Indian Journal of Pure & Applied Physics
Vol. 63, November 2025, pp. 994-1002
DOI: 10.56042/ijpap.v63i11.21372

Objective Enhancement for Image and Video Compression Using Feature Extraction and Fast RNN-Based Motion Estimation Optimisation

Mudhavath Ramesh Naik* & Jayendra Kumar

National Institute of Technology Jamshedpur, Department of Electronics and Communication Engineering, Jamshedpur 831 014, India

Received:7th July 2025; accepted: 19th September 2025

Enhancing compressed visual content remains challenging due to visual degradation, motion distortions, and poor temporal coherence. Existing methods often fail to balance detail preservation with accurate motion estimation, especially under high compression or motion. To address this, we introduce the Faster-Recurrent Neural Network-Swarm Intelligence Metaheuristic of the CT Optimisation Algorithm (F-RNN-SIMCT), a novel method combining a fast recurrent neural network with swarm intelligence inspired by coyotes and tuna fish. This hybrid approach optimises motion estimation and preserves spatial-temporal details under harsh compression. F-RNN-SIMCT leverages advanced feature extraction and metaheuristic optimisation to improve motion accuracy and perceptual quality. Experiments show it outperforms standard methods, making it suitable for video transmission and storage in bandwidth-limited environments.

Keywords: Video compression, Deep learning, Faster recurrent neural network, Coyote optimisation, Tuna swarm optimisation algorithm

1 Introduction

Video content now dominates global web traffic, mainly due to widespread mobile device usage for work and entertainment[1,2]. With the rise of high-resolution formats like 4K UHD and VR 360, efficient video compression is more critical than ever. Traditional codecs such as VP9, H.264, and H.265 have performed well but are designedprimarily for human perception[3-5]. As deep learning applications grow, these manually engineered codecs require optimisation for machine-specific tasks. Recent advances use deep neural networks (DNNs), especially auto encoders, to improve compression efficiency and representation[6-8]. While image compression focuses on spatial redundancy, video compression must also capture temporal patterns. Replacing conventional block-based motion prediction with learned models is key to deep learning-based video optimisation[9,10].

Despite advances in video forecasting using self-attention and GANs, video compression remains challenging[11]. Current approaches often integrate key features or reuse segments via deep neural networks (DNNs), especially auto encoderlike architectures that map frames to latent spaces[12,13]. However, many video encryption methods are computationally expensive, particularly in motion detection,the most demanding part of video encoding. To address this, we propose a more efficient solution that reduces computational complexity while enhancing compression artefact removal[14-16]. Our approach uses a non-local ConvLSTM model to extract spatio-temporal features from surrounding frames, improving motion estimation and visual quality[17-18].

To accelerate the non-local module, we propose an efficient approximation method for computing inter-frame pixel wise similarity, avoiding the limitations of traditional combinatorial optimisation, such as poor boundaries from fine quantisation. Given the sensitivity of video content, ranging from financial and medical data to military intelligence, securing compressed video is essential[19-21]. Our research introduces a motion estimation solution using F-RNN-SIMCT to enhance compression quality and security. Key contributions include:

i A filtering method for noise removal during preprocessing to improve image and video quality.
ii An optimised feature extraction technique using Coyote and Tuna Swarm Optimisation, enabling more effective segmentation.

The structure of the paper consists of a review of the literature, the suggested approach in Section 3, the findings and discussion in Section 4, and a

*Corresponding author: E-mail: rameshnaik466@gmail.com

conclusion with recommendations for further work in Section 5.

2 Material

Learning based video compression has gained increasing attention[22]. Earlier hybrid coding methods focused on removing spatial and temporal redundancies in pixel space but faced limitations in motion estimation and compensation. Field-Coding Video Networks (FCV) were introduced to address this, simulating all key compression steps: motion estimation, compression, compensation, and residual encoding within feature space using auto encoder-based architectures. However, improving coding efficiency within traditional hybrid frameworks remains challenging[23-24].

Deep convolutional neural networks (CNNs) have shown great success in AI and signal processing, offering potential for video/image compression[25,26]. While traditional methods use predictive coding for residuals and motion[27], some combine conventional architectures with CNNs for compression frameworks[28], including learning-based motion estimation[29]. However, these often lack integrated motion complexity and global adaptability. Techniques like adaptive thresholding and Huffman encoding improve performance[30,31], but lack end-to-end optimisation for motion adaptability. Even RaFC[32] focuses on stream resolution without unified spatio-temporal modelling.

We propose a deep video compression model integrating motion estimation, compensation, and residual processing to overcome these limitations. Using LIGAN-CFNet, we combine GAN-based perceptual learning with Conv-LSTM temporal modelling, enhancing latent representation and compression fidelity while improving joint optimisation and spatio-temporal consistency, aligning with our goal of better compression across motion dynamics and content types.

2.1 Proposed Loss-Initialised Generative Adversarial Network with Convolutional Long Short-Term Memory and Cross Fusion Network

This framework outlines an image detection approach using advanced deep learning techniques. As shown in Fig. 1 the process begins with Input Data, which undergoes Data Pre-processing to enhance quality via noise removal (filtering) and value scaling (normalisation). Next, the Feature Extraction module employs cutting-edge architectures to derive high-level representations from the processed data. The models utilised here are Swim Transformer, RAFT (Recurrent All-Pairs Field Transforms), ConvNeXt, FPN (Feature Pyramid Networks), and Vision Transformers.

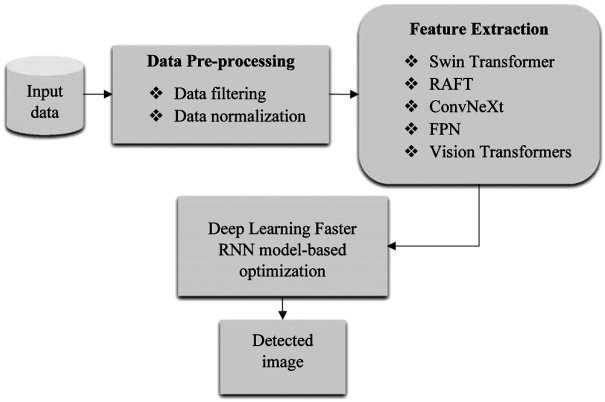

Fig. 1 — Proposed Methodology

Each one of them serves a different purpose. Swim and Vision Transformers offer strong attention mechanisms, RAFT is excellent at flow estimation, ConvNeXt is a vision-specialised convolutional model, and FPN is excellent at multi-scale feature learning. The extracted features are optimised using a deep learning RNN for efficient sequential/ hierarchical learning, enhancing detection accuracy and decision boundaries. The final output is a high-precision detected image with identified objects or regions of interest.

2.1.1 Feature Extraction Techniques:

Feature Extraction leverages Swim Transformer, RAFT, ConvNeXt, FPN, and Vision Transformers to enhance video streaming, surveillance, and AR/VR. These models optimise visual quality and realism across applications.

2.1.2 Swin Transformer:

Swin Transformer splits images into non-overlapping local windows, computing multi-head self-attention (W-MSA) within each. It alternates between regular and shifted window configurations (SW-MSA) across layers to improve cross-window connections while maintaining efficiency. Each Swin Transformer block comprises layer normalisation, window-based MSA (W-MSA or SW-MSA), and a two-layer MLP. This design balances global modelling and computational efficiency. The following is a representation of the sequential Swin Transformer block process:

$$\hat{p} = W - MSA\left(LN\left(p^{i-1} \right) \right) + p^{i-1} \quad \ldots (1)$$

$$p = MLP\left(LN\left(\hat{p}^{i}\right)\right) + \hat{p}^{i} \quad \ldots (2)$$

$$\hat{p}^{i-1} = SW - MSA\left(LN\left(p^{i}\right)\right) + p^{i} \quad \ldots (3)$$

$$p^{i-1} = MLP\left(LN\left(\hat{p}^{i+1}\right)\right) + \hat{p}^{i+1} \quad \ldots (4)$$

where $\hat{p}^{i}$ and p^{i} Represent the output features of the W-MSA and the MLP module, respectively, for block l, $\hat{p}^{i+1}$ and for a block $i+1$, p^{i+1} Stands for the output characteristics of the MLP module and SW-MSA modules' output characteristics, respectively.

2.1.3 Recurrent All-Pairs Field Transform (RAFT)

This architecture uses RAFT to analyse tissue displacement between ultrasound frames, a previously unaddressed challenge. Correlation Volume: Computes inner products of features from two identical CNNs processing pre-/post-displacement frames. Iterative Refinement: Initialises zero displacement and uses a gated recurrent unit to update estimates iteratively. Update Block Input: Combines: Context features (from Frame 1), Correlation features (sampled from volume), Current displacement estimates.

2.1.4 ConvNeXt

The enhanced ConvNeXt architecture modifies stage compute ratios, activation functions, and topology to surpass Swin Transformer and upgrade ResNet. Key innovations: Depth-wise separable convolution reduces parameters/computation vs ResNet's standard convolutions. Inverted bottleneck structure improves accuracy while maintaining efficiency.

2.1.5 Feature Pyramid Network (FPN)

Builds multi-scale feature pyramids using CNNs to detect small objects better while maintaining efficiency. By creating hierarchical features with rich semantics at different scales, FPN improves detection of fine details without significantly increasing computation or memory usage. The pyramid's multiple levels capture both object details and contextual information.

2.1.6 Vision Transformers

Vision Transformers represent the interactions between adjacent and far-off pixels to function according to the attention process.The input picture is initially segmented into tiny portions for attention-based activities. Like a convolutional layer with a kernel, this procedure yields a 4D matrix with batch indexing and three additional dimensions: row, column, and depth.The image $I \in \Re^{h \times w \times c}$ is therefore transformed into $PP \in \Re^{n \times P^2 \times c}$ where h and w Represent the width and height of the picture, and c about the quantity of channels. However, n stands for the number of patches, which is determined as,

$$n = \frac{h \times w}{P^2} \quad \ldots (5)$$

Finally, the above-described features are fused as final feature extraction, which is concatenated, and then the final feature vector is obtained. FE Fed to the model, which is used to classify for further processing.

2.2 Proposed Faster Recurrent Neural Network for Object Enhancement

The Faster R-CNN architecture is the foundation for the proposed two-stage object detection system. The following sections cover the framework's components in more detail, while Fig. 2 provides an overview.

2.2.1 Backbone

ResNet is the foundational feature extractor, using skip connections to avoid vanishing gradients and

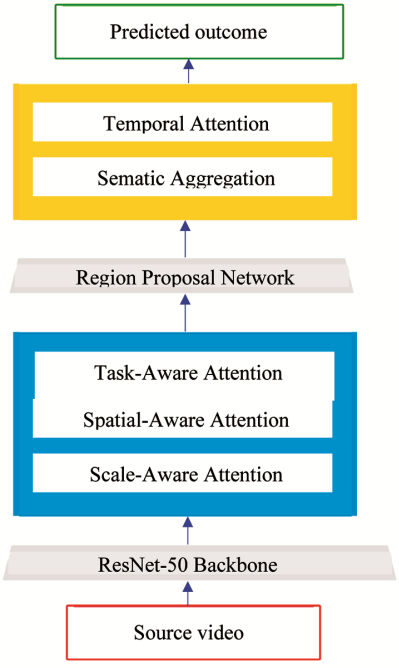

Fig. 2 — The proposed object detection framework

accuracy degradation in deep networks. These residual blocks in Fig. 3 enable efficient feature map generation for subsequent processing modules.

2.2.2 Disentangled Head

Processes multi-level backbone features through sequential attention modules (spatial, scale, and task-aware), illustrated in Fig. 3. Scale-aware: Hard sigmoid output × input tensor, Spatial aware: Applies deform_conv2d to sigmoid outputs with offsets. The starting values $[\alpha 1, \beta 1, \alpha 2, \beta 2] = [1, 0, 0, 0]$ are concatenated with the normalised data in task-aware attention to determine the maximum between the piecewise functions involved $\alpha 1, \beta 1, \alpha 2, \beta 2$, and the tensor. After processing the incoming video frames, they serve as target and support frames in the aggregation head for the disentanglement head. Let $A \in J^{G \times F \times E}$, where $F = I \times M$ Is the height and breadth, and E Is the number of channels, which is a feature tensor from G Levels of a feature pyramid. Equation (6) illustrates how attention is applied generally.

$$Attention = a\left(b\left(c\right), c\right) \quad \ldots (6)$$

Applying the produced attention $b(c)$ to the input c is denoted by the expression $a\left(b\left(c\right), c\right)$. It would be computationally costly to apply the attention function in all dimensions. Instead, the disentanglement network is computationally efficient as it divides into three consecutive attentions used separately and independently, as indicated by Equation (7).

$$M\left(A\right) = \pi_E\left(\pi_F\left(\pi_G\left(A\right).A\right).A\right).A \quad \ldots (7)$$

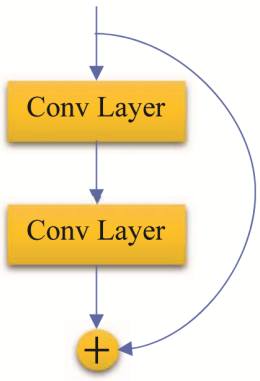

Fig. 3 — A single ResNet residual block

where A represents the backbone network's spatial frame attributes and, π_G, $\pi_G, \pi_F, and\ \pi_E$, denotes the scale, spatial, and channel-wise attention modules, correspondingly. When applied to the level dimension, scale-aware attention π_G Highlights the objects' different scales and dynamically combines the characteristics according to their semantic significance. Applied to the fused characteristics of scale-awareattention, on the dimension F spatially aware attention is usedto concentrate on the positions of the items. Task-aware attention π_E It is applied to the feature channels by dynamically activating and deactivating them to favour specific activities.

2.2.3 Region Proposal Network

The Region Proposal Network (RPN) is a convolutional neural network that can do regression and classification. While the regressor is employed to ascertain the object's coordinates in the proposal, the classifier assesses if the proposal contains the item. The input image is initially covered with anchors of various sizes and scales, and the RPN then assesses if the anchor includes an item. Multiple criteria filter out anchors with low Intersection over Union (IoU); the remaining high-quality anchors go to the following stage for further processing.

2.2.4 Aggregation Head

In this design, the head handles task-specific outputs like a mask or bounding box prediction. We utilise Temporal RoI Align as the ROI extractor and the Sequence-Level Semantics Aggregation (SELSA) module as the Region of Interest (RoI) head for video object recognition. Instead of depending only on nearby frames, the SELSA module aims to leverage characteristics from every frame in a movie efficiently. More robust and descriptive characteristics are produced as a result of this method. To function, SELSA first extracts characteristics from every frame. After that, it calculates how similar these qualities are semantically and groups them according to those similarities. This procedure uses global temporal context to improve object recognition efficiency significantly. On the other hand, Traditional ROI Align merely collects features from the feature map of the current frame, which is devoid of temporal information. This restriction is overcome by Temporal RoI Align, which combines characteristics from many frames. The first step is to extract ROI features from the feature map of the

target frame. From the feature maps of supporting frames, it then determines the K most comparable spots for every proposal. Accordingly, these comparable ROI characteristics are retrieved. Lastly, temporal attention across frames is carried out by Temporal Attentional Feature Aggregation, which is composed of N attention blocks. The model's capacity to reliably identify objects across video frames is further enhanced by this stage, which generates temporally-aware ROI features. These are then forwarded to the optimisation layer for a better optimal solution for accurate detection, as described below.

2.3 Coyote Optimisation and Tuna Swarm Optimisation Algorithm

Meta-heuristic optimisation algorithms have been developed based on natural principles. However, a tiny subset of these procedures is termed swarm intelligence, or SI, and they address the action of a population of heterogeneous users who interact based on pre-programmed rules, such as the group movements of fish, birds, or insects. The information passed between parties and throughout the process makes most SI-based algorithms highly efficient. As a metaheuristic feature extraction technique, this research presents the Swarm Intelligence Metaheuristic of the CT Optimisation Algorithm (SIMCT). Coyote and tuna swarm optimisation have been integrated to show a distinct optimisation strategy for choosing vital variables to boost classification accuracy.

The Tuna Swarm optimisation technique and Coyote optimisation: The coyote'sC_o,positionP_o, and i_{max}there inputs used by SIMCT. Social Condition (soc), which is generated uniformly at random in the search space using a swarm-based metaheuristic, and the initial population of coyotes and tuna, respectively (c and PN), are used by SIMCT to begin optimisation.

$$X_i^{int}=rand.(ub-Ib)+Ib \text{ i=1, 2…….PN} \quad \ldots (8)$$

The top and bottom search area bounds are denoted by u_bPN indicates tuna populations, and rand is a random vector with a range of 0 to 1 evenly distributed. The Coyote then adjusts to similar social circumstances. The packs are randomly allocated to Coyote's fitness calculation using the free parameters x and y.

$$fitness_{soc} = \text{fitness } (X_i^{int}) \quad \ldots (9)$$

Next, using spiral searching, the T_{best} Is determined. When the tuna begins chasing its victim, it occurs. A school of fish uses its senses to move in a specific direction. They impart the information to one another. The spiral that envelops the prey is created as a result. Similarly, the spiral foraging approach formula uses weight to identify the best image characteristics.

$$X_i^{i+1}=\begin{cases}\alpha_1\left(X_{best}^i+\beta.\left|X_{best}^i-X_i^i\right|\right)+\alpha_2.X_i^i, i=1\\ \alpha_1\left(X_{best}^i+\beta.\left|X_{best}^i-X_i^i\right|\right)+\alpha_2.X_i^i, i=2,3\ldots..PN\end{cases} \quad \ldots (10)$$

$$\alpha_1=x+(1-x).\frac{i}{i_{max}} \quad \ldots (11)$$

$$\alpha_{2=(1-x)-(1-x)\cdot}\frac{i}{i_{max}} \quad \ldots (12)$$

The coyote pack's weight coefficients and position, which are α_1, α_2And forward toward prior and ideal individuals, respectively, control individual propensity. Whether the movement space is optimum and unique depends on the constant x in the first phase. Y stands for a randomly distributed random integer between 0 and 1. The letters i and i_{max}represent iteration and maximum iteration, respectively, as is customary. Consequently, t_best is calculated to be the current optimal feature with the best solution. Furthermore, it is possible that the characteristics of the perfect person may not be discovered. We call the random search space A_rand^i at this point. Consequently, the following equation shows that the ability to search globally is improved.

$$X_i^{i+1}=\begin{cases}\alpha_1\left(X_{rand}^i+\beta.\left|X_{rand}^i-X_i^i\right|\right)+\alpha_2.X_i^i, i=1\\ \alpha_1\left(X_{rand}^i+\beta.\left|X_{rand}^i-X_i^i\right|\right)+\alpha_2.X_i^i, i=1,2\ldots PN\end{cases} \quad \ldots (13)$$

After that, the population is examined to see whether one repetition equals one. The quantity, position, and pack of coyotes are then updated by comparing the free parameter of y with a random value. The new findings were different, as the comparison process showed. If the condition (rand>z) is satisfied, the initialisation process changes the position. The descriptor rand<0.5 has been added to the coyote pack. By meeting the criterion ($\frac{i}{i_{max}}$), The

population is updated. The following is how the mathematical model represented this update.

$$X_i^{i+1}=\begin{cases}\alpha_1\left(X_{rand}^i+\beta.\left|X_{rand}^i-X_i^i\right|\right)+\alpha_2.A_i^i, i=1\\ \alpha_1\left(X_{rand}^i+\beta.\left|X_{rand}^i-X_i^i\right|\right)+\alpha_2.A_i^i, i=1,2\ldots PN\end{cases} \text{ if rand} < \frac{i}{i_{max}} \quad \ldots (14)$$

$$X_i^{i+1}=\begin{cases}\alpha_1\left(X_{best}^i+\beta.\left|X_{best}^i-X_i^i\right|\right)+\alpha_2.X_i^i, i=1\\ \alpha_1\left(X_{best}^i+\beta.\left|X_{best}^i-X_i^i\right|\right)+\alpha_2.X_i^i, i=2,3\ldots PN\end{cases} \text{ if rand} \geq \frac{i}{i_{max}} \quad \ldots (15)$$

$$X_i^{i+1}=\begin{cases}X_{best}^i+rand.\left(X_{best}^i-X_i^i\right)+TF.P^2.\left(X_{best}^i-X_i^i\right) \text{ if rand} < 0.5\\ TF.P^2.X_{i,}^i \text{ if rand} \geq 0.5\end{cases} \quad \ldots (16)$$

Following the social condition-based updating of the coyote pack, the death and birth process may be used to investigate the age-related mortality risk for coyotes. Similarly, the missing components of DR are identified using the estimated coyote age technique.

$$\text{Po}=\begin{cases}soc_{r1} \; rand < PR_s\\ soc_{r2} \; rand \geq PR_s, PR_a\\ otherwise\end{cases} \quad \ldots (17)$$

After the lost feature-finding procedure is finished, a second fitness calculation is carried out to improve the quality of the feature finding. The population of coyotes, the coyote package, and the population definitions were then updated due to their evolving social situations. Based on social conditions, this new update is performed using the alpha and coyote packages as shown in the equation below. Meanwhile, a new fitness value was also found as

$$\text{Best update= soc} + \alpha_1.x + \alpha_2.y \quad \ldots (18)$$

$$\text{Best fit =f (new update)} \quad \ldots (19)$$

Finally, the population is evaluated to confirm that $i > i_{max}$And meet the conditions. This condition has been satisfied; the fitness value and optimal feature are obtained, or the loop function is started by increasing the iteration value. The flow chart may be seen in the above Fig. 4. The SIMCT was implemented in the study to improve regions with smooth content and rich textures to obtain the optimal solution.

3 Results And Discussion

This section displays the outcomes of the FRNN-SIMCT model, performance analysis and a comparison with other models to demonstrate its effectiveness.

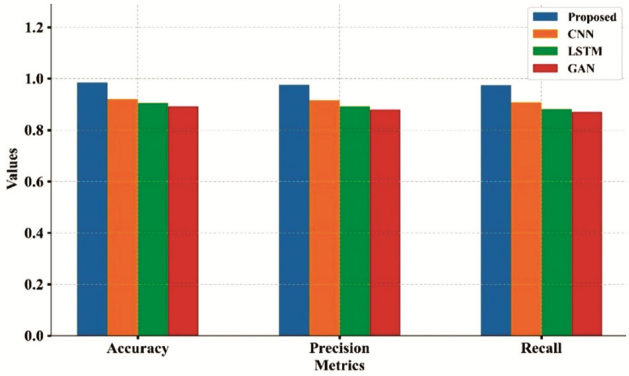

Fig. 4 — Comparative analysis of accuracy, precision, and recall metrics

3.1 Dataset Description

Seven sophisticated, stationary cameras with overlapping fields of vision were used to collect the dataset. In particular, four GoPro Hero 3 and three GoPro Hero 4 cameras were utilised. The data was collected in pleasant weather in front of ETH Zurich's main building in Switzerland. The scenes were captured at 60 frames per second and have a 1920×1080pixel quality. Their areas of vision mostly overlap due to the camera arrangement. As is evident, the camera placements are higher than the average height of people[33].

3.2 Evaluation Metrics

3.2.1 Accuracy: The ratio of properly categorised samples to all samples in the assessment dataset is accurate. This is mathematically represented in equation (20),

$$Acc=\frac{\text{TP}^+ + \text{TN}^-}{\text{TP}^+ + \text{TN}^- + F\text{P}^+ + F\text{N}^-} \quad \ldots (20)$$

3.2.2 Precision: Precision is computed by dividing the total number of retrieved instances by the number of accurate cases.

$$pre=\frac{\text{TP}^+}{\text{TP}^+ + F\text{P}^+} \quad (21)$$

Here pre is denoted as precision, TP^+ is termed as a true positive, and $F\text{P}^+$ It is denoted as a false positive.

3.2.3 F1-score: The F1-score is also known as the F-measure. The precision and recall weights in the F1 score can vary depending on the index.

$$Fscr = 2 \times \frac{pre \times rec}{pre + rec} \quad (22)$$

Here $Fscr$ stands for the F1-score, pre referred to as precision, and rec itis stated as a recall.

3.2.4 Recall: The number of successfully retrieved instances divided by the total number of correctly recovered instances yields recall.

$$rec = \frac{\mathrm{TP}^{+}}{\mathrm{TP}^{+} + F\mathrm{N}^{-}} \quad \ldots (23)$$

Here rec the recall TP^{+} is represented as a true positive and $F\mathrm{P}^{+}$ It is indicated as a false positive.

3.3 Comparative Analysis

The performance analysis of FRNN-SIMCT is evaluated using the dataset by varying the training percentage, and the results are displayed as follows:

3.3.1 Comparative analysis of accuracy, precision, and recall metrics

The performance of four models, CNN, LSTM, GAN, and the suggested FRNN-SIMCT, is compared in Fig. 5 using three performance metrics: accuracy, precision, and recall. The FRNN-SIMCT model performs better, with a precision of 0.98, a recall of 0.97, and an F1 score of 0.98. All measures show that it outperforms the other models. With an accuracy of 0.92, precision of 0.91, and recall of 0.91, the CNN model comes in second place, while the LSTM model has the same results: accuracy of 0.91, precision of 0.89, and recall of 0.89. The GAN model's accuracy, precision, and recall are 0.90, 0.88, and 0.87, respectively, the lowest of all the models. The research shows the efficacy of the proposed FRNN-SIMCT model in yielding more accurate, precise, and reliable results, validating its superiority in classification tasks and making it a more dependable option among the models tested.

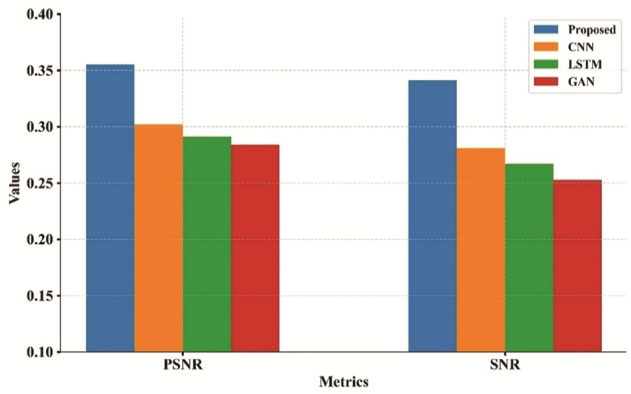

Fig. 5 — Comparative analysis of PSNR and SNR

3.3.2 Comparative analysis of PSNR and SNR

Based on PSNR (Peak Signal-to-Noise Ratio) and SNR (Signal-to-Noise Ratio), Fig. 6 shows a comparison of the performance of four models: CNN, LSTM, GAN, and the proposed FRNN-SIMCT. The FRNN-SIMCT model has a PSNR value of 0.36 and an SNR value of 0.34, which is visibly better than the other models. As against this, the LSTM model is inferior with PSNR: 0.29 and SNR: 0.27 and the CNN model has PSNR: 0.30 and SNR: 0.28. The minimum values of 0.28 PSNR and 0.25 SNR are the lowest generated by the GAN model. These results reflect the higher Denoising and Signal Preservation skills of the FRN-SIMCT approach. The improved quality in image reconstruction is indicated by increased PSNR and SNR values, indicating the robustness and superiority of the proposed model compared to traditional deep learning methods for signal quality improvement applications.

3.3.3 Comparative analysis of F-measure and compression Ratio

A global performance analysis of the four models, CNN, LSTM, GAN, and the proposed FRNN-SIMCT, based on F-measure and Compression Ratio, is presented in Fig. 7. An F-measure of 0.97 and a Compression Ratio of 0.30 indicate that the FRNN-SIMCT model performs exceptionally well, reflecting its accuracy and data compression effectiveness. Second, the CNN model has an F-measure of 0.91 and

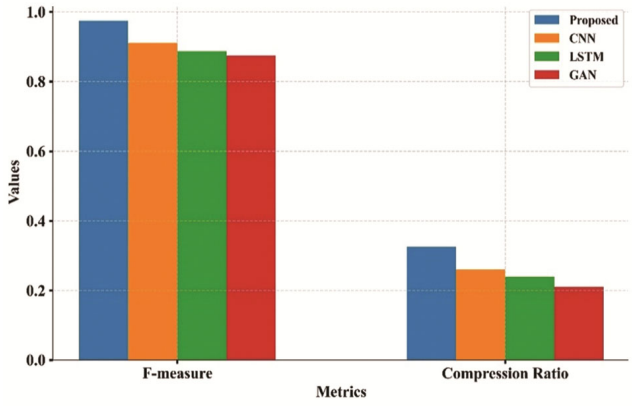

Fig. 6 — Comparative analysis of F-measure and compression Ratio

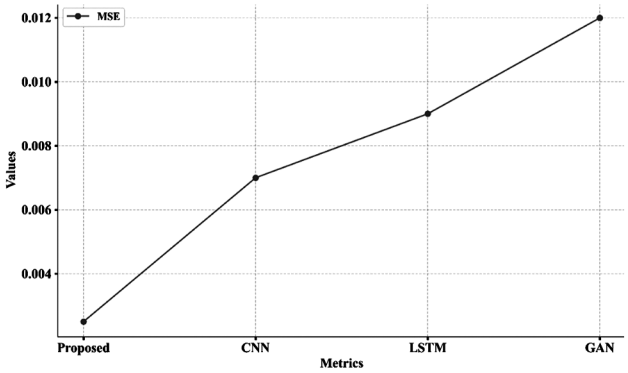

Fig. 7 — Comparative analysis of Mean Squared Error

a Compression Ratio of 0.26, while the LSTM model has an F-measure of 0.89 and a Compression Ratio of 0.24, ranking third. Last is the GAN model with an F-measure of 0.88 and a Compression Ratio 0.21. These findings imply that the FRNN-SIMCT version is an excellent choice where storage space performance and accuracy are most important, because it performs adequately in classification and is better in terms of compression performance.

3.3.4 Comparative Analysis of Mean Squared Error

The MSE of the suggested models (FRN-SIMCT), CNN, LSTM, and GAN are compared and shown in Fig. 7. With an MSE of 0.0025, the proposed model has the smallest error, showing its excellent performance in reconstruction and forecasting tasks. The LSTM model has a higher error at 0.0090, while the CNN model is second with a 0.0070 MSE. The GAN model has the poorest accuracy among the four models, with the highest MSE value of 0.0120. The results distinctly indicate the efficiency of the proposed FRN-SIMCT model since the smaller MSE values represent optimal model performance. It performs much better than conventional deep learning methods and is appropriate for those applications that demand minimal error.

4 Conclusion

The suggested F-RNN-SIMCT successfully surpasses vital visual degradation and movement distortion problems in severely compressed images and videos. Using a rapid, recurring neural network and natural behaviour intelligence algorithms, the method significantly improves the accuracy of the movement estimate and the preservation of details, particularly under high-moving conditions. This hybrid optimisation model enables the system to be dynamically tuned to varied forms of content, enhancing the overall perceptual quality of compressed visual information. Results of experiments affirm that F-RNN-SIMCT achieves better performance than traditional approaches regarding loyalty and coherence, rendering a sustainable solution for applications requiring efficient compression without compromising visual details. Moreover, the methodology's capability to accelerate the motion estimation process without compromising accuracy makes it ideal for real-time processing in bandwidth-limited environments. This work provides valuable insights into integrating deep learning and bio-inspired optimisation methods for improving compression performance. Future research can delve deeper into further optimisation of the algorithm, scalability for higher-resolution videos, and application to real-world video coding standards to maximise bandwidth usage and viewer satisfaction in multimedia transmission and storage systems.

References

1 Mohamed H, Elliethy A, Amr A & Aly H, *Multimedia Tools Appl*, 2024 (2024) 1.
2 Ahmad W, Mahdavi H & Hamzaoglu I, *J Real Time Image Process*, 21 (2) (2024) 25.
3 Mishra A & Kohli N, *Int J Comput Sci Eng*, 27(2) (2024) 133.
4 Sheng X, Li L, Liu D & Li H, "Spatial decomposition and temporal fusion based inter prediction for learned video compression," *IEEE Trans Circuits Syst Video Technol*, 2024.
5 Ghoul K, Zaidi S & Laboudi Z, *Adv Electr Comput Eng*, 24 (1) (2024) 33.
6 Hu *et al*., "HDVC: Deep video compression with hyperprior-based entropy coding," *IEEE Access*, 2024.
7 Argaw D M, Kim J & Kweon H, "Blurry video compression: A trade-off between visual enhancement and data compression," in *Proc IEEE/CVF Winter Conf Appl Comput Vis*, P. 4280– 4290 2024.
8 Xu Y, Lu L, Saragadam V & Kelly K F, *Nat Commun*, 15 (1) (2024) 1456.
9 Zhang *et al*., *ACM Comput Surv*, 55 (12) (2023) 1.
10 Yang *et al*, "Video coding for machines: Compact visual representation compression for intelligent collaborative analytics," *IEEE Trans Pattern Anal Mach Intell*, 2024.
11 Anatharaman *et al*., *Theor Comput Fluid Dyn*, 37 (1) (2023) 61.
12 Wang Y, ChanH P & DonzellaV, "Semantic-aware video compression for automotive cameras," *IEEE Trans Intell Veh*, 8 (6) (2023) 3712.
13 Gomes C, Azevedo R, & Schroers C, "Video compression with entropy-constrained neural representations," in *Proc IEEE/CVF Conf Comput Vis Pattern Recognit*, (2023) 18497.
14 Khan M U K, Chadha A, Anam M A, & Andreopoulos Y Perceptual Video Compression with Neural Wrapping. *In Proceedings of the Computer Vision and Pattern Recognition Conference* (2025) P. 17743-17754.

15 Mersha M, Lam K, Wood J, Alshami A K & Kalita J, *Neurocomputing*, 599 (2024) 128111.
16 Wang *et al*., "Compression-aware video super-resolution," in *Proc IEEE/CVF Conf Comput Vis Pattern Recognit*, (2023) P. 2012–2021.
17 Yang R, TINC: Tree-structured implicit neural compression. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023) P. 18517-18526.
18 Ding *et al*., "Advances in video compression system using deep neural network: A review and case studies," *Proc IEEE*, 109 (9) (2021) 1494.
19 Cheng Z, Sun H, Takeuchi M & Katto J, Learned image compression with discretized gaussian mixture likelihoods and attention modules. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020) P. 7939-7948.
20 Li J, Li B & Lu Y, "Deep contextual video compression," *Adv Neural Inf Process Syst*, 34 (2021) 18114.
21 Mertens A, Nicolas G & Rovira, S, Convolution-Friendly Image Compression with FHE. In International Conference on Cryptology in Africa, P. 3-24July 2025.
22 Hu *et al*., "Improving deep video compression by resolution-adaptive flow coding," in *Comput Vis*, (Springer), 16 (2020) 193.
23 Jin *et al*., "Enhanced bi-directional motion estimation for video frame interpolation," in *Proc IEEE/CVF Winter Conf Appl Comput Vis*, (2023) 5049.
24 Han *et al*., "Deep generative video compression," in *Proc 33rd Int Conf Neural Inf Process Syst*,. (2019) P. 9287.
25 Rai S, Shrivastava A & Nigam R, *Int J Comput Appl*, 975 (2019) 8887.
26 Amoolya M, Amrutha B P, AmbikaY N, Patil A R & Thirumagal E, *J Adv Zool*, 44 (2023) 593.
27 Mochurad L, *Technologies*, 12 (4) (2024) 52.
28 Kulsoom *et al*., *Neural Comput Appl*, 34 (21) (2022) 18289.
29 Tang *et al*., "Scene matters: Model-based deep video compression," in *Proc IEEE/CVF Int Conf Comput Vis*, (2023) 12481.
30 Latha H R & Prasath A R, "ICPCH: A hybrid approach for lossless DICOM image compression using combined approach of linear predictive coding and Huffman coding with wavelets," in *Int Conf Cognition Recognit*, (Springer), (2021) 269.
31 Habibian A, van Rozendaal T, Tomczak J M & Cohen T S, "Video compression with rate-distortion autoencoders," in *Proc IEEE/CVF Int Conf Compu Vis*, p. 7033–7042, 2019.
32 Amosa T I, Sebastian P, Izhar L I, Ibrahim O, Ayinla L S, Bahashwan A A & Samaila Y A, *Neurocomputing*, 552 (2023) 126558.