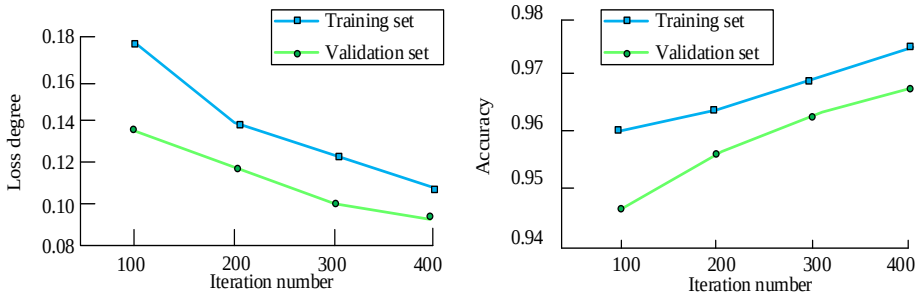

Figure 8: Training results of WSLCC model

As shown in Figure 8 (a), the loss of WSLCC model in the TS and VS gradually decreased with the iteration of learning times. When the last training ended, the loss rate in the TS decreased from 0.1800 to 0.1084, and the loss rate in the VS decreased from 0.1362 to 0.0915, indicating a continuous improvement in generalization ability. As shown in Figure 8 (b), the WSLCC model had an accuracy of over 90% on both the TS and VS, and as the number of iterations increased, the final accuracies were 96.93% and 98.21%, respectively. The results indicated that the WSLCC model performed well in training and could be used for CC tasks.The test results of MCNN, CSRNet, LSC-CNN, and the WSLCC model proposed by the research on the SHTech Part_A test set are shown in Figure 9.

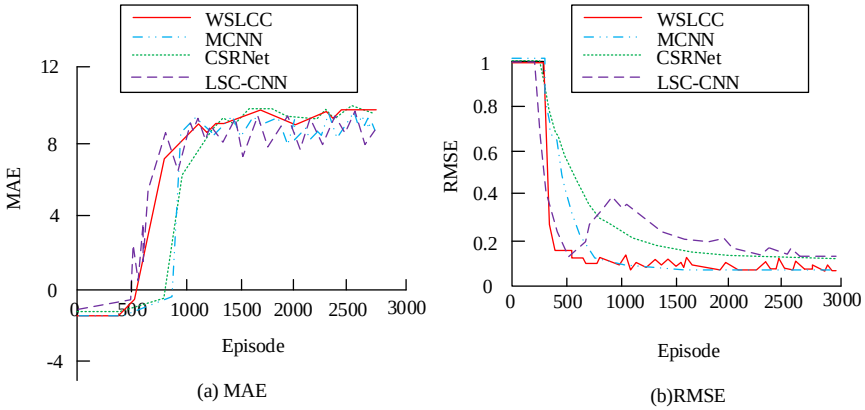

Figure 9: WSLCC model test results

In Figure 9(a), the MAE of the WSLCC model rapidly decreased during training and stabilized at approximately 4.5, which was lower than MCNN (around 10), CSRNet (around 8), and LSC-CNN (around 7). This indicated that the WSLCC model had higher accuracy in counting. In Figure 9(b), the RMSE was used to measure the loss. The RMSE of the WSLCC model eventually dropped to about 0.25, while the RMSE of MCNN, CSRNet, and LSC-CNN remained above 0.3. This suggested that the WSLCC model performed better in reducing prediction errors and enhancing stability.

Next, the performance of MCNN, CSRNet, LSC-CNN, and WSLCC models was tested on the test sets of ShanghaiTech and UCF_CC_50 datasets. The MAE and RMSE test results of each model are presented in Table 4.

Table 4: MAE and RMSE test outcomes for various models

DS	DS	MCNN	CSRNet	LSC-CNN	WSLCC
MAE	SHTechPart_A	66.1±3.0	60.9±2.8	61.2±2.9	60.5±2.6
	SHTechPart_B	9.3±1.2	8.9±1.1	8.2±1.0	8.1±0.9
	UCF_CC_50	272.2±15.0	248.3±14.5	211.6±13.0	188.2±12.0
DS	DS	MCNN	CSRNet	LSC-CNN	WSLCC
RMSE	SHTechPart_A	105.1±4.0	93.8±3.8	94.3±3.9	90.7±3.6
	SHTechPart_B	16.1±1.4	5.2±1.3	13.3±1.2	13.2±1.1
	UCF_CC_50	395.3±20.0	64.5±18.5	317.3±16.0	300.3±15.0

A meticulous examination of the data presented in Table 4 reveals unequivocally that the WSLCC model exhibited exceptional performance across all tests. Among them, on the SHTech Part_A dataset, the MAE of the WSLCC model was 60.5, and the RMSE was 90.7, which was the lowest compared to other models. This indicated that the WSLCC model had the smallest deviation between the forecasted results and the real values on this dataset, and had higher accuracy. On the SHTech Part_S dataset, the WSLCC model also performed well, with MAE and RMSE of 8.1 and 13.2, respectively, both lower than other comparison models, demonstrating its stability and reliability in different scenarios. On the UCF_CC_50 dataset, the MAE of the WSLCC model was 188.2 and the RMSE was 300.3. Despite the challenges of this dataset, the WSLCC model still achieved better results than other models. Overall, regardless of the dataset, the WSLCC model demonstrated optimal performance, fully demonstrating its effectiveness and superiority in CC tasks. Through t-test and other statistical tests, combined with the standard deviation in the table, the performance improvement of WSLCC model was statistically significant, and it could improve the counting accuracy in the actual CC scenario, and had practical impact.

In the above results, the WSLCC model performed exceptionally well, primarily due to two key factors. Firstly, the architectural innovation by integrating advanced CFM and TFM modules. CFM could accurately capture high-level semantic information in video frames, while TFM comprehensively constructed global context and long-range dependencies, enhancing the model's ability to perceive and process complex scenes. Secondly, the model demonstrated strong adaptability across various datasets, showing the lowest error rate on the SHTech dataset with minimal deviation between predictions and actual values. It also outperformed other models on the UCF_CC_50 dataset, which was more challenging. However, the model might have limitations such as higher computational complexity and higher hardware resource requirements, potentially limiting its performance in real-

time applications.

To more comprehensively evaluate the performance of the WSLCC model, the study conducted multiple experiments with different training and test set divisions. The evaluation metrics included R^2, AE, and RMSE, and the 95% confidence intervals for each metric were calculated to more accurately reflect the model's performance fluctuations. The results are presented in Table 5.

Table 5: Test results of different test/training set division methods of the model

Training set/test set division ratio	R^2 fraction	AE	AE95% confidence interval	RMSE	RMSE95% confidence interval
80%/20%	0.75	12.5	[10.2,14.8]	15.3	[13.1,17.5]
75%/25%	0.72	13.2	[11.0,15.4]	16.1	[13.9,18.3]
70%/30%	0.70	14.0	[11.8,16.2]	17.0	[14.8,19.2]
85%/15%	0.78	11.8	[9.6,14.0]	14.5	[12.3,16.7]
82%/18%	0.76	12.2	[10.0,14.4]	15.0	[12.8,17.2]

In Table 5, the WSLCC model exhibited varying performance on the UCF_CC_50 dataset under different training/test set configurations. The R^2 score ranged from 0.70 to 0.78, and the AE and RMSE also showed varying degrees of fluctuation. This suggested that the smaller size of the UCF_CC_50 dataset made the impact of different data splits on model performance more pronounced. Although the model performed relatively well with some configurations, the overall instability highlighted the importance of conducting multiple trials on small-scale datasets to more comprehensively and accurately assess the model's generalization ability and reliability.

To verify the effectiveness of each component of WSLCC model, three ablation experiments were carried out on UCF_CC_50 data set, including model parameter ablation, model component ablation and loss function ablation. The results of ablation experiments are shown in Figure 10.

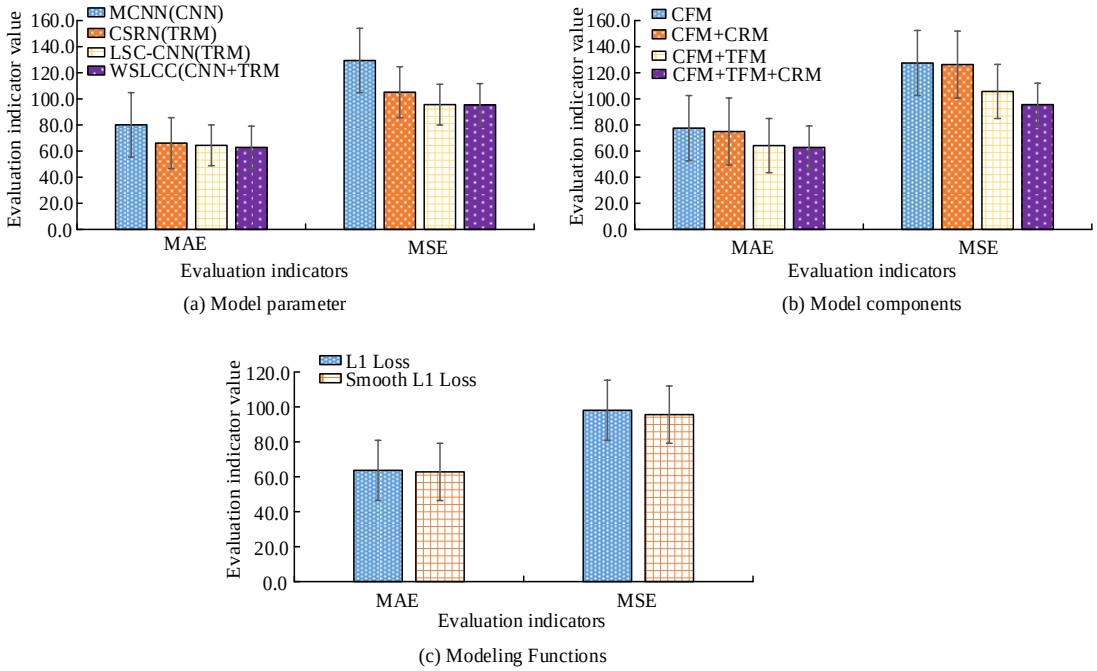

Figure 10: Experimental outcomes of WSLCC model ablation

Figure 10 (a) indicates the ablation results of the model parameters, Figure 10 (b) indicates the ablation outcomes of the model components, and Figure 10 (c) indicates the ablation results of the model LF. In the model parameter ablation experiment shown in Figure 10(a), the parameters of the MCNN, CSRNet, and LSC-CNN models were set to 29%,33%, and 27% of the WSLCC model's parameters, respectively. This experimental design, which set the parameter sizes of existing models to a certain ratio of the WSLCC model's parameters, aimed to ensure a fair comparison in terms of model complexity. By controlling the parameter sizes, the study could better

assess the effectiveness of the WSLCC model architecture itself, avoiding the impact of parameter size differences. Although this design was not standard, it helped focus on the impact of the model architecture on performance. The experiment was conducted on the SHTech Part_A dataset. As shown in the figure, the WSLCC model had the lowest MAE and RMSE values, at 62.8 and 95.4, respectively, which were 2.5% and 0.2% lower than those of the LSC-CNN model. The results indicated that the research model could achieve lower counting errors while using fewer parameters. As shown in Figure 10(b), overall, the model's performance gradually improved with the addition of each module. Specifically, adding the CRM module to the CFM resulted in a slight decrease in MAE and RMSE. Adding the TFM to the CFM significantly enhanced the model's performance, with MAE and RMSE decreasing by 17.3% and 17.1%, respectively. The research model combining CFM, CRM, and TFM had the lowest MAE and RMSE values, at 62.8 and 95.6, respectively, indicating the best performance. As shown in Figure 10(c), the loss function used in the study reduced both MAE and RMSE by 2.5% and 1.4%, respectively. This differed from the WSLCC model on the SHTech Part_A dataset, where MAE (60.5) and RMSE (90.7) were higher. This difference was due to variations in the datasets and experimental conditions, which affected the model performance metrics in the two experiments. The above outcomes indicated that the developed WSLCC model performed the best in all aspects and could efficiently complete the CC task.

To assess the significance of these metric differences, statistical tests were conducted. The results showed that the WSLCC model significantly outperformed other models in terms of MAE and RMSE (p<0.05), indicating a genuine and reliable performance improvement. The ablation results are detailed in Table 6.

Table 6: Ablation experiment data

Model configuration		MAE	RMSE	p
Parameter ablation	MCNN	80.2	132.1	p< 0.05
	CSRNet	68.4	116.7	p< 0.05
	LSC-CNN	80	98.5	p< 0.05
	WSLCC(CNN-TRM)	62.8	95.4	/
Component ablation	CFM+CRM	60	93	p < 0.05
	CFM+TFM	52	79	p< 0.05
	WSLCC	50	78	/

Regarding the consistency of changes across different datasets, although this experiment only showcased the ablation results on the UCF_CC_50 dataset, similar experiments were conducted on other datasets like SHTech in earlier studies. The trend of model performance improvement was consistent, with improvements in metrics such as MAE and RMSE after adding specific modules. However, the extent of metric reduction varied across different datasets, possibly due to differences in scene characteristics and crowd density distribution. For instance, on the more complex UCF_CC_50 dataset, where the crowd density was higher, the WSLCC model showed a more significant performance improvement compared to other models. In contrast, on the simpler SHTech dataset, the improvement was less pronounced but still maintained a good performance advantage.

3.2 Application analysis of sports venue management platform

To verify the effectiveness of the WSLCC model in sports venue management, a practical application analysis was conducted on the proposed sports venue management platform. The study first integrated the WSLCC model into the proposed sports venue management platform based on CC model, and tested its performance using the LoadRunner tool. The minimum response time, maximum response time, CPU and memory usage of the platform under different concurrent user numbers are shown in Table 7.

Table 7: Performance test results of sports venue management platform

Number of concurrent users	100		300		500	
	This platform	Traditional manual method	This platform	Existing platform	This platform	Existing platform

Minimum response time/s	First	0.667	2.5	0.723	1.8	0.956	2.2
	Second	0.592	2.3	0.684	1.7	0.903	2.0
	Third	0.637	2.4	0.745	1.75	0.924	2.1
Maximum response time/s	First	0.942	3.5	1.023	2.8	1.543	3.2
	Second	0.927	3.4	1.132	2.7	1.493	3.1
	Third	0.949	3.45	1.079	2.75	1.509	3.15
CPU and memory footprint/%	First	7.5	15	26.7	35	47.2	50
	Second	6.9	14	27.3	34	40.8	48
	Third	7.1	14.5	24.6	33	39.7	47

In Table 7, the overall performance of the platform and the baseline system improved as the number of concurrent users increased. The minimum and maximum response times, as well as CPU and memory usage, all increased. When the number of concurrent users was 500, the platform's minimum response time was 0.956 seconds, a significant improvement over the 2.2 seconds for both the traditional manual method and the existing platform. The maximum response time was 1.543 seconds, also outperforming the traditional manual method and the existing platform. In terms of CPU and memory usage, the platform's usage rate was 47.2% at 500 concurrent users, which was lower than the 50% used by the existing platform and the higher usage of the traditional manual method (assuming the traditional manual method has a higher usage). This indicated that the sports venue management platform proposed in this study offered significant performance advantages over traditional manual methods and existing platforms, providing more stable, reliable, and efficient services for sports venue management. Then, the research introduced the platform into a sports center in a particular city to monitor foot traffic density in real-time, optimize venue scheduling, and allocate resources effectively. The practical implementation test at the sports center was conducted from December 2024 to April 2025, lasting for a total of five months. During the test, a combined approach of manual counting and sensor-data collection was utilized to obtain accurate and reliable information. Specifically, designated personnel were responsible for performing regular manual counts in key areas of the sports center, carefully recording the number of people in each area. Meanwhile, sensor devices installed on-site collected real-time data on venue usage and crowd density. The manual counts and sensor data were cross-verified to ensure the accuracy and reliability of the data. The key evaluation indicators for the three months before and after the application of the statistical platform are shown in Figure 11.

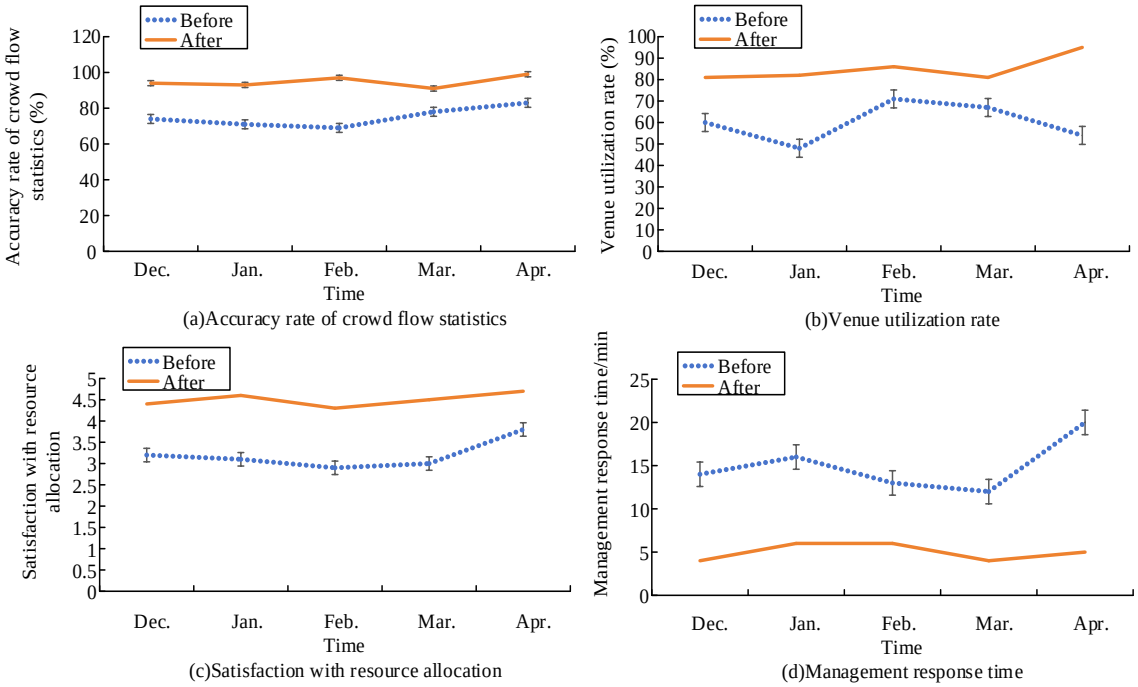

Figure 11: Comparison results of indicators before and after platform application

As depicted in Figure 11, the sports venue management platform, which is grounded in the CC model proposed by this study, has notably enhanced overall management efficiency. Among them, the accuracy of crowd flow statistics increased from 75.5% to 95.1%, ensuring the reliability of the data. The utilization rate of the venue increased from 60.2% to 85.4%, optimizing the use of resources. The satisfaction score for resource allocation increased from 3.2 to 4.5, indicating a significant increase in user recognition of venue allocation. The management response time was reduced from 15.4 minutes to 5.3 minutes, improving operational efficiency. Overall, the platform improved management efficiency through intelligent management, providing strong support for the scientific operation of sports venues.

To further validate the robustness of CC methods based on weakly supervised learning in sports venue management, an additional practical application test was conducted at a large football stadium. During the event, the stadium experienced high foot traffic and complex crowd movement, which significantly differed from the previous test dataset scenarios. The test results are presented in Table 8.

Table 8: Further evaluation results of model robustness

Indicator/Crowd density zoning	Entry area	The audience area	Food and beverage area	Export area
Average count error rate	8.2%(7.5%-8.9%)	7.5%(6.8%-8.2%)	9.1%(8.3%-9.9%)	8.7%(7.9%-9.5%)
Maximum response time (s)	1.2(1.1-1.3)	1.0(0.9-1.1)	1.3(1.2-1.4)	1.1(1.0-1.2)
Minimum response time (s)	0.3(0.2-0.4)	0.2(0.1-0.3)	0.4(0.3-0.5)	0.3(0.2-0.4)
CPU occupancy (%)	32(30-34)	28(26-30)	35(33-37)	30(28-32)
Memory occupancy rate (%)	45(42-48)	40(38-42)	48(45-51)	43(40-46)

As shown in Table 8, the CC model performed well across all areas of the football stadium, regardless of crowd density. The confidence intervals for the average counting error rate indicated that the model's counting errors were relatively stable and low in each area, indicating that the model could accurately count people in various functional areas. The confidence intervals for response time showed that the model's feedback speed remained within a reasonable and stable range. The confidence intervals for CPU and memory usage also showed that these resources were used stably and at an acceptable level, without placing excessive strain on the existing sports venue system. This demonstrated that the model was robust and practical in complex sports venue scenarios.

4 Conclusion

To address the issues of low accuracy and poor adaptability faced by CC methods in real-world scenarios, a WSLCC model was proposed. This model first utilized

an advanced CFM module to accurately capture and extract high-level semantic information of the crowd in video frames, and then sent it into an efficient TFM module to comprehensively construct global contextual information and long-range dependencies using its powerful modeling capabilities, significantly improving CC performance under weakly supervised conditions. The sports venue resource information management platform designed based on the WSLCC model effectively improved the adaptability and accuracy of CC methods in sports venue management, significantly enhancing management efficiency. The accuracy of CC increased from 75.5% to 95.1%, the venue utilization rate has increased from 60.2% to 85.4%, the satisfaction score of resource allocation increased to 4.5, and the management response time was shortened to 5.3 minutes. On multiple datasets, the WSLCC model also demonstrated excellent performance, with MAE of 60.5 and RMSE of 90.7 on the SHTech Part_SA dataset. On the SHTech Part_S dataset, the MAE and RMSE were 8.1 and 13.2, respectively. On the UCF_CC_50 dataset, the MAE was 188.2 and the RMSE was 300.3. However, the model lacked adaptability to complex scenes and occlusion situations, and the counting accuracy is limited by the sparsity of annotated data.

In the future, more robust feature representation methods can be further studied, considering the introduction of 3D input to better handle occlusion problems and utilize 3D information to more accurately perceive the spatial distribution of crowds. Meanwhile, exploring multi-camera fusion technology to integrate information from different perspectives and enhance the model's ability to understand complex scenes. In addition, it is necessary to continuously explore more efficient data annotation and enhancement techniques to improve the generalization performance and counting accuracy of the model under limited annotated data.

References

[1] Deng L, Zhou Q, Wang S. Deep learning in crowd counting: A survey. CAAI Transactions on Intelligence Technology, 2024, 9(5): 1043-1077. https://doi.org/10.1049/cit2.12241

[2] Gao M, Souri A, Zaker M. A comprehensive analysis for crowd counting methodologies and algorithms in Internet of Things. Cluster Computing, 2024, 27(1): 859-873. https://doi.org/10.1007/s10586-023-03987-y

[3] Gouiaa R, Akhloufi M A, Shahbazi M. Advances in convolution neural networks-based crowd counting and density estimation. Big Data and Cognitive Computing, 2021, 5(4): 50. https://doi.org/10.3390/bdcc5040050

[4] Tian X, Hiraishi H. Design of crowd counting system based on improved CSRNet. Artificial Life and Robotics, 2025, 30(1): 3-11. https://doi.org/10.1007/s10015-024-00993-0

[5] Cheng J, Xiong H, Cao Z. Decoupled two-stage crowd counting and beyond. IEEE Transactions on Image Processing, 2021, 30: 2862-2875. doi: 10.1109/TIP.2021.3055631.

[6] Xu Z, Jain D K, Shamsolmoali P. Slime Mold optimization with hybrid deep learning enabled crowd-counting approach in video surveillance. Neural Computing and Applications, 2024, 36(5): 2215-2229. https://doi.org/10.1007/s00521-023-09083-x

[7] Song B, Sheng R. Crowd counting and abnormal behavior detection via multiscale GAN network combined with deep optical flow. Mathematical Problems in Engineering, 2020, 2020(1): 6692257. https://doi.org/10.1155/2020/6692257

[8] Liu Z, Yuan R, Yuan Y. A sensor-free crowd counting framework for indoor environments based on channel state information. IEEE Sensors Journal, 2022, 22(6): 6062-6071. doi: 10.1109/JSEN.2022.3144454.

[9] He X, Wang R, Cao T. Fusion CNN-Transformer Model for Target Counting in Complex Scenarios. Informatica, 2025, 49(12): 2141-2149. https://doi.org/10.31449/inf.v49i12.7315

[10] Khan K, Khan R U, Albattah W. Crowd counting using end-to-end semantic image segmentation. Electronics, 2021, 10(11): 1293. https://doi.org/10.3390/electronics10111293

[11] Sindagi V A, Yasarla R, Patel V M. Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. IEEE transactions on pattern analysis and machine intelligence, 2020, 44(5): 2594-2609. doi: 10.1109/TPAMI.2020.3035969

[12] Pang Y, Ni Z, Zhong X. Federated learning for crowd counting in smart surveillance systems. IEEE Internet of Things Journal, 2023, 11(3): 5200-5209. doi: 10.1109/JIOT.2023.3305933

[13] Oh M, Olsen P, Ramamurthy K N. Crowd counting with decomposed uncertainty//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 11799-11806. https://doi.org/10.1609/aaai.v34i07.6852

[14] Vivekanandam B. Speedy image crowd counting by light weight convolutional neural network. Journal of Innovative Image Processing, 2021, 3(3): 208-222. https://doi.org/10.36548/jiip.2021.3.004

[15] Chen J, Wang K, Su W. SSR-HEF: Crowd counting with multiscale semantic refining and hard example focusing. IEEE Transactions on Industrial Informatics, 2022, 18(10): 6547-6557. doi: 10.1109/TII.2022.3160634.

[16] Sajid U, Sajid H, Wang H. Zoomcount: A zooming mechanism for crowd counting in static images. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30(10): 3499-3512. doi: 10.1109/TCSVT.2020.2978717.

[17] Wang Y, Ma Z, Wei X. Eccnas: Efficient crowd counting neural architecture search. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2022, 18(1s): 1-19. https://doi.org/10.1145/3465455

[18] Wang Q, Lin W, Gao J. Density-aware curriculum learning for crowd counting. IEEE Transactions on Cybernetics, 2020, 52(6): 4675-4687. doi:

10.1109/TCYB.2020.3033428.

[19] Pandey A, Pandey M, Singh N. KUMBH MELA: a case study for dense crowd counting and modeling. Multimedia Tools and Applications, 2020, 79(25): 17837-17858. https://doi.org/10.1007/s11042-020-08754-4

[20] Wang Q, Han T, Gao J. Neuron linear transformation: Modeling the domain shift for crowd counting. IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(8): 3238-3250. doi: 10.1109/TNNLS.2021.3051371

[21] Zhang T, Yang X. Energy-Saving Design of Smart City Buildings Based on Deep Learning Algorithms and Remote Sensing Image Scenes. Informatica, 2024, 48(19): 114-11. https://doi.org/10.31449/inf.v48i19.6049