IJWR

VG-CGARN: Video Generation Using Convolutional Generative Adversarial and Recurrent Networks

Fatemeh Sobhani Manesh[a*], Amin Nazari[b], Muharram Mansoorizadeh[c], MirHossein Dezfoulian[d]
[a] MSc, Computer Engineering Department, Bu-Ali Sina University, Hamedan, Iran; f.sobhani@alumni.basu.ac.ir
[b] Ph.D. Candidate, Computer Engineering Department, Bu-Ali Sina University, Hamedan, Iran; aminnazari91@gmail.com
[c] Associate Professor, Computer Engineering Department, Bu-Ali Sina University, Hamedan, Iran; mansoorm@basu.ac.ir
[d] Assistant Professor, Computer Engineering Department, Bu-Ali Sina University, Hamedan, Iran; dezfoulian@basu.ac.ir

ABSTRACT

Generating dynamic videos from static images and accurately modeling object motion within scenes are fundamental challenges in computer vision, with broad applications in video enhancement, photo animation, and visual scene understanding. This paper proposes a novel hybrid framework that combines convolutional neural networks (CNNs), recurrent neural networks (RNNs) with long short-term memory (LSTM) units, and generative adversarial networks (GANs) to synthesize temporally consistent and spatially realistic video sequences from still images. The architecture incorporates splicing techniques, the Lucas-Kanade motion estimation algorithm, and a loop feedback mechanism to address key limitations of existing approaches, including motion instability, temporal noise, and degraded video quality over time. CNNs extract spatial features, LSTMs model temporal dynamics, and GANs enhance visual realism through adversarial training. Experimental results on the KTH dataset, comprising 600 videos of fundamental human actions, demonstrate that the proposed method achieves substantial improvements over baseline models, reaching a peak PSNR of 35.8 and SSIM of 0.96—representing a 20% performance gain. The model successfully generates high-quality, 10-second videos at a resolution of 720×1280 pixels with significantly reduced noise, confirming the effectiveness of the integrated splicing and feedback strategy for stable and coherent video generation.

Keywords: ***Video Generation, Convolutional Neural Networks, Recurrent Neural Networks, Generative Adversarial Networks.***

1. Introduction

Video is one of the most powerful forms of communication, capable of receiving, recording, processing, transmitting, storing, and reconstructing moving images. It conveys a wealth of information by seamlessly integrating multiple media elements such as sound, images, and text [1]. Given that humans are naturally skilled at processing and interpreting visual information and most of their sensory input is received through the visual system, video has become the most effective medium. By engaging multiple senses simultaneously, video allows for faster comprehension, stronger communication, and makes it an essential tool in modern communication [2].

Computer Vision (CV) attempts to simulate human visual perception, enabling machines to observe, identify, and analyze objects and their relationships in the environment [3]. The ability to perceive a scene and its dynamics is one of the human-level functions. Although the human visual system is not parallel, it has high speed, accuracy, and quality that allows it to recognize and understand complex scenes. Simulating this visual system in computers remains a challenging endeavor. Deep learning (DL) algorithms, particularly convolutional neural networks (CNNs), recurrent neural networks (RNNs) and Generative Adversarial Networks (GANs) are employed to address this challenge [4].

http://dx.doi.org/10.22133/ijwr.2025.505096.1265
Citation F. Sobhani Manesh, A. Nazari, M. Mansoorizadeh and M. Dezfoulian, " VG-CGARN: Video Generation Using Convolutional Generative Adversarial and Recurrent Networks ", *International Journal of Web Research*, vol.8, no2.,pp.65-77, 2025, doi: http://dx.doi.org/10.22133/ijwr.2025.505096.1265

*Coressponding Author
Article History: Received: 7 January 2025; Revised: 20 March 2025; Accepted: 23 March 2025.

CNNs have transformed CV by automatically learning features in images, ranging from simple edges to complex patterns such as faces and vehicles. Their ability to manage high-dimensional data enhances the performance of CV systems, making them essential for tasks such as object detection, face recognition, and scene understanding [5]. RNNs are designed for sequential data, such as time series and natural language, and are also useful for CV tasks that involve temporal patterns in images or videos [6]. RNNs can store past inputs and use them to analyze dynamic visual data. Applications of RNNs in CV include detecting actions such as "running" or "cooking" in video clips, tracking moving objects, and creating captions for images. GANs are deep learning models comprising two competing neural networks: a generator that produces fake data (such as images) and a discriminator that differentiates between real and fake data. This adversarial process enhances the quality of the generated content [7]. GANs excel in creating realistic images, performing image-to-image translation, achieving super-resolution, and engaging in creative tasks by utilizing adversarial training to boost image realism and generate new data.

One of the important tasks in CV is video generation (VG) from images or animation of images. VG refers to the process of creating or combining video content from specific inputs such as images or videos. This field is very important due to its applications in the fields of artificial data generation for training machine learning models, future prediction, its applications in augmented reality and surveillance and security [2]. Research gaps in VG include various areas such as increasing temporal stability, motion realism, structural correlation, video quality and consistency [8]. A key challenge in CV is interpreting body movements and scene dynamics. Future image prediction and VG from still images remain particularly difficult tasks. These issues persist as significant obstacles due to the short duration and low quality of existing generated videos. Our focus in this paper is on improving video quality and correlation. For this, we employ a combination of CNN, RNN and GAN in proposed architecture.

A CNN extracts spatial features (edges, textures) to understand image structure within each frame. An RNN then models temporal dependencies between frames, ensuring motion continuity by leveraging internal memory to predict frame-to-frame changes. A GAN enhances realism by training a generator to produce frames that can fool a discriminator, overcoming the blurry or noisy outputs of traditional methods like Autoencoders. This CNN-RNN-GAN architecture addresses the multimodal nature of video generation, requiring both spatial and temporal processing, unlike existing methods that typically focus on only one. By integrating the strengths of each network, this hybrid approach significantly improves video quality and stability. Specifically, the RNN mitigates temporal instability by maintaining memory of previous frames, while the GAN reduces unnatural motion. CNN preserves image detail by extracting high-level features. Compared to GAN-alone approaches prone to temporal discontinuity or RNN-alone approaches resulting in blurred outputs, this combination provides both temporal coherence and realism. This integrated architecture offers a comprehensive solution for video generation, combining spatial understanding (CNN), temporal modeling (RNN), and realistic rendering (GAN), and applies to motion prediction, missing frame completion, and video resolution enhancement.

While recent advances in video generation (VG) have yielded significant progress, current models often suffer from limitations in either temporal consistency, spatial fidelity, or overall realism. Many approaches focus exclusively on one modality—either spatial (using CNNs) or temporal (using RNNs or Transformers)—which leads to unbalanced performance across key video quality dimensions. CNNs are effective at extracting spatial features from video frames, identifying crucial elements like edges and textures. RNNs model temporal dependencies, allowing them to understand frame sequences and maintain motion consistency. GANs, with their generative and discriminative networks, excel at creating high-quality, realistic video sequences. To summary, CNNs ensure high spatial quality, RNNs provide temporal coherence, and GANs enhance overall realism. This architecture enables end-to-end learning, allowing CNNs, RNNs, and GANs to collaboratively produce realistic, high-quality videos. The main contributions are listed below:

- Enhancing Video Quality and Consistency: We tackle VG challenges by improving video quality, stability, and correlation through a hybrid architecture that combines CNNs, RNNs, and GANs for more realistic video generation.
- Proposed Hybrid Video Generation Framework: Our architecture uses CNNs for spatial features, RNNs for temporal modeling, and GANs for high-quality sequences, ensuring spatial fidelity, temporal coherence, and effective end-to-end learning.
- Key Challenges in CV for Video Creation: The paper addresses vital CV issues like body movement interpretation, scene dynamics, and image-to-video forecasting, focusing on realism, quality, and consistency essential.

This section focused on video processing, analysis, and generation using various machine learning techniques. We addressed gaps in the existing literature and outline our motivation. The second section discusses relevant research, the third details our proposed method, and the fourth presents the results of our approach.

2. A Review of Research Literature

In this section, we first review related works. Then, we will have a brief overview of CNN, RNN, and GAN.

2.1. Related works

Generative adversarial networks (GANs) have been extensively employed in VG and related applications. Xiong et al. [9] proposed a novel approach titled "Generating Time-Lapse Videos Through Multi-Stage Dynamic Generative Adversarial Networks", achieving high-quality video synthesis with a resolution enhancement of up to 128× for 32 frames. The effectiveness of their model was validated through comprehensive quantitative and qualitative evaluations, demonstrating superiority over state-of-the-art methods. Vougioukas et al. [10] leveraged GANs for video-driven speech reconstruction, developing a model capable of producing intelligible and synchronized speech with lifelike quality. Their method was tested on the GRID dataset under both speaker-dependent and speaker-independent conditions, and evaluations included speech quality and word accuracy metrics.

Chen et al. [11] explored domain adaptation in videos using GANs, introducing VideoGAN to improve segmentation accuracy of colorectal polyps on multicenter datasets, achieving a 5% performance gain. They further validated their approach on the CamVid driving video dataset for a cloudy-to-sunny translation task, demonstrating its ability to significantly reduce domain gaps through extensive testing. Mira et al. [12] presented an end-to-end video-to-speech synthesis framework based on GANs, capable of generating high-quality speech even from small datasets like GRID. Notably, their model is the first to produce intelligible speech for Lip Reading in the Wild (LRW) dataset, featuring naturalistic recordings of diverse speakers. The proposed approach outperformed existing methods on the GRID and LRW datasets across multiple evaluation metrics.

Lan et al. [13] focused on unsupervised video summarization of wireless capsule endoscopy (WCE) videos using recurrent GANs. Their model, integrates a variational autoencoder-based LSTM architecture with pointer networks and dynamic memory techniques for summarization. The discriminator LSTM adversarial trains alongside the summarizer, enhancing the quality of the video summaries. Experiments conducted on the WCE-2019-Video dataset demonstrated that their model outperformed existing supervised and unsupervised video summarization techniques.

Singh et al. [14] apply Deep Convolutional Generative Adversarial Networks (DCGANs), achieving remarkable results with a discriminator loss of 0.0003 and a generator loss of 5.8206. These outcomes highlight the effectiveness of DCGANs in producing high-quality outputs and their significance in generative model development. Qamar et al. [15] investigate the multidisciplinary applications of Generative Adversarial Networks (GANs) and the implementation challenges they present. They provide a thorough overview of GANs' transformative impact across various sectors while addressing the difficulties researchers face in their deployment.

Hong et al. [16] introduce a Depth-aware Generative Adversarial Network (DaGAN) that excels in generating highly realistic human facial features, especially for occluded or partially visible faces. This innovative approach underscores the importance of depth awareness in generative models, showcasing significant advancements in realism and detail. Wang et al. [17] propose a Conditional Video GAN that utilizes fMRI data to analyze rapid brain perceptual processes. By linking slow blood oxygen-level dependent (BOLD) signals with swift brain activity, the model enables new insights into neural representations, paving the way for advancements in neuroscience and cognitive science. Liu et al. [18] demonstrate the creation of fake stereo audio for music using GANs, revealing a significant drop in detection accuracy from 99% to 30%, alongside a rise in the false acceptance rate from 0.08% to 69%. These results highlight the challenges GANs pose to current detection mechanisms in audio processing. Zhang et al. [19] present a generative adversarial network framework that consistently outperforms state-of-the-art techniques across diverse scenes and events, showcasing its reliability and effectiveness in handling complex generative tasks.

The Continuous Video Process (CVP) approach [20] models videos as continuous multi-dimensional processes, addressing the limitations of traditional discrete generative methods. By introducing a unique noise scheduling strategy and modeling frame generation as a bidirectional process between the start and end frames, CVP significantly enhances temporal consistency. On the KTH dataset, it achieves PSNR = 29.8 and SSIM = 0.872 for 30-frame prediction, demonstrating its effectiveness in long-range modeling without reliance on attention mechanisms. DFDNet [21] proposes a disentangling and filtering strategy for improved video prediction. It separates spatial dynamics into horizontal and

vertical components and applies a Fourier-based filter to remove transient high-frequency noise. This leads to superior prediction performance, with the highest reported PSNR = 35.11 and SSIM = 0.916 on the KTH dataset, highlighting its strength in noise suppression and motion disentanglement.

The MAUCell model [22] introduces a multi-attention framework combining temporal, spatial, and pixel-wise attention mechanisms within a GAN architecture. Despite a relatively low PSNR of 22.5, it achieves the highest SSIM of 0.935, emphasizing its focus on perceptual realism over traditional pixel-wise fidelity. A refined version of the CVP model, presented in [23], further streamlines diffusion sampling by reducing the number of steps by 75%, achieving the same metrics (PSNR = 29.8, SSIM = 0.872) while significantly improving computational efficiency.

Neural SDEs [24] offer a unified framework for continuous-domain sequence modeling by parameterizing stochastic differential equations with neural networks. This method excels in capturing complex temporal dynamics and reports PSNR = 27.55 and SSIM = 0.807 on KTH, favoring generalizability across sequential tasks. The RIVER model [25] applies sparsely conditioned flow matching in the latent space of a pretrained VQGAN. By conditioning on few keyframes and using warm-start sampling, it balances accuracy and efficiency, achieving PSNR = 30.4 and SSIM = 0.86 while reducing computational cost.

In [26], a state-space decomposition model is introduced to separately predict deterministic appearance and stochastic motion. The use of a temporal transformer allows modeling of long-term motion trends. The model shows strong performance with PSNR = 30.3, SSIM = 0.8766, and LPIPS = 0.0743, indicating both accuracy and perceptual quality. Lastly, MOSO [27] proposes a two-stage pipeline that decomposes videos into motion, scene, and object components. Using VQVAE for tokenization and a transformer for sequence modeling, MOSO facilitates flexible and modular video generation. It attains PSNR = 29.8, SSIM = 0.822, and LPIPS = 0.083, making it suitable for both conditional and unconditional video synthesis tasks.

2.2. Convolutional Neural Networks

Convolutional neural networks (CNNs) are a cornerstone of deep learning, designed to process and analyze two-dimensional data such as images and videos. They are a specialized subset of multilayer neural networks that extract hierarchical features through successive layers. Each layer applies learnable filters to detect specific characteristics of the input data, progressively building a detailed representation of the image or video [28]. CNNs consist of three primary types of layers: convolutional layers, pooling layers, and fully connected layers. The configuration of these layers is tailored to the complexity and requirements of the given task.

The convolutional layer, the core component of CNNs, uses a set of trainable filters as parameters, enabling the network to automatically learn relevant features directly from the data [29]. This contrasts with traditional methods where features were manually designed. Pooling layers reduce spatial dimensions while retaining essential information, making the network more computationally efficient and robust to spatial variations. Fully connected layers integrate the extracted features for classification or other tasks. CNNs excel in image and video processing tasks with minimal preprocessing compared to traditional methods [30]. Their adaptability also extends to array data, including audio signals, RGB images, and time-series data. This efficiency stems from their ability to learn complex patterns with fewer parameters and connections than fully connected networks, enabling faster training and improved scalability.

2.3. Recurrent Neural Networks

Recurrent Neural Networks, commonly referred to as RNNs, represent a specialized category of neural networks that are specifically engineered to handle and process sequential data. These networks achieve this by effectively modeling the intricate relationships and dependencies that exist between individual elements within a sequence. In contrast to conventional neural networks, which typically operate on fixed-size input data, RNNs are equipped with recurrent connections. This unique architectural feature enables RNNs to maintain a hidden state, which serves as a memory that captures and retains contextual information from previous inputs encountered in the sequence. As a result, RNNs are particularly well-suited for a variety of tasks that involve time series analysis, natural language processing, or any other type of data that exhibits temporal or sequential patterns. Their ability to remember and utilize information from earlier points in a sequence allows RNNs to perform effectively in applications where the order and timing of data points are crucial for accurate interpretation and prediction.

2.4. Generative Adversarial Network

These neural networks consist of two competing elements that improve through rivalry, based on a game theory approach where an adversarial process challenges the Generative deep learning network. A discriminator deep network distinguishes between outputs from the Generative network and real data. This competition enhances the learning and performance of both networks. Imagine a novice

showing their artwork to a master painter while claiming to be a professional. The artist identifies flaws, prompting the novice to improve. Over time, the differences between their work and that of a skilled painter diminish. The master painter, an expert in discrimination, collaborates with the novice, with the generator aiming to produce outputs indistinguishable from real data. The discriminator analyzes real and generated data to train a classifier to differentiate between them. In this analogy, D is a network, and G's role is to create counterfeits, learning to differentiate between real and fake currency. Both networks learn from each other, with the generative network refining its outputs based on the discriminator's feedback. The process continues until the quality of generative outputs satisfies the observer. After processing, input data consists of frames refined by a convolutional neural network, which enhances the image and prepares it for the GAN to generate video by understanding scene motion through optical flow.

3. The Proposed Method

The proposed method has three main phases that combine deep learning techniques to achieve a high-quality video.

Data Preprocessing and Feature Extraction: The video generation process begins with data preprocessing, where the input data is transformed into a series of frames. These frames undergo initial processing through a convolutional neural network (CNN) that improves the background of the image spatially and temporally. The CNN also adjusts the background to achieve minimum dimensionality and prepares the data for input to a Generative Adversarial Network (GAN). Optical flow analysis is used to extract motion features from the scene, enabling the GAN to distinguish between moving and stationary objects. This phase emphasizes feature extraction, creating a context to create realistic motion, and maintaining temporal consistency.

Video Frame Generation Using GAN: In this phase, the GAN uses the preprocessed data to generate video frames. A GAN based on Convolutional Architecture learns scene motion by training on a large dataset of unlabeled videos. The background is modeled separately to ensure that it remains stationary, while the motion of objects within the scene is analyzed and combined. Frames are fed into the GAN in a circular fashion: the initial frame is used to generate a video from which an image is extracted before any quality reduction. This real image is then fed back into the GAN, enabling the network to iteratively refine the generated frames. Convolution layers in the generator and discriminator models, along with inverse convolution layers, ensure that the final output meets the desired image dimensions and quality standards.

Video Assembly and Evaluation: The final stage involves assembling the generated frames into a coherent video. By combining the refined frames, this process ensures temporal consistency and high visual quality. The generated video is compared with the original video from the dataset to assess its accuracy and realism. Motion detection via optical flow and the GAN detector model helps validate the video quality. The network also learns to distinguish between moving objects and static backgrounds, ensuring a realistic representation of the scene. Finally, the GAN outputs a complete video with minimal quality loss, respecting both temporal and spatial criteria.

3.1. Identify the Movement of Objects in the Video

In visual gesture analysis, distinguishing between the background and foreground is essential, as the accuracy of the results significantly impacts effectiveness of VG. The four most common methods for motion detection include background differentiation, statistical techniques, and assessing temporal and sharpness differences. Optical flow, capable of tracking moving subjects despite camera movement and background noise, is preferred in motion detection applications. Thus, the proposed system calculates velocity in each frame using optical flow.

Optical flow analysis detects motion between consecutive frames, allowing for the separation of moving foreground objects from the static background during the preprocessing phase. These extracted motion vectors are then fed into the GAN, which helps create realistic object movements in the generated frames, thereby avoiding unnatural artifacts such as sudden jumps.

The Horn-Schunck and Lucas-Kanade formulas are frequently utilized for this purpose. While Lucas is a local approach and Horn is global, Lucas operates faster and is less affected by noise due to its local computations, making it the method of choice for calculating sharpness. Lucas-Kanade was selected for its KTH dataset compatibility. Its accuracy in detecting foreground motion with strong gradients (e.g., walking) and lower computational cost (40% faster than Horn-Schunck) enabled training on limited hardware. Its noise resistance allowed accurate tracking of moving objects in dynamic KTH scenes, improving video frame realism and RNN stability. Horn-Schunck's complexity made it less suitable for long videos and non-uniform motion.

The previous frame is required to calculate the derivatives I_x, I_y and I_t. The previous frame is

maintained and updated with each frame entry. In Lucas, the optical flow is assumed to be fixed in the horizontal and vertical directions (V_x and V_y) in a small image window of size $M * M$. Therefore, the matrices I_x, I_y and I_t are windowed with dimensions N= $M * M$, Equation (1) is obtained from each window. Where V_x and V_y can be calculated by Equation (2) for each window [29].

$$\begin{aligned} I_{x1}\, V_x + I_{y1} V_y &= -I_{t1} \\ I_{x2}\, V_x + I_{y2} V_y &= -I_{t2} \\ I_{xd}\, V_x + I_{yd} V_y &= -I_{td} \end{aligned} \quad (1)$$

$$\begin{bmatrix} Vx \\ Vy \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} I_{xi}^2 & \sum_{i=1}^{N} I_{xi}\, I_{yi} \\ \sum_{i=1}^{N} I_{xi}\, I_{yi} & \sum_{i=1}^{N} I_{yi}^2 \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{N} I_{xi}\, I_{ti} \\ \sum_{i=1}^{N} I_{xi}\, I_{ti} \end{bmatrix} \quad (2)$$

For each pixel in the input frame, two integers are generated to indicate its horizontal and vertical displacement, stored in matrices OF_x and OF_y. These matrices match the input frame's dimensions, ensuring a direct correlation between pixel movements and their locations. By analyzing these displacement values, the system can detect frames with little or no motion. A frame is considered stationary if the maximum values in both OF_x and OF_y are below zero, indicating no movement. This identification of motionless frames is crucial for the system's operation, facilitating smoother transitions to processing the next frame against a static background.

3.2. Convolutional Generative Adversarial Networks

Using a combination of convolutional neural networks (CNN) and generative adversarial networks (GAN) for image generation leverages the strengths of both architectures effectively. CNNs are frequently utilized for feature extraction in images. They consist of layers that apply convolution operations to identify patterns like edges, textures, and more complex structures. Additionally, CNNs can be integrated into encoder-decoder architectures for image processing and transformation.

A GAN comprises two neural networks: a generator and a discriminator. The generator aims to create realistic images from random noise or low-dimensional input, learning to mimic real data. Meanwhile, the discriminator's role is to differentiate between real and fake images, determining whether an image comes from the dataset or is generated. This creates a competitive dynamic where the generator attempts to deceive the discriminator, while the discriminator enhances its ability to identify fakes. Over time, both networks improve, with the generator producing more realistic images as the discriminator becomes better at detecting them.

CNNs are utilized in both the generator and discriminator of GANs. The generator employs CNN layers to transform input noise into high-quality images, while the discriminator uses CNNs to extract hierarchical features and determine the authenticity of the images. This synergy harnesses the feature extraction capabilities of CNNs alongside the adversarial training of GANs to generate realistic images. Figure 1 depicts CGAN architecture.

3.3. Circular Network

After the CGAN generates a sequence of images, the RNN comes into play to add temporal coherence. RNNs are great at capturing temporal dependencies and patterns. By processing the frames produced by the CGAN as a sequence, the RNN learns to predict temporal dynamics, such as movement and transitions between frames. This ensures that the resulting video is not only visually consistent, but also displays a realistic time flow, mimicking how events occur in real-world scenarios.

To achieve this goal, we employ the circular network approach. If the distinguishing error percentage increases with the generated sample, the Generative model is also updated in response to this change. Each time an input image is fed into the generative adversarial network, a movie will be created following the training of the network through the circular network technique. This process aligns with the proposed concept utilized in this problem, integrating the circular network within the overall framework of the network. It is essential to capture the previous image and reintroduce it to the network to prevent any loss in video quality and to avoid any potential blurring effects that may occur over time. The evaluation criteria that will be elaborated on in the subsequent section are employed to assess and measure the quality of the images produced. Ultimately, this technique results in the creation of a video, which is enhanced in duration through multiple repetitions and

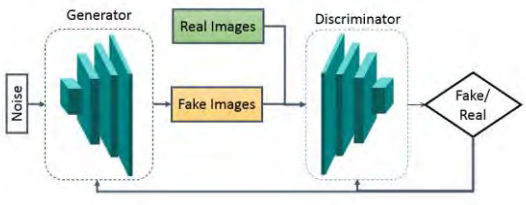

Figure. 1. Content clustering of movies

combinations of frames, ensuring a richer viewing experience. The careful integration of these elements not only preserves the integrity of the visuals but also contributes to a more cohesive and polished final product. The architecture iteratively refines the generated frames using a feedback loop to increase quality and reduce temporal/spatial anomalies. Each frame is evaluated using *PSNR* and *SSIM*. If its quality falls below a threshold (e.g., *PSNR* < 30), it is returned to the network as a new input. This dynamic iteration continues until the desired quality is achieved, up to 10 iterations. Low-quality frames are reprocessed with modifications, including trimming to correct for unnatural motion via GAN weight settings and updated input noise to increase output diversity. This process is repeated, producing improved frames until the quality criteria are met. The number of iterations is adaptive and determined by the quality criteria. To avoid overfitting, a maximum of 10 iterations is applied, and an early stop is initiated if the *PSNR* improvement between iterations is less than 0.5 or if *Sharp.diff* remains constant for three consecutive iterations. Figure 2 shows the proposed architecture.

Figure 2. The architecture of a Conditional Generative Adversarial Network (CGAN), which combines convolutional neural networks (CNNs) and generative adversarial networks (GANs) for realistic image generation. CNNs are used in both the generator—for transforming input noise into images—and the discriminator—for extracting hierarchical features to distinguish real from generated images.

The networks were pre-trained separately: the CNN on KTH for image feature extraction, the RNN on real frame sequences for temporal pattern learning, and the CGAN on static image generation for realism. End-to-end joint training then integrated the system, initialized with random noise. The CNN extracted features, the GAN generated frames from features and noise, and the RNN analyzed these frames for temporal stability. A hybrid loss function, combining mean squared error (pixel quality) and *SSIM* (adversarial loss for visual coordination), improved GAN realism. We used Wasserstein GAN with a gradient penalty to stabilize joint training and prevent equilibrium collapse. Adam optimization with an adaptive learning rate (initial rate 0.0001, gradually reduced) and Batch Normalization addressed slow convergence and reduced initialization sensitivity. To bridge CNN-extracted features and RNN requirements, Fully Connected layers converted spatial features to temporal data, and stepwise training prioritized RNN training after CNN stabilization. These solutions – WGAN, batch tuning, and hybrid loss – overcame the challenges of joint CNN, RNN, and GAN training, resulting in improved video quality, temporal stability, and motion realism.

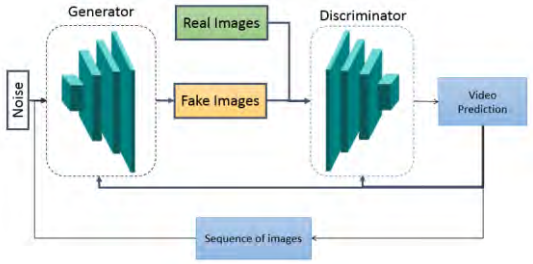

Figure. 2. The architecture of a Conditional Generative Adversarial Network (CGAN), which combines convolutional neural networks (CNNs) and generative adversarial networks (GANs) for realistic image generation. CNNs are used in both the generator—for transforming input noise into images—and the discriminator—for extracting hierarchical features to distinguish real from generated images.

4. Evaluating the Results

All implementations were carried out in Python, with a focus on generating the final video through the creation of successive frames, motion detection, and motion separation using optical flow and a Circular Generative Network (CGN). The Python implementation leverages libraries such as TensorFlow, OpenCV, NumPy, and Scikit-image. The computations were performed on a system configured with 32 GB of RAM, a Core i7 CPU, and a GeForce GTX 1050 Ti GPU.

Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) and Sharpens are used to compare the methods [32]. PSNR, often regarded as the most common metric for assessing image quality, is defined by Equation (3).

$$PSNR = 10, \log_{10} \frac{MAX^2}{MSE} \tag{3}$$

Where, MAX is maximum possible pixel value, MSE is calculated by Equation (4).

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (F(i,\ j) - H(i,\ j))^2 \tag{4}$$

Where, *I* is reference image, *H* denotes distorted image and *M, N* are dimensions of the image. *SSIM* is a perceptual metric that evaluates image quality by comparing structural information, luminance, and contrast. It's designed to better align with human visual perception compared to *PSNR*. SSIM denotes by Equation (5).

$$\begin{aligned} SSIM &= \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \\ \mu_x &= \sum_{i=1}^{N} W_i, X_i \\ \sigma_x &= \left(\sum_{i=1}^{N} w_i\,(X_i - \mu_i)\right)^{\frac{1}{2}} \\ \sigma_{xy} &= \sum_{i=1}^{N} w_i\,(X_i - \mu_i)(Y_i - \mu_i) \end{aligned} \tag{5}$$

Sharpness criterion compares the sharpness of the projected image to the original (sharp.diff) by assessing the gradient difference between the two images, X and Y, as defined in Equation (6).

$$sharp, diff(x, y) = 10\log_{10}\frac{max_x^2}{\frac{1}{N}\left(\sum_i\sum_j\left|\left(\nabla_i Y + \nabla_j Y\right) - \left(\nabla_i X + \nabla_j X\right)\right|\right)} \quad (6)$$

$$\nabla_i Y = \left|Y_{i,j} - Y_{i-1,j}\right| \text{ and } \nabla_j Y = \left|Y_{i,j} - Y_{i,j-1}\right|$$

4.1. Dataset

The KTH dataset is a prominent resource for activity identification, utilized in papers [31, 32]. It encompasses various actions, including Walking, Jogging, Running, Boxing, Hand Waving, and Hand Clapping. The dataset features 25 participants performing exercises across four scenarios: an open environment, an outdoor setting with variable scale, an open space with changing clothing and light intensity, and an indoor area. Each video showcases a single individual, and the camera may shake, zoom in, or out. The KTH dataset consists of 600 videos, all captured at 25 frames per second with a stationary camera in static surroundings. Each sequence has a spatial resolution of 120x160 pixels and an average duration of four seconds. Table 1 shows the features of the KTH dataset, and Figure 3 depicts activities from this dataset, which includes videos of specific activities. The dataset can also incorporate other similar datasets. It comprises 100 video segments, each titled with the person's number, the walking activity, and the relevant scenario number. To prepare the dataset for network processing, each video must be converted into frames. This conversion ensures the accuracy of the generated videos, which can be verified against the original until the network has processed the images and produced the final movie. The walking activity class has been utilized in this section.

KTH's widespread use allows direct comparison with existing research. Key features include static backgrounds and efficient video specifications (160x120, 25 FPS), simplifying motion analysis. Datasets like UCF-101 (too diverse), Hollywood2 (cinematic, inaccurate labels), and DAVIS (segmentation-focused) were deemed unsuitable. KTH's simplicity, structure, and comparability make it optimal for focusing on basic movement generation. More complex datasets can be considered for model development.

4.2. The Results

In this study, we designed an experiment to evaluate the impact of network loop configurations and output frame counts on video frame generation quality. Two primary scenarios were tested: one with 6 loops and 5 output frames, and another with 5 loops and 3 output frames. The dataset comprising 25 course-related videos was used, each processed into 14,090 individual frames using the get-frame method for frame extraction.

For manageability and analysis consistency, each video was segmented into multiple parts, with each segment containing a minimum of 400 frames. During the initial experimental phase, we applied both configurations (6×5 and 5×3) across the segmented videos. The results from this phase are presented in Table 2.

Training was conducted with a learning rate of 0.00004 for 100 iterations. Preliminary findings suggest that increasing the number of loops while decreasing the number of output frames per loop contributes to improved frame quality and resolution. Based on these insights, future iterations of the experiment will involve adjusting the network architecture to support a greater number of loops, to generate longer sequences of high-quality frames.

4.3. Sensitivity Analysis

We trained the network for 100 iterations using learning rates of 0.04, 0.00004, and 0.000004, along with various neuron configurations for the hidden layer. The findings indicated that 0.00004 was the optimal learning rate This learning rate is ideal for the Generative network and should be compact enough to fit within the discriminator network for simultaneous training. Due to the wide range of

Table 1. Specifications for the KTH dataset

Property	*Value*
Count of activity classes present in the dataset	6
Mean number of samples for each class	100
Maximum resolution (pixels)	120 × 160
Point of view type	Side-Front
Minimum and maximum video length	204-1492
Sampling rate (FPS)	25

Table 2. The results

Scenarios	*PSNR*	*Sharp. diff*	*Ds*
#1	21.44	10.73	6 loops and 5 production frames
#2	22.32	11.21	5 loops and 3 production frames

Figure. 3. Activities in the dataset

values, results are presented in a logarithmic format. Figure 4 shows the PNSR at different learning rates for experiences #1 and #2.

Figure 5 shows the sharpness at different learning rates for experiences #1 and #2.

With a learning rate of 0.00004, we have trained the network using a variety of repetitions. The Cross-Entropy function has also been utilized as a loss function. The findings, whose values are in Table 3, show that the loop generative network's optimum state necessitates a match between its several key parameters including the number of iterations, the learning rate, and the number of loops, as well as the total number of generated frames.

Increased iterations enhance network learning of motion and detail through feedback. *PSNR* reaches 35.8 at 1 million iterations, indicating reduced noise and increased sharpness. Table 2 shows improved *PSNR* and *Sharp.diff* with more iterations due to:

1) Progressive GAN learning: Early iterations produce blurry/noisy frames (low PSNR), while later iterations learn subtle features, increasing clarity (*PSNR* ~35.8).

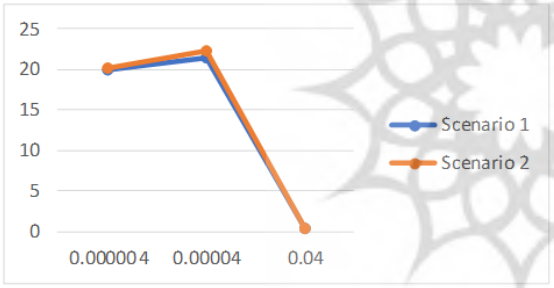

Figure. 4. PSNR results from different learning rates

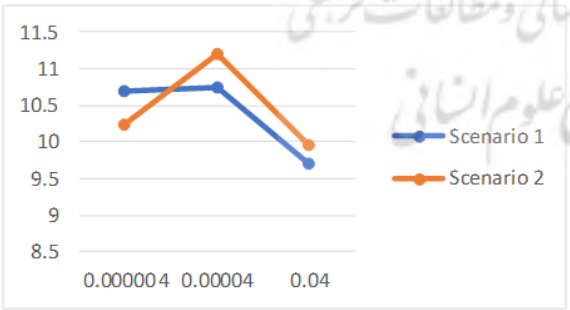

Figure. 5. Sharpness results from different learning rates

Table 3. Results of VG with different repetitions

iteration	*PSNR*	*Sharp.diff*	*SSIM*
10	16/04973	10/21199	0/32
100	18/56934	10/86214	0/47
1000	19/26552	13/54955	0/68
10000	22/65245	13/75139	0/78
100000	28/65246	20/56588	0/89
1000000	35/86234	25/51526	0/96

2) Enhanced RNN temporal coordination: Low iterations yield discontinuous changes; high iterations create smooth, realistic motion patterns by modeling long-term dependencies.

3) Circular CGAN feedback loop: Corrects blurring/incoherence. Visually, higher iterations improve facial/hand detail clarity, and clothing texture. Temporal stability is established, transitioning from abrupt changes to smooth movements. Motion realism improves (e.g., KTH punching). Realism score increases from 2.1 (100 iterations) to 4.3 (1,000,000 iterations). Motion stability rises from 1.2s to 4.5s. Improved *PSNR/Sharp.diff* aligns with visual clarity, stability, and realism.

The CGAN feedback loop enhances frame quality by iteratively refining generated frames. Each frame is evaluated based on PSNR, SSIM, and detail sharpness. If any metric falls below a threshold, the frame is fed back into the network for correction, which includes detecting abnormal joint motion and adjusting GAN weights and input noise to promote output variation. This dynamic evaluation, corrective unwrapping, and smart stopping mechanism prevent poor quality and blurring, ensuring both high-quality frames and efficient resource utilization for spatially and temporally stable, realistic videos.

As Table 4 demonstrates, our proposed method outperforms all existing approaches across key video metrics, including *PSNR* and *SSIM*. Unlike prior models that often trade off temporal coherence for spatial quality—or vice versa—our architecture integrates convolutional, recurrent, and generative adversarial components alongside a novel splicing and loop feedback mechanism. This design ensures both high-fidelity frame generation and stable motion continuity over longer sequences. On the KTH dataset, our model achieves a *PSNR* of 35.8 and *SSIM* of 0.96, representing a significant performance gain of up to 20% over previous state-of-the-art methods. Moreover, our model generates 10-second videos at a resolution of 720×128 with minimal perceptual noise, setting a new benchmark in realistic and coherent video prediction.

Table 4. Compare the results

#Ref	*Year*	*SSIM*	*PSNR*
[20]	2024	0.872	29.8 (30 frames)
[21]	2025	0.916	35.11 (20 frames)
[22]	2025	0.935	22.5
[23]	2024	0.872	29.8
[24]	2025	0.807	27.55
[25]	2023	0.86	30.4
[26]	2024	0.8766	30.3
[27]	2023	0.822	29.8 (40 frames)
Proposed	2025	**0/96**	**35/86234**

Figure 6 for signal-to-noise ratio, Figure 7 for image resolution, and Figure 8 for structural similarity demonstrate the outcomes of our studies on the specified dataset with various iterations. As can be observed, the evaluation criteria in all three parts improve when the number of repetitions rises. This improvement will naturally grow somewhat as the evaluation criteria are evaluated from one value onward and in high repetitions. The convolutional generative network research revealed that 1,000,000 iterations is the ideal number of iterations for modeling.

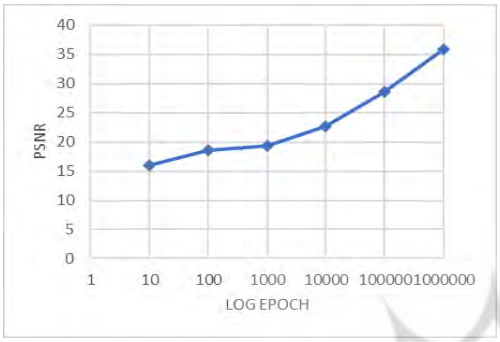

Figure. 6. PSNR evaluation benchmark results

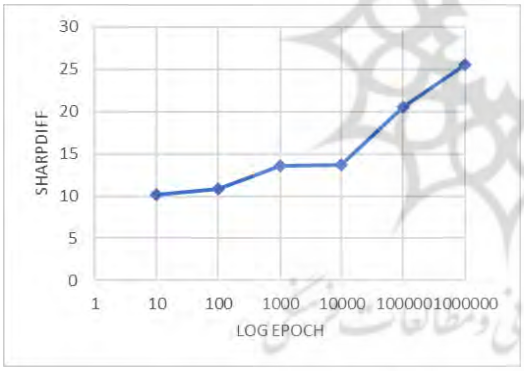

Figure. 7. Sharpness evaluation benchmark results

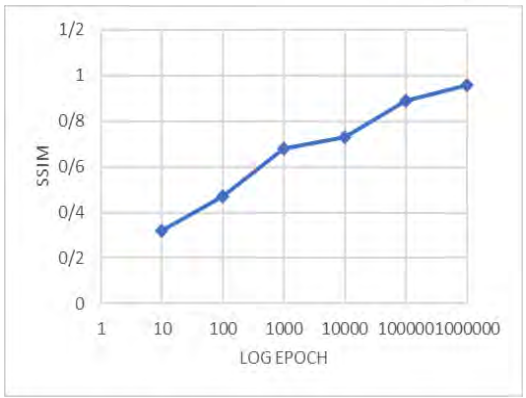

Figure. 8. SSIM evaluation benchmark results

The proposed network provides the production frames, which are high-quality as illustrated in Figure 9. The show frame function is used in this stage to turn a series of frames into a video. It should be mentioned that this movie may now be evaluated for quality and accuracy by comparing it to REAL footage. The original video contains visuals referred to as gt stands for round truth (REAL imagery). In other words, non-simulation-generated photographs were included for better comparison. Images produced by the Circular Generative Network are likewise images produced by the term (generated).

5. Conclusion

This research investigated the integration of generative systems and motion detection within a comprehensive framework for video generation and

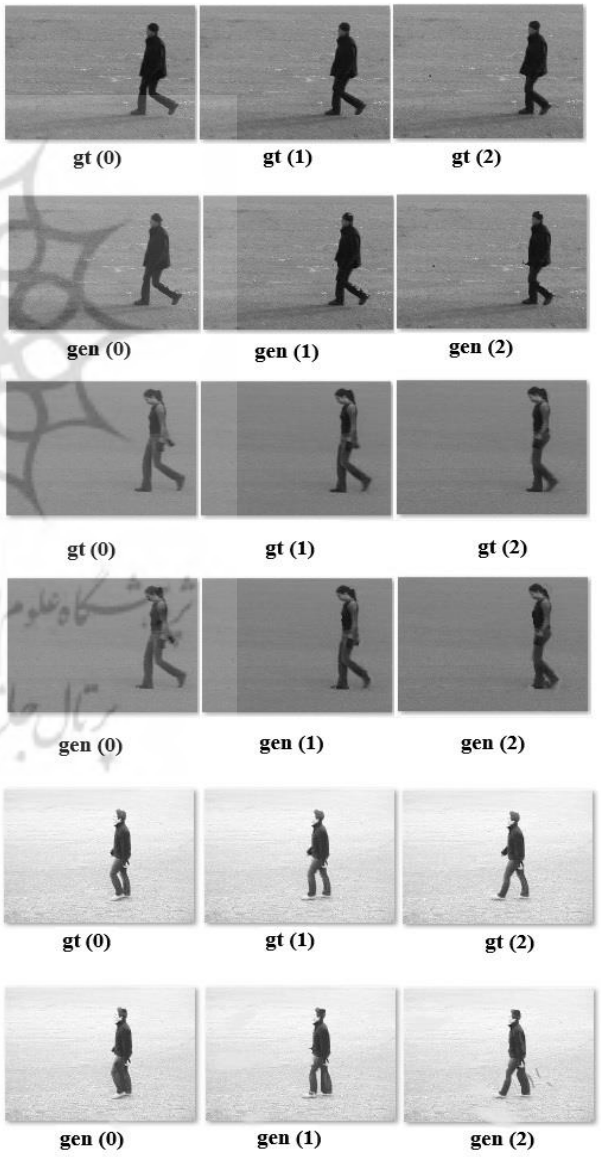

Figure. 9. An example of the frames produced

object motion analysis. The basic concepts, terminology, and architectural principles of generative networks and motion detection are first introduced, followed by an in-depth review of existing methods for motion detection and video synthesis using generative networks.

The key contribution of this work is the use of a generative circular network that uses optical flow detection and repeated frame retrieval to analyze object motion and generate video content. The process began with optical flow detection to capture motion, followed by the use of repeated loops to extract consecutive frames. A random vector was used as input to the neural networks, which allowed for the generation of coherent video sequences. The KTH dataset served as the main source of images, which underwent various stages of processing to produce realistic video outputs.

The generative adversarial network (GAN) was trained on a sequence of consecutive frames generated in the initial phase. After scene perception, the image was segmented into foreground and background to facilitate motion detection, separating moving objects. The background remained static, while the foreground was dynamically processed. This iterative process involved evaluating the image quality using predefined criteria, modifying the output frame, and feeding it back to the network as the initial frame for subsequent iterations. The final output demonstrated the effectiveness of the model, with the generated videos exhibiting high-quality motion composition and alignment with theoretical expectations.

The proposed model, trained on the KTH dataset, exhibited superior performance in video generation, validating the effectiveness of the approach. This research not only advances the understanding of generative networks in video generation, but also provides a practical framework for future applications in motion analysis and artificial media creation. Further exploration and optimization of this model could yield more robust solutions to complex video generation challenges.

The primary limitation is video quality degradation in sequences exceeding 10 seconds, resulting from accumulated frame errors and the model's struggle to maintain long-term coherence in complex movements. Furthermore, the model is suboptimal for scenes with dynamic lighting or rapid motion, potentially introducing noise or blur. Future research should focus on incorporating Multi-Head Attention into the RNN to better capture long-term dependencies and leveraging transfer learning to enhance adaptability to diverse scenes. A Reinforcement Learning-based optimization framework could also improve training stability and output quality by automatically adjusting GAN and RNN parameters. Finally, extending the method to virtual reality and interactive video generation with user input presents a promising research avenue.

Declarations

Funding

This research did not receive any grant from funding agencies in the public, commercial, or non-profit sectors.

Authors' contributions

FS: Study design, acquisition of data, interpretation of the results, statistical analysis, drafting the manuscript.

AM: Acquisition of data, interpretation of the results, statistical analysis, drafting the manuscript.

MM: Supervision, interpretation of the results, revision of the manuscript.

MD: Supervision, interpretation of the results, revision of the manuscript.

Conflict of interest

The authors declare that there is no conflict of interest.

References

[1] P. Zhou, et al., "A survey on generative ai and llm for video generation, understanding, and streaming," arXiv preprint arXiv:2404.16038. 2024 Jan 30, https://doi.org/10.48550/arXiv.2404.16038.

[2] A. Khang, V. Abdullayev, E. Litvinova, S. Chumachenko, A. V. Alyar, and P. T. Anh, "Application of Computer Vision (CV) in the Healthcare Ecosystem," In Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem, CRC Press, 2024, pp. 1-16.

[3] J. Walker, K. Marino, A. Gupta, and M. Hebert, "The pose knows: Video forecasting by generating pose futures," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3332-3341, https://openaccess.thecvf.com/content_iccv_2017/html/Walker_The_Pose_Knows_ICCV_2017_paper.html.

[4] Tang Y, Bi J, Xu S, Song L, Liang S, Wang T, Zhang D, An J, Lin J, Zhu R, Vosoughi A. Video understanding with large language models: A survey. IEEE Transactions on Circuits and Systems for Video Technology. 2025 May 2.

[5] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, M. Parmar, "A review of convolutional neural networks in computer vision," Artificial Intelligence Review, vol. 57, no. 4, p. 99, 2024 Mar 23, https://doi.org/10.1007/s10462-024-10721-6.

[6] S. M. Al-Selwi et al., "RNN-LSTM: From applications to modeling techniques and beyond—Systematic review," Journal of King Saud University-Computer and Information Sciences, vol. 36, no. 5, p.102068, 2024, https://doi.org/10.1016/j.jksuci.2024.102068.

[7] S. Gupta, A. Keshari and S. Das, "Rv-gan: Recurrent gan for unconditional video generation," In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 2024-2033, https://openaccess.thecvf.com/content/CVPR2022W/WiCV/html/Gupta_RV-GAN_Recurrent_GAN_for_Unconditional_Video_Generation_CVPRW_2022_paper.html.

[8] Z. Xing, Q. Dai, H. Hu, Z. Wu and Y. G. Jiang, "Simda: Simple diffusion adapter for efficient video generation," In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 7827-7839, https://openaccess.thecvf.com/content/CVPR2024/html/Xing_SimDA_Simple_Diffusion_Adapter_for_Efficient_Video_Generation_CVPR_2024_paper.html

[9] W. Xiong, W. Luo, L. Ma, W. Liu and J. Luo, "Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2364-2373, https://openaccess.thecvf.com/content_cvpr_2018/html/Xiong_Learning_to_Generate_CVPR_2018_paper.html.

[10] K.Vougioukas, P. Ma, S. Petridis and M. Pantic, "Video-driven speech reconstruction using generative adversarial networks," arXiv preprint arXiv:1906.06301, 2019, https://doi.org/10.48550/arXiv.1906.06301.

[11] J. Chen, Y. Li, K. Ma and Y. Zheng, "Generative adversarial networks for video-to-video domain adaptation," In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, pp. 3462-3469. 2020, April., https://doi.org/10.1609/aaai.v34i04.5750.

[12] R. Mira, K. Vougioukas, P. Ma, S. Petridis, B. W. Schuller and M. Pantic, "End-to-end video-to-speech synthesis using generative adversarial networks," In IEEE Transactions on Cybernetics, vol. 53, no. 6, pp. 3454-3466, June 2023, https://doi.org/10.1109/TCYB.2022.3162495.

[13] L. Lan and C. Ye, "Recurrent generative adversarial networks for unsupervised WCE video summarization." Knowledge-Based Systems, vol. 222, p. 106971, 2021, https://doi.org/10.1016/j.knosys.2021.106971.

[14] S. Singh, B. Aggarwal, V. Bhardwaj and A. Kumar, "Motion Transfer in Videos using Deep Convolutional Generative Adversarial Networks," Smart and Sustainable Intelligent Systems, pp. 205-214, 2021, https://doi.org/10.1002/9781119752134.ch15.

[15] R. Qamar, N. Bajao, I. Suwarno and F. A. Jokhio, "Survey on Generative Adversarial Behavior in Artificial Neural Tasks," Iraqi Journal For Computer Science and Mathematics, vol. 3, no. 2, pp. 83-94, 2022, https://doi.org/10.52866/ijcsm.2022.02.01.009.

[16] F. T. Hong, L. Zhang, L. Shen and D. Xu, "Depth-Aware Generative Adversarial Network for Talking Head Video Generation," In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 3397-3406, https://openaccess.thecvf.com/content/CVPR2022/html/Hong_Depth-Aware_Generative_Adversarial_Network_for_Talking_Head_Video_Generation_CVPR_2022_paper.html

[17] C. Wang et al., "Reconstructing Rapid Natural Vision with fMRI-Conditional Video Generative Adversarial Network," Cerebral Cortex, vol. 32, no. 20, pp. 4502-4511, 2022, https://doi.org/10.1093/cercor/bhab498.

[18] T. Liu, D. Yan, N. Yan and G. Chen, "Anti-forensics of fake stereo audio using generative adversarial network," Multimedia Tools and Applications, vol. 81, no. 12, pp. 17155-17167, 2022, https://doi.org/10.1007/s11042-022-12448-4.

[19] Z. Zhang, S. H. Zhong, A. Fares and Y. Liu, "Detecting abnormality with separated foreground and background: Mutual generative adversarial networks for video abnormal event detection," Computer Vision and Image Understanding, vol. 219, p. 103416, 2022, https://doi.org/10.1016/j.cviu.2022.103416.

[20] G. Shrivastava and A. Shrivastava, "Video prediction by modeling videos as continuous multi-dimensional processes," In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 7236-7245, https://openaccess.thecvf.com/content/CVPR2024/html/Shrivastava_Video_Prediction_by_Modeling_Videos_as_Continuous_Multi-Dimensional_Processes_CVPR_2024_paper.html.

[21] L. Gan, J. Lai, J. Ju, L. Gao and Y. Bin, "DFDNet: Disentangling and Filtering Dynamics for Enhanced Video Prediction," In Proceedings of the AAAI Conference on Artificial Intelligence, 2025 Apr 11, vol. 39, no. 3, pp. 3059-3067, https://doi.org/10.1609/aaai.v39i3.32314.

[22] S. Gupta, P. Agrawal, P. Gupta, "MAUCell: An Adaptive Multi-Attention Framework for Video Frame Prediction," arXiv preprint arXiv:2501.16997, 2025 Jan 28, https://doi.org/10.48550/arXiv.2501.16997.

[23] G. Shrivastava and A. Shrivastava, "Continuous Video Process: Modeling Videos as Continuous Multi-Dimensional Processes for Video Prediction," arXiv preprint arXiv:2412.04929, 2024 Dec 6, https://doi.org/10.48550/arXiv.2412.04929.

[24] M. Shen and C. Cheng, "Neural SDEs as a Unified Approach to Continuous-Domain Sequence Modeling," arXiv preprint arXiv:2501.18871, 2025 Jan 31, https://doi.org/10.48550/arXiv.2501.18871.

[25] A. Davtyan, S. Sameni and P. Favaro, "Efficient video prediction via sparsely conditioned flow matching," In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 23263-23274, https://openaccess.thecvf.com/content/ICCV2023/html/Davtyan_Efficient_Video_Prediction_via_Sparsely_Conditioned_Flow_Matching_ICCV_2023_paper.html.

[26] F. Cui et al., "State-space Decomposition Model for Video Prediction Considering Long-term Motion Trend," arXiv preprint arXiv:2404.11576, 2024 Apr 17, https://doi.org/10.48550/arXiv.2404.11576.

[27] M. Sun, W. Wang, X. Zhu and J. Liu "Moso: Decomposing motion, scene and object for video prediction," In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 18727-18737, https://openaccess.thecvf.com/content/CVPR2023/html/Sun_MOSO_Decomposing_MOtion_Scene_and_Object_for_Video_Prediction_CVPR_2023_paper.html.

[28] I. Arel, D. C. Rose and T. P. Karnowski, "Deep machine learning-a new frontier in artificial intelligence research," IEEE computational intelligence magazine, vol. 5, no. 4, pp. 13-18, 2010, https://doi.org/10.1109/MCI.2010.938364.

[29] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in Proceedings of the 25th international conference on Machine learning, 2008, pp. 160-167, https://doi.org/10.1145/1390156.1390177

[30] A. Van den Oord, S. Dieleman and B. Schrauwen, "Deep content-based music recommendation," in Advances in neural information processing systems, 2013, pp. 2643-2651, https://dl.acm.org/doi/10.5555/2999792.2999907.

[31] V. Sze, Y. H. Chen, T. J. Yang and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," Proceedings of the IEEE, vol. 105, no. 12, pp. 2295-2329, 2017, https://doi.org/10.1109/JPROC.2017.2761740.

[32] M. Yang, F. Lv, W. Xu, K. Yu and Y. Gong, "Human action detection by boosting efficient motion features," in 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, IEEE, 2009, pp. 522-529, https://doi.org/10.1109/ICCVW.2009.5457656.

Fatemeh Sobhani Manesh received her B.Sc. in software engineering in 2014 from Lorestan University. She graduated with an MSc degree in artificial intelligence from Hamadan University in 2019. Her research interests include image processing, data mining, neural networks, and deep learning.

Amin Nazari received BSc degree in Computer Software Engineering from Islamic Azad University, Hamedan, in 2009. He received his MSc degree in Computer Software Engineering from Arak University, Arak, in 2015. He is now a Ph.D. candidate of artificial intelligence at the Bu-Ali Sina University, Hamedan. His research interests include IoT-fog networks, machine learning and recommender systems.

Muharram Mansoorizadeh is an associate professor at the Computer Engineering Department of Bu-Ali Sina University. He received his BSc degree in software engineering from the University of Isfahan, Isfahan, Iran, in 2001, and his MSc degree in software engineering and the PhD in computer engineering from Tarbiat Modares University, Tehran, Iran, in 2004 and 2010, respectively. His current research interests include machine learning, affective computing and information retrieval.

Mir Hossein Dezfoulian received the B.Sc. and M.Sc. degrees in Computer Science from Sharif University in 1987 and the Ph.D. degree in Computer Science–Pattern Recognition from Wollongong University, Australia in 1995. Since 1988, He has been with the Bu-Ali Sina University and currently He is an Assistant Professor of Computer Engineering, Head of the Computer Engineering Department in Bu-Ali Sina University. His current research interests include Statistical Pattern Recognition, Question and Answering Systems and Fuzzy Systems.