Optimization of Video Stability Technology by Integrating Tiny-Res-PWNet Model

Wenji Zhong[1], Liping Wu[2*]
[1]College of Information Engineering, Guangxi Vocational College of Water Resources and Electric Power, Nanning 530023, China
[2]Big Data Academy, Guangxi Vocational and Technical College, Nanning 530226, China
E-mail: wlp20232023@126.com
[*]Corresponding author

Keywords: image stabilization technology, residual module, warping field, structural similarity, stability

Received: May 11, 2024

This study proposes an improved pixel by pixel stabilization network to achieve high-quality video stabilization. Fourier spectrum constraints and local motion constraints are added to the output structure of the image stabilization network. Meanwhile, affine matrix parameters are optimized. Then, the transformation parameters output by the network are more similar, thereby reducing the difficulty of learning the overall jitter pattern of the image. In addition, this study replaces the encoder convolutional layer with a residual module and combines feature fusion to process feature information extracted from different network layers. This can achieve optimization and lightweight processing of pixel-by-pixel stable network models. The results showed that the stability evaluation index of the improved pixel by pixel stable network model increased by about 3.7% compared to the previous pixel by pixel stable network model. The parameter counts of the pixel-by-pixel stable network model encoder fused with residual module was reduced by 12.1%. compared to the pixel-by-pixel stable network model encoder. The model size was reduced by 11.7%, the floating-point operation was reduced by 13.2%, and the running frame rate was increased by 5.6%. The lightweight pixel by pixel stable network model achieved a frame rate of 131.2 at high-performance operation, far higher than the 83.1 of the pixel-by-pixel stable network model. The outcomes showcase that the network model is an effective optimization method for video stabilization technology and can be applied to many real-time video processing scenarios. This helps to improve the technical level and application effectiveness in this field.

Povzetek: Razvito je izboljšano omrežje za stabilizacijo videa na ravni slikovnih točk, ki vključuje Fourierjeve spektralne omejitve in lokalne omejitve gibanja. Rezultati kažejo, da izboljšan model dosega 3,7 % boljšo stabilnost in zmanjšano velikost modela za 11,7 %, kar predstavlja pomemben doprinos k optimizaciji video stabilizacijske tehnologije.

1 Introduction

With the rapid development of digital media technology, video has become one of the important ways for people to record and share their lives [1]. However, many people inevitably encounter shaking problems when shooting videos due to the popularity and convenience of various camera devices. This leads to a decrease in video quality and poor viewing experience [2]. Therefore, video stabilization technology has become an important means to solve this problem. In the past few decades, many video stabilization methods were presented and achieved certain results [3]. Early methods were mainly based on traditional image processing techniques, such as motion estimation-based methods and image block matching-based methods. However, these methods are often limited by high computational complexity and poor robustness, which perform poorly in practical applications [4]. Recently, as the boost of deep learning technology, video stabilization methods based on Convolutional Neural Network (CNN) have made significant breakthroughs [5]. Among them, the Pixel-Wise Stable Network (PWSNet) model has good stability and real-time performance. However, in practical applications, the PWSNet model still has some problems. In addition, the stability and robustness of network models in handling videos under different scenes and lighting conditions also need to be further optimized. Therefore, the study aims to optimize and improve the PWSNet model to provide an efficient and accurate video stabilization method, providing users with a better viewing experience. The study consists of four parts. The first is a summary of the relevant research. The second is the optimization and improvement methods of video stabilization technology, which are verified in the third part. The fourth is a summary of the entire study.

2 Related works

Video stabilization technology is used to suppress jitter and vibration in videos. The Son team proposed an effective recursive video deblurring network. It improved the motion estimation accuracy between blurred frames through effectively aggregating information from multiple video frames. They solved motion estimation errors by using pixel volumes containing candidate sharpened pixels. The experiment showcased that compared with traditional deep learning methods, this method improved efficiency by 13% [6]. The Chen team proposed an end-to-end training method that integrated deep learning and state space models for estimating and predicting the state space model of physical systems. The results showed that this method could leverage the relative advantages of deep neural networks and demonstrate effectiveness in estimation and prediction of many physically challenging tasks [7]. Zhou et al. proposed a unified motion correction and denoising resistance network for generating motion compensated low noise images from low-dose gated PET data. The experiment showcased that the network could directly generate accurate motion estimates from low-dose gated images and produce high-quality motion compensated low noise reconstruction [8]. Wang's team proposed a real-time dynamic vision system to achieve accurate pose estimation of cameras in indoor dynamic environments. The system utilized geometric motion removal modules and template-based motion removal modules to process dynamic feature points. They found complete dynamic regions with the help of depth image clustering. The outcomes showcased that the effectiveness of the system could reach 80% [9]. The Asad team proposed a spatial and temporal feature learning method based on video equidistant sequence frames. This method combined the multi-level features of two consecutive frames extracted from the top and bottom layers of CNN to consider motion information. The experiment showcased that the accuracy could reach 85% [10].

Recently, deep learning technology has made significant progress in video stabilization. Deep learning models could automatically learn feature representations of images and videos by learning a large amount of video data and could better capture complex motion patterns. Shahbazi et al. utilized a recursive neural network-based Long Short-Term Memory (LSTM) architecture for incorporating motion features into a single object tracker. A new motion model was trained to predict the position of the target in each frame. The results indicated that the motion model had low computational cost and was in line with the basic tracking performance [11]. The Iraei group proposed a new deep learning algorithm that estimated the fuzzy kernels through CNN. Then, objects were tracked through particle filters and the probability distribution of motion information obtained through kernel estimation. The experiment showcased that compared with existing technologies, this method could improve tracking accuracy by 10% [12]. The Liu team proposed a fault detection method based on high-dimensional features of video image depth. This method selected deep and highly sensitive features with a large amount of fault information as the features to be detected. Euclidean distance was used for fault detection and moving average window function for reducing sudden noise interference. The experiment showcased that the detection efficiency of this method could reach 90% [13]. Chen and his team members proposed a video-based action recognition network that used channel attention mechanism in residual units to learn the action features of each view. The results showed that the accuracy of this method could reach 91% [14]. Liu et al. proposed a dynamic spatiotemporal network to integrate spatiotemporal information. Under the guidance of coarse saliency maps, features and decoders were modified through spatial attention to obtain the final saliency map. The experiment showcased that the accuracy of this method in extracting motion features could reach 90% [15]. The summary table of related works is shown in Table 1.

Table 1: Summary of related works

Field	Researchers	Research content	Research resultxity	Index
Video stabilization technology	Son et al [6]	An effective recursive video deblurring network	Enhance motion estimation accuracy.	Computational efficiency is improved by 13%
	Chen et al [7]	An end-to-end training method	Display the effectiveness of estimation and prediction	The prediction efficiency reaches 85%
	Zhou et al [8]	A unified motion correction and denoising resistance network	Produce high-quality motion compensated low noise reconstruction	The accuracy rate can reach 90%

	Wang et al [9]	A real-time dynamic vision system	Find complete dynamic regions	The efficiency reaches 80%
	Asad et al [10]	A spatial and temporal feature learning method	Combine the multi-level features	Accuracy can reach 85%
Deep learning	Shahbazi et al [11]	A recursive neural network-based LSTM	Be in line with the basic tracking performance	Computing cost is reduced by 10%
	Iraei et al [12]	A target tracking algorithm based on CNN and PF	Track objects through PF	The tracking accuracy is improved by 10%
	Liu et al [13]	A fault detection method	Reduce sudden noise interference	The detection efficiency can reach 90%
	Chen et al [14]	A video-based action recognition network	Learn each view action feature	Precision is up to 91%
	Liu et al [15]	A dynamic spatiotemporal network	The final significant plot is obtained	The accuracy rate can reach 90%

In summary, video stabilization technology based on deep learning has made significant progress. However, there are still challenges in dealing with complex scenes and multivariate problems. In addition, most existing methods need a large amount of pre training data and computational resources. For some real-world application scenarios, real-time performance and efficiency remain key challenges. Therefore, this study proposes an optimization method for video stabilization technology that integrates the Tiny-Res-PWNet model. This is to achieve higher quality, more stable, and more efficient image stabilization technology, and to perform better in different application scenarios.

3 Design of optimization method for video stabilization technology integrating Tiny-Res-PWNet model

This chapter proposes the design of optimization methods for video stabilization technology, including improvements and optimization methods for the PWSNet model. Faster network model running speed and better image stabilization effect are achieved with less resource consumption. This can be achieved by replacing the encoder convolutional layer with the residual module, extracting feature information from different network layers through feature fusion, Batch Normalization (BN) layer, bottleneck residual module, etc. Meanwhile, the PWSNet model is lightweight processed to obtain a smaller Tiny-Res-PWNet image stabilization network model.

3.1 Design of optimization method for video stabilization effect

With the continuous development of mobile devices and camera technology, users have an increasing demand for video stabilization effects [16]. To this end, research is being conducted to improve the output structure of the image stabilization network and add Fourier spectrum constraints and local motion constraints. Meanwhile, the affine matrix parameters are optimized to make the transformation parameters output by the network more similar, thereby reducing the difficulty of learning the overall jitter pattern of the image. PWSNet is a deep learning network used for image processing and computer vision tasks. The purpose is for enhancing the performance of image processing tasks by learning pixel level stability [17]. Pixel warping field is the motion model of PWSNet, used to directly map the relationships between all pixels between stable and unstable frames [18]. In the pixel warping, the relationship between the warping matrix and the corresponding pixels is shown in equation (1).

$$\begin{cases} T_x(i,j) = x_0 \\ T_y(i,j) = y_0 \end{cases} \tag{1}$$

In equation (1), the two warping matrices are T_x and T_y, with pixel coordinates (x_0, y_0) and pixel points (i, j). The warp matrix records the horizontal and vertical coordinates of the source pixel. The pixel values in stable frame $\hat{I}$ come from unstable frame I. The pixel warping field is shown in Figure 1.

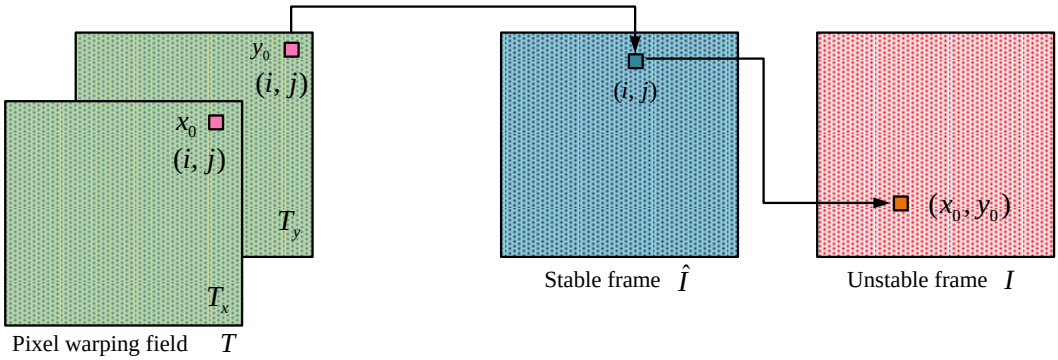

Figure 1: Pixel warping field

The single-stage structure of PWSNet is similar to the encoder decoder framework of U-net architecture [19]. The encoder consists of multiple convolutional layers, gradually downsampling to generate smaller feature maps and increasing the quantity of channels to enhance learning ability. The decoder structure is similar to the encoder, generating larger feature maps through convolutional layer upsampling and reducing the number of channels. The convolutional layers of the decoder and encoder are connected to each other through skip connections. At each stage in the figure, PWSNet generates two equally sized pixel warping fields to record the relationship between the source pixel and the target pixel and to achieve image stability. During the generation of pixel warping field by PWSNet, the horizontal and vertical movement positions of pixels are shown in equation (2).

$$\begin{cases} T_0^x(i,j) = \sum_{n=1,2,3} H_t(1,n) \times A(i,j,n) \\ T_0^y(i,j) = \sum_{n=1,2,3} H_t(2,n) \times A(i,j,n) \end{cases} \quad (2)$$

In equation (2), the pixel's horizontal and vertical movement positions are T_0^x and T_0^y, respectively, the affine transformation matrix is H_t, and the constant matrix is A. The overall architecture of PWSNet training is based on a dual branch neural network with shared identical parameters. This architecture can ensure the temporal and spatial consistency of continuous stable frames. In the network testing phase, there is no need to constrain the network. Only one branch is used to generate stable videos, reducing the consumption of computing resources. The overall framework of PWSNet training is shown in Figure 2.

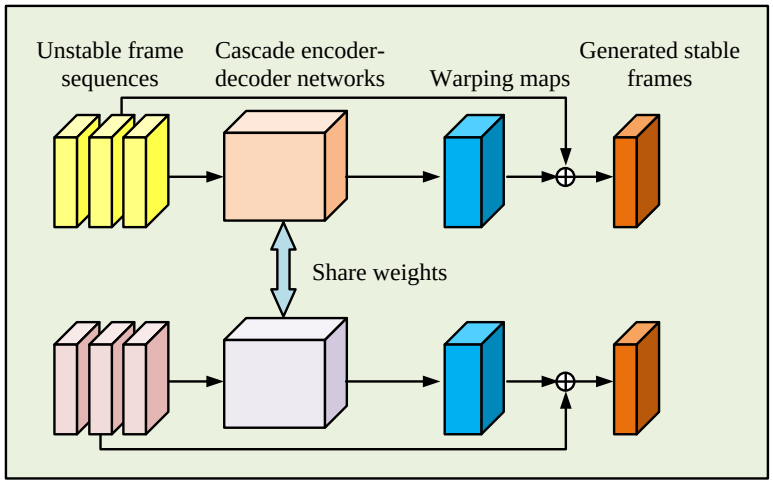

Figure 2: Overall framework for PWSNet training

In the frequency domain, using Fourier spectral constraints can be used as a method to enhance the spatial smoothness of the warping field. The frequency domain loss function is calculated as shown in equation (3).

$$L_{frequency} = \left\| \hat{G} \cdot F(W_x) \right\|_2 + \left\| \hat{G} \cdot F(W_y) \right\|_2 \quad (3)$$

In equation (3), the frequency domain loss is $L_{frequency}$, the weighted filtering function is $\hat{G}$, and the two-dimensional Fourier transform is F. The pixel warping matrices in the x and y directions are W_x and W_y, respectively. The frequency spectrum after Fourier transform of the warping matrix is $F(W)$. The

weighted filtering function is calculated as shown in equation (4).

$$\hat{G} = \frac{\max(G) - G}{\max(G)} \tag{4}$$

In equation (4), the two-dimensional Gaussian distribution function that satisfies mean 0 and variance 10 is G. The global affine transformation calculation is shown in equation (5).

$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & b_1 \\ m_{21} & m_{22} & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{5}$$

In equation (5), the coordinates of the grid vertices before the affine transformation are $[X, Y]$. The coordinate of the network vertices after the affine transformation is $[X', Y']$. The global affine transformation parameters are m_{11}, m_{12}, m_{21}, m_{22}, b_1, and b_2. The global vector is defined as equation (6).

$$M_d = \begin{bmatrix} X' \\ Y' \end{bmatrix} - \begin{bmatrix} X \\ Y \end{bmatrix} \tag{6}$$

In equation (6), the global vector is M_d. The intensity of local motion is measured by the L2 norm of the sparse motion field. The calculation of local motion loss is shown in equation (7).

$$L_{motion} = \sum_{n=1}^{N} \left\| M_{n,g} - M_d \right\|_2 \tag{7}$$

In equation (7), the local motion loss is L_{motion}. The quantityr of grid vertices in a frame is N. The number of grid vertices serves as n. The vector of the n-th grid vertex is $M_{n,g}$. The affine transformation matrix is shown in equation (8).

$$H_t = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \end{bmatrix} \tag{8}$$

In equation (8), $m_0 - m_5$ represent the scaling and rotation relationships before and after the transformation. To overcome systematic errors in affine transformations, an improved method is proposed. This method only retains the scaling, rotation, and translation transformations to reduce the coupling between the four parameters representing rotation and scaling. The improved network encoder generation structure is shown in Figure 3.

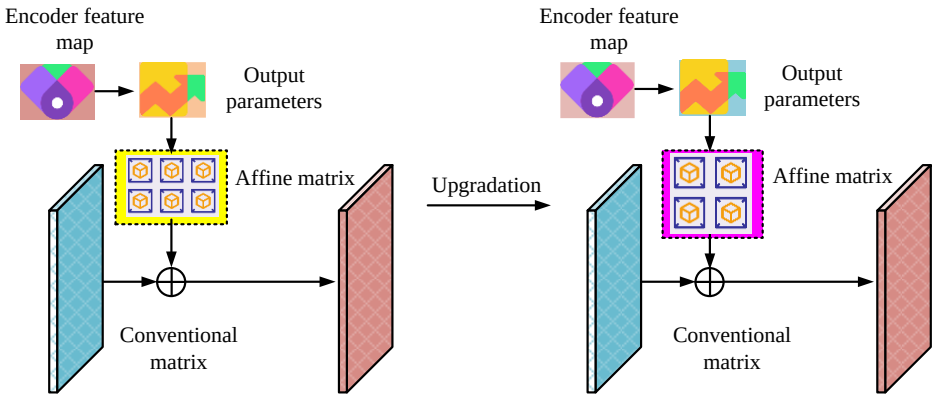

Figure 3: Improved network encoder generation structure

The improved network encoder generation structure first obtains transformation parameters through a 1*1 convolution operation on the input feature map. Then, the transformation parameters are filled into the affine transformation matrix to generate a feature map with a shape of 1*1*512. This feature map facilitates the subsequent calculation of generated parameters. The entire process is equivalent to learning a weight of 1*1*512, similar to the operation method of fully connected layers. The improved affine transformation calculation is shown in equation (9).

$$H_t = \begin{bmatrix} S\cos\theta & -S\sin\theta & \Delta x \\ S\sin\theta & S\cos\theta & \Delta y \end{bmatrix} \tag{9}$$

In equation (9), the scaling factor is S, the rotation angle is θ, and the displacements in the x and y directions are Δx and Δy.

3.2 Design of Tiny-Res-PWNet model

A lightweight image stabilization network Tiny-Res-PWNet is proposed to solve the problems of complex network structure, high computational complexity, and low output frame rate in PWSNet. Tiny-Res-PWNet replaces the encoder convolutional layer with a residual module to improve the training convergence and fitting ability of the network. By using feature fusion to extract feature information from different network layers, neural networks can achieve

faster model running speed and better image stability with less resource consumption [20]. In Tiny-Res-PWNet, the residual module is used to replace the encoder convolutional layer. The residual module introduces short-circuit connections, allowing the network to directly learn the residual mapping between input and output features. The residual module helps to improve the training convergence speed of the network and makes it easier for the network to fit complex nonlinear relationships. Tiny-Res-PWNet extracts feature information from different network layers through feature fusion, combining feature maps from different levels to obtain richer and more comprehensive feature representations. In Tiny-Res-PWNet, feature fusion can be achieved through skip connections. Jumping connections fuse the feature maps of the encoder with those of the decoder, allowing the network to utilize more information for prediction. In addition, the PWSNet model is lightweight processed to obtain a faster running speed, higher output frame rate, and smaller model structure of the Tiny-Res-PWNet image stabilization network model. The output feature map of the residual network is showcased in equation (10).

$$H(x) = x_l + F(x_l) \tag{10}$$

In equation (10), the input feature map and output feature map of layer F are G and E, respectively. The disadvantage of traditional residual networks is that as the network depth increases. The computational load increases and the effectiveness gradually weakens [21]. The bottleneck residual module can perform convolution operations on relatively low dimensions by reducing the number of channels and using a 1*1 convolution kernel, thereby improving computational efficiency and effectiveness. The bottleneck residual module consists of a 1*1 convolutional layer, a 3*3 convolutional layer, and a 1*1 convolutional layer, used to extract features and solve the gradient vanishing problem in deep networks. To optimize the training structure, this study introduces BN layers to normalize the data distribution. This causes the input value of the activation function to fall in areas with larger gradients, thereby avoiding gradient vanishing and reducing training time. The BN layer is usually used after the convolutional layer. In addition, this study has made improvements to the encoder structure to enhance the network feature extraction capability and improve operational efficiency. It replaces the ordinary 3*3 convolutional layer with a bottleneck structure combination layer, and connects the input and output together through skip connections to form a bottleneck residual structure. The optimized single-layer structure of the encoder is shown in Figure 4.

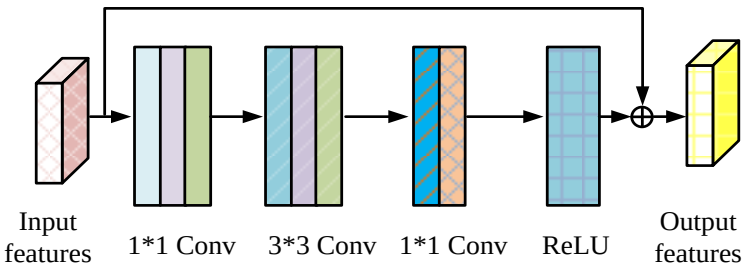

Figure 4: Optimized single-layer structure of encoder

To learn multi-scale and multi-dimensional features in images, the bottleneck residual block of the encoder structure reduces the output feature size by half layer by layer. The output feature channel first increases and then remains unchanged. Under the condition that channel transformation is required, an additional 1*1 convolutional layer is introduced for channel transformation. This study applies depthwise separable convolution to the input layer, warping field output layer, encoder bottleneck residual module intermediate layer, and decoder transpose convolution layer of the network. In addition, the computational load within the network is reduced by reducing the network hierarchy. The structure of the lightweight Tiny-Res-PWNet model is shown in Figure 5.

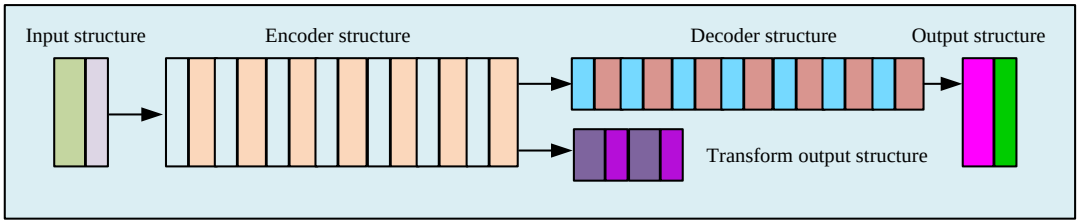

Figure 5: Tiny-Res-PWNet model structure

Tiny-Res-PWNet uses three different activation functions: ReLU, Leaky ReLU, and Tanh. Most of the convolutional layers are activated using ReLU, the output layer is activated using Tanh. The convolutional layers connected to the output layer are activated using Leaky ReLU. The ReLU activation function is shown in equation (11).

$$\begin{cases} g_1(x) = \max(0, x) \\ g_1^{'}(x) = \begin{cases} 1, x > 0 \\ 0, x \le 0 \end{cases} \end{cases} \tag{11}$$

In equation (11), the ReLU activation function is $g_1(x)$, and its derivative is $g_1^{'}(x)$. The Leaky ReLU activation function is shown in equation (12).

$$\begin{cases} g_2(x) = \max(ax, x) \\ g_2^{'}(x) = \begin{cases} 1, x > 0 \\ a, x \le 0 \end{cases} \end{cases} \tag{12}$$

In equation (12), the Leaky ReLU activation function is $g_2(x)$, its derivative is $g_2^{'}(x)$, and the constant term is a. The Tanh activation function is shown in equation (13).

$$\begin{cases} g_3(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}} \\ g_3^{'}(x) = 1 - \left(\dfrac{e^x - e^{-x}}{e^x + e^{-x}} \right)^2 = 1 - \left(g_3(x) \right)^2 \end{cases} \tag{13}$$

In equation (13), the Tanh activation function is $g_3(x)$, and its derivative is $g_3^{'}(x)$. The content loss function is shown in equation (14).

$$L_{MSE} = \frac{\sum_{i=1, j=1}^{W,H} (\hat{I}(i, j) - \tilde{I}(i, j))^2}{WH} \tag{14}$$

In equation (14), the content loss function is L_{MSE}, the true stable frame is $\tilde{I}$, and the pixel width and height of the frame are W and H, respectively. The feature loss function is shown in equation (15).

$$L_{fea} = \frac{\sum_{i=1}^{n_f} \left\| \tilde{P}_i - \omega(P_i) \right\|_2}{n_f} \tag{15}$$

In equation (15), the feature loss function is L_{fea}, the number of matched feature points is n_f, and the coordinate of a feature point in the real stable frame is $\tilde{P}$. The coordinate of a feature point in an unstable frame is P_i, and the coordinate of the feature point transformed by the warping field is $\omega(P_i)$.

4 Video stabilization technology optimization method integrating Tiny-Res-PWNet model

This chapter mainly analyzed the application of optimization methods for video stabilization technology that integrated the Tiny-Res-PWNet model. By setting the experimental environment and adjusting the training parameters, the performance of different image stabilization networks was compared. The improved PWSNet model was validated in terms of video structure similarity, fidelity, and stability.

4.1 Application analysis of optimization methods for video stabilization effect

The study used two different experimental environments, namely high performance and moderate performance. In a high-performance environment, Intel i9-12900k CPU, Nvidia RTX-3090 GPU, 32G graphics memory, 64G memory, Python 3.9 programming language, and Python 1.7 deep learning framework were used. In a medium performance environment, Intel i3-10100f CPU, Nvidia GTX-1650S GPU, 8GB of graphics memory, and 16GB of memory were used. The size of the experimental image is uniformly 640*360, and the pixel warping field size is 256*256. The weight initialization adopted a normal distribution, and the training used an Adam optimizer with a batch size of 16 and an initial learning rate of 0.001. The experimental dataset adopts GoPro, which includes 3214 blurred images with a size of 1280×720, of which 2103 are training images and 1111 are test images. The GoPro dataset consists of one-to-one corresponding real blurred images and ground truth images, both captured by high-speed cameras. For verifying the performance of the improved PWSNet model, the study compared the Optical Flow model, Block Matching model, and the pre-improved PWSNet model. The structural similarity evaluation results of different image stabilization networks are shown in Figure 6.

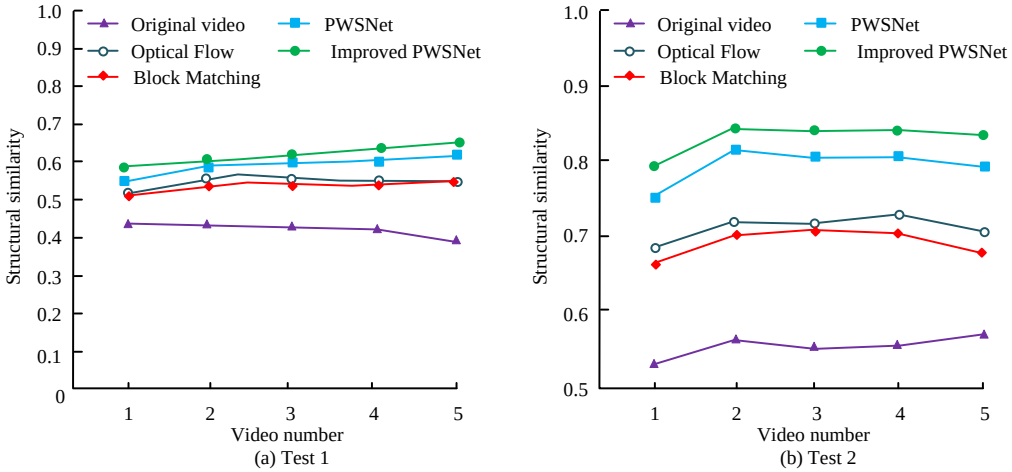

Figure 6: Structural similarity evaluation results of different image stabilization networks

In Figure 6 (a), the original video structure similarity is less than 0.5, and the improved PWSNet model has improved the video similarity evaluation index by about 41.8%. Compared to the Optical Flow model, Block Matching model, and PWSNet, the similarity evaluation index of the improved PWSNet model has increased by about 7.3%, 7.9%, and 2.7%, respectively. In Figure 6 (b), the similarity of the original video structure is greater than 0.5. The improved PWSNet model has increased the video similarity evaluation index by approximately 52.8%. Compared to the Optical Flow model, Block Matching model, and the pre-improved PWSNet model, the similarity evaluation index of the improved PWSNet model has increased by approximately 14.1%, 17.3%, and 4.3%, respectively. The outcomes showcase that the improved PWSNet model possesses excellent accuracy and reliability in evaluating video structural similarity, which can better capture the structural similarity between videos. The fidelity and stability evaluation results of different image stabilization networks are shown in Figure 7.

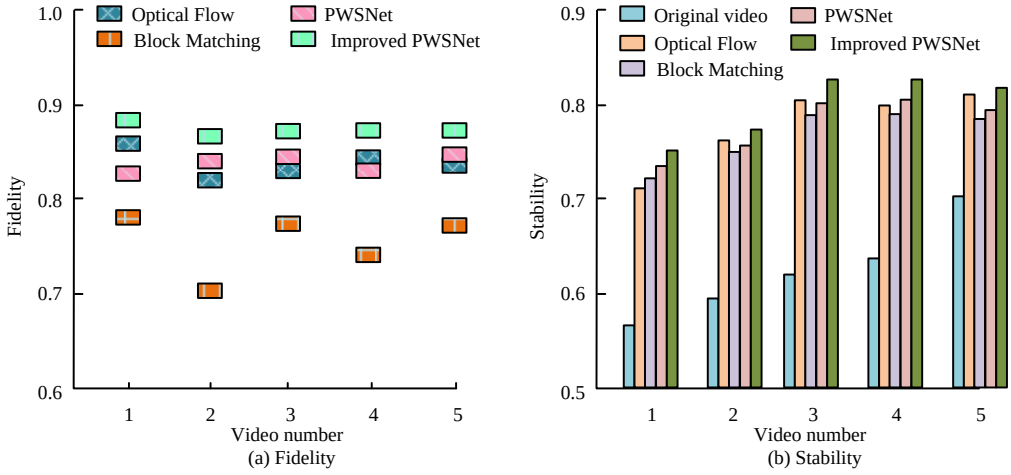

Figure 7: Evaluation results of fidelity and stability for different image stabilization networks

Figure 7 (a) shows the fidelity evaluation results, and the average fidelity of the improved PWSNet model is 0.87. Compared to the Optical Flow model, Block Matching model, and the pre-improved PWSNet model, the improved PWSNet model has improved its fidelity evaluation metrics by approximately 2.1%, 9.9%, and 1.9%, respectively. Figure 7 (b) shows the stability evaluation results, and the average stability value of the improved PWSNet model is 0.78. Compared to the pre-improved PWSNet model, the stability evaluation index of the improved PWSNet model has increased by about 3.7%. The outcomes showed that the improved PWSNet model achieved essential improvements in both fidelity and stability. This indicates that the model can better maintain image quality and stability in video processing, providing a more reliable solution for video processing tasks. The stabilization results of different types of low-quality videos are shown in Figure 8.

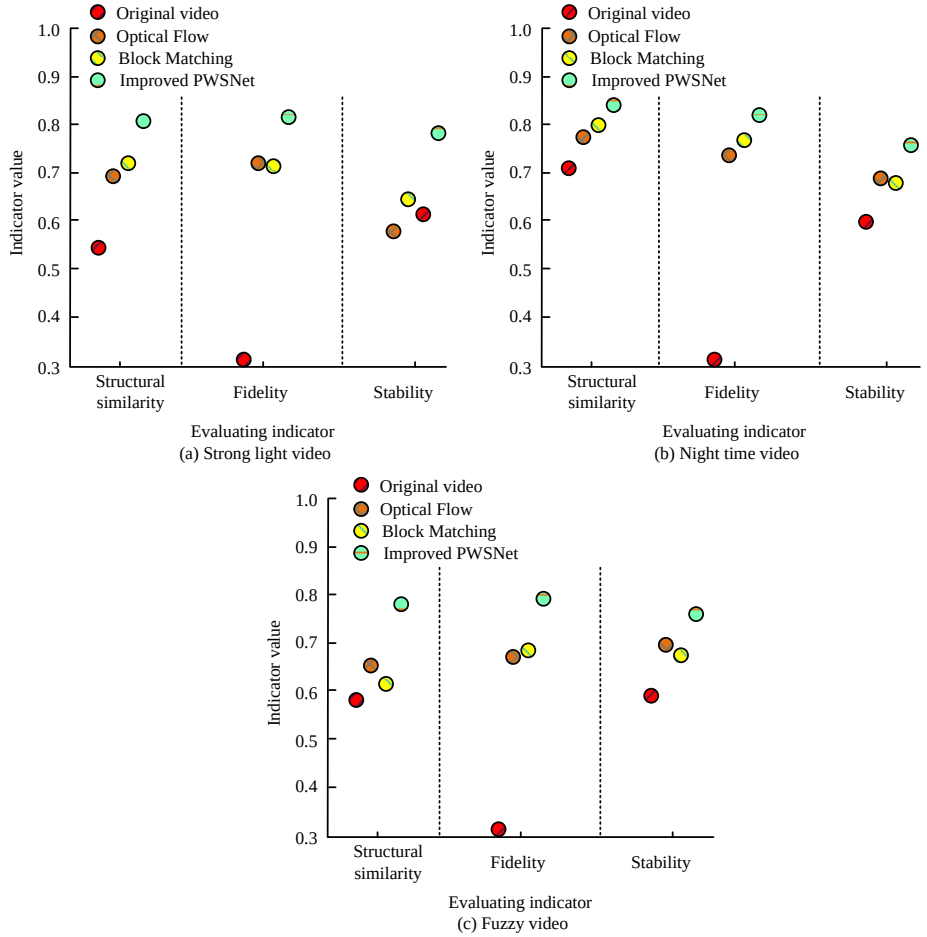

Figure 8: Stable image processing results of low-quality videos of different types

Figure 8 (a) shows the comparison of image stabilization results for strong light videos. The improved PWSNet model has evaluation indicators for structural similarity, fidelity, and stability of 0.82, 0.84, and 0.79, respectively. Compared with the Optical Flow model and Block Matching model, the improved PWSNet model has improved the evaluation indicators of structural similarity, fidelity, and stability by about 16.9%, 15.1%, and 16.5%. Figure 8 (b) showcases the comparison of image stabilization results for nighttime videos. The evaluation indicators for structural similarity, fidelity, and stability of the improved PWSNet model are 0.83, 0.82, and 0.77, respectively. Relative to the Optical Flow model, the improved PWSNet model possesses improved the evaluation indicators of structural similarity, fidelity, and stability by about 4.2%, 7.8%, and 13.1%. Figure 8 (c) showcases the comparison of image stabilization results for blurred videos. The evaluation indicators for structural similarity, fidelity, and stability of the improved PWSNet model are 0.78, 0.80, and 0.76, respectively. Compared to the Optical Flow model and Block Matching model, the improved PWSNet model has improved the structural similarity, fidelity, and stability evaluation indicators by approximately 25.1%, 20.9%, and 13.2%. Based on the analysis of the above data, the improved PWSNet model has shown significant improvement in image stabilization processing results for different types of low-quality videos. Whether it is strong light videos, nighttime videos, or blurry videos, the PWSNet model can better maintain the structural similarity of videos and improve the fidelity and stability. The comparison of screenshots before and after video stabilization is shown in Figure 9. Figure 9 (a) shows the original video image. Figure 9 (b) shows the video image after image stabilization processing. After image stabilization processing, the clarity of the image has been improved. During the image stabilization process, noise reduction is applied to the image to make the video more pure.

(a) Original image (b) Stable image

Figure 9: Video stabilization image before and after the screenshot comparison

4.2 Application analysis of Tiny-Res-PWNet model

The Res-PWNet model introduced residual modules for structural improvement on the basis of the original PWSNet. The impact of residual modules was compared and analyzed in different types of video scenes. The comparison of Res-PWNet image stabilization performance in different types of video scenes is showcased in Table 2.

Table 2: Comparison of Res-PWNet image stabilization performance in different types of video scenes

Video type	Algorithm	Structural similarity	Fidelity	Stability
Routine	Optical Flow	0.802	0.852	0.813
	PWSNet	0.773	0.832	0.819
	Res-PWNet	0.814	0.845	0.842
High speed	Optical Flow	0.792	0.782	0.806
	PWSNet	0.761	0.771	0.791
	Res-PWNet	0.806	0.811	0.824
High density	Optical Flow	0.752	0.782	0.705
	PWSNet	0.746	0.809	0.748
	Res-PWNet	0.761	0.817	0.726
High light intensity	Optical Flow	0.742	0.801	0.731
	PWSNet	0.732	0.801	0.725
	Res-PWNet	0.756	0.809	0.765

In Table 2, Res-PWNet exhibits excellent image stabilization performance in different types of video scenes, with high structural similarity, fidelity, and stability. Res-PWNet has better image stabilization performance compared to PWSNet in different types of video scenes. In conventional scenarios, the structural similarity score of Res-PWNet is 0.814, which is higher than the score of PWSNet by 0.773. Meanwhile, the fidelity score of Res-PWNet is 0.845, which is higher than the score of PWSNet by 0.832. In both high-speed and high-density scenarios, Res-PWNet has higher scores for structural similarity and fidelity than PWSNet. In high light intensity scenes, the structural similarity and fidelity scores of Res-PWNet are slightly higher than those of PWSNet. By introducing residual modules, Res-PWNet can better capture motion information in videos and make more accurate predictions and compensations. The comparison of network module complexity after the introduction of residual module is shown in Figure 10.

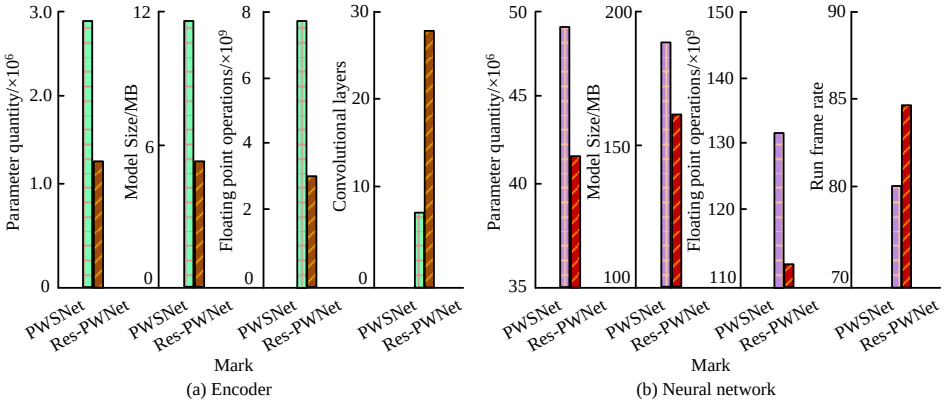

Figure 10: Comparison of network module complexity after introducing residual modules

Figure 10 (a) shows the complexity comparison of the encoder module. Compared to PWSNet encoder, Res-PWNet encoder reduces the number of parameters by 53.7%, model size by 54.7%, floating-point operation by 66.9%, and convolutional layer depth by three times. Figure 10 (b) showcases a comparison of the overall complexity of the network. Compared to PWSNet encoder, Res-PWNet encoder reduces parameter count by 12.1%, model size by 11.7%, floating-point operation by 13.2%, and running frame rate has increased from 80.1 to 84.6, an increase of 5.6%. The outcomes show that the introduction of residual modules could markedly decrease the parameters and model size of the network and also reduce the floating-point computational complexity. This improves the performance and stability of the network. In addition, introducing residual modules can significantly decrease the overall complexity of the network and improve the running frame rate, thereby improving the real-time performance and stability of the network. Some videos from the aviation video dataset were used for testing to test the image stabilization performance of Tiny-Res-PWNet in airborne video stabilization application scenarios. The stability performance test results of Tiny-Res-PWNet are shown in Figure 11.

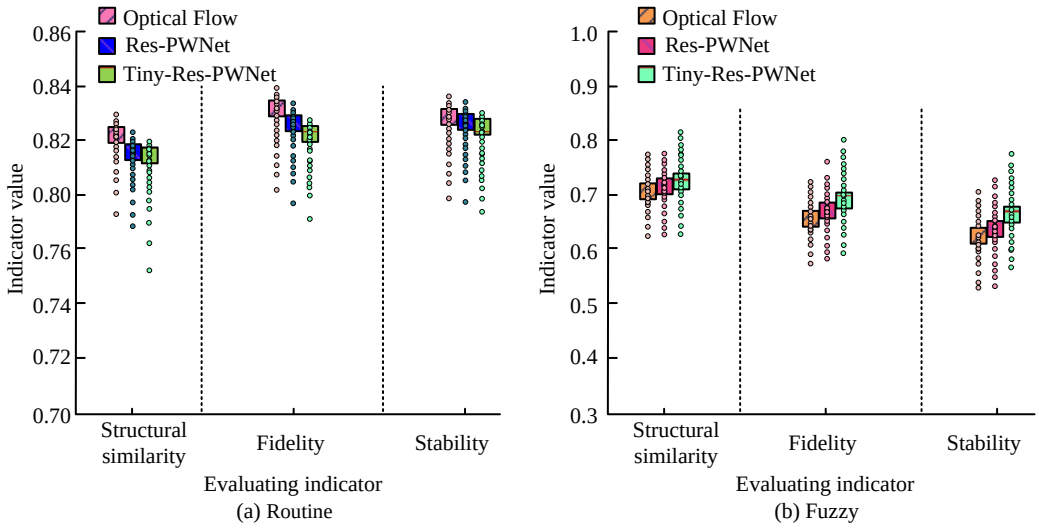

Figure 11: Tiny-Res-PWNet image stabilization performance test results

Figure 11 (a) showcases the image stabilization results of conventional clear videos. Figure 11 (b) showcases the image stabilization results of very large-scale blurred videos. The traditional optical flow model for image stabilization has better performance, but its processing effect for complex scene videos is poor. Tiny-Res-PWNet performs better in handling complex scenes and has high robustness. Compared with Res-PWNet, Tiny-Res-PWNet can better handle various complex scenes while maintaining image stability performance. The comparison results of the running speed of the Tiny-Res-PWNet model are showcased in Table 3.

Table 3: Comparison results of running speed of Tiny-Res-PWNet model

Project	PWSNet	Res-PWNet	Tiny-Res-PWNet
Parameter quantity	49.2	42.8	6.9
Model size/×106	186.5	164.2	27.5
Floating point operations/MB	129.4	114.9	12.3
High performance running frame rate/×109	78.3	83.1	131.2
Low performance running frame rate	3.8	3.9	7.9

In Table 3, the Tiny-Res-PWNet model is significantly smaller in terms of parameter count, model size, and floating-point operations. There is an excellent performance in high-performance frame rate, reaching 131.2, far higher than the 78.3 and 83.1 of Res-PWNet and PWSNet models. In terms of low performance frame rates, Tiny-Res-PWNet also performs well, reaching 7.9, higher than Res-PWNet and PWSNet models at 3.8 and 3.9. Therefore, the Tiny-Res-PWNet model has a fast-running speed and a small model size, making it suitable for use in resource limited environments.

5 Discussion

The study proposed a video stability optimization method that integrates the Tiny-Res-PWNet model. Compared to existing advanced methods, the proposed improved PWSNet model achieved significant improvements in video structure similarity, realism, and stability. Especially when dealing with different types of low-quality videos, such as strong light, nighttime, and blurry videos, the improved PWSNet model performed particularly well. These achievements were attributed to the introduction of residual modules, which enabled the network to better capture motion information in videos while maintaining model simplicity, achieving more accurate prediction and compensation. In addition, the Tiny-Res-PWNet model had lower complexity in terms of parameter count, model size, and floating-point operations, which made the model have higher performance and stability in practical applications. Compared with Res-PWNet and PWSNet models, Tiny-Res-PWNet had higher robustness and faster running speed when dealing with complex scenes. These advantages enabled the proposed method to achieve good results in aviation video stabilization application scenarios. Through comprehensive comparison, the proposed video stability optimization method outperformed existing methods in multiple aspects. These advantages mainly stem from the introduction of residual modules and optimization of network structures, which enable the model to maintain high performance while possessing stronger robustness and generalization ability. In addition, the Tiny-Res-PWNet model is particularly suitable for resource constrained environments by reducing model size and computational complexity.

6 Conclusion

To improve the performance and stability of video stabilization technology, the PWSNet model was optimized. A new network model, Tiny-Res-PWNet, was proposed. In the Tiny-Res-PWNet model, ResNet was first used as the basic network structure for strengthening the depth and expressive power of the network. Then, based on ResNet, the pixel level weight mechanism of PWSNet was introduced to improve the effectiveness of video stabilization processing. In addition, techniques such as batch normalization and residual connections were also used to accelerate network training and improve network convergence to further enhance the performance and stability of the Tiny-Res-PWNet model. The results showed that the improved PWSNet model improved the video similarity evaluation index by approximately 41.8%. Compared to the Optical Flow model, Block Matching model, and the pre-improved PWSNet model, the similarity evaluation index of the improved PWSNet model increased by about 7.3%, 7.9%, and 2.7%, respectively. The improved PWSNet model enhanced video stabilization performance. The Tiny-Res-PWNet model reduced computational complexity while maintaining high-performance frame rates. This study provides a new method for video stabilization technology, which can be applied in various practical scenarios and has great potential. The limitations of this study mainly lie in the limited hardware equipment and software tools in the experimental environment. The limitations may have a certain impact on the research results. Future research can consider exploring more types of network structures and optimization algorithms to broaden the applicability of research methods, improve their reliability and effectiveness. At the same time, in-depth research is conducted on the optimization and improvement of hardware devices and software tools combined with practical application scenarios to reduce their limitations on research results.

References

[1] K. A. Mills, and A. Brown, “Immersive virtual reality (VR) for digital media making: transmediation is key,” Learning, Media and Technology, vol. 47, no. 2, pp. 179-200, 2022. https://doi.org/10.1080/17439884.2021.1952428

[2] X. Jin, F. Jiang, L. Li, and T. Zhong, "Plenoptic 2.0 intra coding using imaging principle," IEEE Transactions on Broadcasting, vol. 68, no. 1, pp. 110-122, 2022. https://doi.org/10.1109/TBC.2021.3108058

[3] G. Du, K. Wang, S. Lian, and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review," Artificial Intelligence Review, vol. 54, no. 3, pp. 1677-1734, 2021. https://doi.org/10.1007/s10462-020-09888-5

[4] S. Shimada, V. Golyanik, W. Xu, P. Pérez, and C. Theobalt, "Neural monocular 3d human motion capture with physical awareness," ACM Transactions on Graphics (ToG), vol. 40, no. 4, pp. 1-15, 2021. https://doi.org/10.1145/3450626.3459825

[5] M. Poggi, F. Tosi, K. Batsos, P. Mordohai, and S. Mattoccia, "On the synergies between machine learning and binocular stereo for depth estimation from images: a survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 9, pp. 5314-5334, 2021. https://doi.org/10.1109/tpami.2021.3070917

[6] H. Son, J. Lee, J. Lee, S. Cho, and S. Lee, "Recurrent video deblurring with blur-invariant motion estimation and pixel volumes," ACM Transactions on Graphics (TOG), vol. 40, no. 5, pp. 1-18, 2021. https://doi.org/10.1145/3453720

[7] C. Chen, C. X. Lu, B. Wang, N. Trigoni, and A. Markham, "DynaNet: Neural Kalman dynamical model for motion estimation and prediction," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 12, pp. 5479-5491, 2021. https://doi.org/10.1109/TNNLS.2021.3112460

[8] B. Zhou, Y. J. Tsai, X. Chen, J. S. Duncan, and C. Liu, "MDPET: a unified motion correction and denoising adversarial network for low-dose gated PET," IEEE Transactions on Medical Imaging, vol. 40, no. 11, pp. 3154-3164, 2021. https://doi.org/10.1109/TMI.2021.3076191

[9] K. Wang, X. Yao, N. Ma, and X. Jing, "Real-time motion removal based on point correlations for RGB-D SLAM in indoor dynamic environments," Neural Computing and Applications, vol. 35, no. 12, pp. 8707-8722, 2023. https://doi.org/10.1007/s00521-022-07879-x

[10] M. Asad, J. Yang, J. He, P. Shamsolmoali, and X. He, "Multi-frame feature-fusion-based model for violence detection," The Visual Computer, vol. 37, no. 1, pp. 1415-1431, 2021. https://doi.org/10.1007/s00371-020-01878-6

[11] M. Shahbazi, M. H. Bayat, and B. Tarvirdizadeh, "A motion model based on recurrent neural networks for visual object tracking," Image and Vision Computing, vol. 126, no. 1, pp. 104533-104544, 2022. https://doi.org/10.1016/j.imavis.2022.104533

[12] I. Iraei, and K. Faez, "A motion parameters estimating method based on deep learning for visual blurred object tracking," IET Image Processing, vol. 15, no. 10, pp. 2213-2226, 2021. https://doi.org/10.1049/ipr2.12189

[13] B. Liu, Y. Chai, Y. Liu, C. Huang, Y. Wang, and Q. Tang, "Industrial process fault detection based on deep highly-sensitive feature capture," Journal of Process Control, vol. 102, no. 1, pp. 54-65, 2021. https://doi.org/10.1016/j.jprocont.2021.04.003

[14] B. Chen, H. Tang, Z. Zhang, G. Tong, and B. Li, "Video-based action recognition using spurious-3D residual attention networks," IET Image Processing, vol. 16, no. 11, pp. 3097-3111, 2022. https://doi.org/10.1049/ipr2.12541

[15] J. Liu, J. Wang, W. Wang, and Y. Su, "DS-Net: Dynamic spatiotemporal network for video salient object detection," Digital Signal Processing, vol. 130, no. 1, pp. 103700-103711, 2022. https://doi.org/10.1016/j.dsp.2022.103700

[16] C. Z. Dong, and F. N. Catbas, "A review of computer vision–based structural health monitoring at local and global levels," Structural Health Monitoring, vol. 20, no. 2, pp. 692-743, 2021. https://doi.org/10.1177/1475921720935585

[17] I. Salman, and J. Vomlel, "Learning the structure of Bayesian networks from incomplete data using a mixture model," Informatica, vol. 47, no. 1, pp. 83-96, 2023. https://doi.org/10.31449/inf.v47i1.4497

[18] M. Majd, and R. Safabakhsh, "A motion-aware ConvLSTM network for action recognition," Applied Intelligence, vol. 49, no. 7, pp. 2515-2521, 2019. https://doi.org/10.1007/s10489-018-1395-8

[19] C. Hong, "Basketball video image segmentation using neutrosophicFuzzy C-means clustering algorithm," Informatica, vol. 48, no. 9, pp. 145-154, 2024. https://doi.org/10.31449/inf.v48i9.5929

[20] S. Choudhuri, S. Adeniye, and A. Sen, "Distribution alignment using complement entropy objective and adaptive consensus-based label refinement for partial domain adaptation," Artificial Intelligence and Applications, vol. 1, no. 1, pp. 43-51, 2023. https://doi.org/10.47852/bonviewAIA2202524

[21] J. Purohit, and R. Dave, "Leveraging deep learning techniques to obtain efficacious segmentation results," Archives of Advanced Engineering Science, vol. 1, no. 1, pp. 11-26, 2023. https://doi.org/10.47852/bonviewAAES32021220