European Journal of Artificial Intelligence and Machine Learning
Vol 5 | Issue 2 | April 2026
ISSN 2796-0072

RESEARCH ARTICLE

Adapted Deep Key Generation Using Fourier–Riesz Features for Secure Video Encryption

Haingonirina Ignace Rajaosolomanantena*, Toky Basilide Ravaliminoarimalalason, and Hery Zo Andriamanohisoa

ABSTRACT

Video encryption protects multimedia data over insecure networks. This paper introduces a hybrid key-generation framework combining Fourier–Riesz features with an adapted deep neural model to produce dynamic, frame-dependent keys. A four-channel representation integrating spectral magnitude, spectral phase, directional amplitude, and orientation ensures key decorrelation. Experiments conducted on standard video datasets showed entropy values ranging between 7.96 and 7.99 bits, a strong avalanche effect with an average Hamming distance of 129.62, near-zero inter-frame and inter-channel correlations, and preserved visual quality with a PSNR of 42 dB. Security analysis confirmed overall robustness through extensive evaluations.

Keywords: Deep learning, fourier transform, key generation, riesz transform.

Submitted: December 26, 2025

Published: April 16, 2026

10.24018/ejai.2026.5.2.70120

Ecole Doctorale en Sciences et Techniques de l'Ingénierie et de l'Innovation (ED STII), Ecole Supérieure Polytechnique, Laboratory of Cognitive Sciences and Applications, University of Antananarivo, Madagascar.

Corresponding Author:
e-mail: rhignace@gmail.com

1. Introduction

The rapid growth of video-based applications has made multimedia security a critical challenge, as traditional encryption algorithms such as AES, DES, and RSA remain computationally costly for large-scale video data [1], [2]. Faster alternatives, including selective and chaos-based encryption, improve efficiency but often suffer from reduced robustness and exploitable vulnerabilities [3], [4]. To overcome these limitations, adaptive deep learning–based key generation has emerged as an effective solution by exploiting content-dependent characteristics [5].

This work proposes a hybrid video encryption framework that integrates Fourier and Riesz transforms within a deep neural architecture to capture both spectral and directional information [6]. A spectro-directional tensor is processed by an orthogonally constrained network, ensuring key stability, decorrelation, and numerical robustness [7]. Hybrid activation, adaptive training, and Jacobian control enable high entropy, strong avalanche effects, and inter-frame independence. Extensive experiments confirm the effectiveness of the proposed framework in achieving secure, robust, and high-quality video encryption.

2. Literature Review

Recent studies show that deep learning is increasingly used for secret key generation from biometric data. Symmetric keys have been generated from fingerprint images using a VGG-16 network [8], while multimodal biometric fusion combining face and finger-vein features with FaceNet, VGG19, and siamese architectures has been used to derive stable keys [9]. Post-quantum compatible keys based on facial CNNs and code-based extractors were proposed in [10], and high-entropy fingerprint-based keys using CNNs with Particle Swarm Optimization were introduced in [11]. In parallel, encryption keys derived from trinion Fourier transforms driven by chaotic systems were presented in [12], without deep learning or temporal adaptation. Unlike these static approaches, the present work focuses on dynamic video sequences using a temporally adaptive Fourier–Riesz deep model for content-dependent key generation.

3. Materials and Methods

3.1. Video Datasets

For experimental evaluation, the Akiyo sequence from the Xiph.org Video Test Media Repository, a standard

YUV video collection, was used as a reference, comprising 300 frames [13]. Each frame, representing both static and dynamic scenes, was extracted and resized to 128 × 128 pixels before feature extraction.

3.2. Hardware and Software Environment

Experiments were conducted on a Windows 10 platform using Python 3.10 with TensorFlow/Keras. Training was performed on a system equipped with an 8-core CPU, 16 GB RAM, and an NVIDIA GPU with 8 GB VRAM, enabling efficient tensor processing and accelerated optimization.

3.3. Feature Extraction

In the proposed pipeline, four complementary features are extracted from each video frame in order to build a compact yet expressive spectro-directional representation. More precisely, we derive a spectral magnitude map M_t, a spectral phase component Φ_t, a Riesz-based directional response R_t, and a temporal variation map Θ_t. These extracted features form a compact structure that enables the generation of content-dependent keys.

3.4. Network Architecture

3.4.1. Adapted Dense Layer

We define a tensor-dependent orthogonal projection based on $\overline{\Psi}_t$:

$$y = \Omega\left(\overline{\psi}_t\right) \cdot x + b, \Omega\left(\overline{\psi}_t\right) \in O(n) \tag{1}$$

where $\Omega\left(\overline{\psi}_t\right)$ is an orthogonal matrix dependent on the tensor $\overline{\psi}_t$ belonging to the set $O(n)$ of orthogonal matrices of dimension n and b is a bias term.

Here, the matrix $\Omega\left(\overline{\psi}_t\right)$ is regularized to remain orthogonal:

$$\Omega\left(\overline{\psi}_t\right)^T \cdot \Omega\left(\overline{\psi}_t\right) = I \tag{2}$$

3.4.2. Activation Function

The proposed activation function is not fixed but depends on the spectral-directional features:

$$\varphi_{Hybride}(x) = \alpha(\overline{\psi}_t) \cdot ReLU(x) + \left(1 - \alpha(\overline{\psi}_t)\right) \cdot \sigma(x) \tag{3}$$

where $ReLU(x)$ denotes the Rectified linear function, $\sigma(x)$ denotes the sigmoid function, and $\alpha(\overline{\psi}_t) \in (0, 1)$ is a coefficient dynamically computed from $\overline{\psi}_t$.

3.4.3. Jacobian

The input-output Jacobian, as mentioned in [14], factorizes layer by layer:

$$J_t = \frac{\partial K_t}{\partial \overline{\psi}_t} = \prod_{l=1}^{L} \left(\Omega\left(\overline{\psi}_t\right) . Diag(\varphi'_{hybride}(z_l))\right) \tag{4}$$

where K_t denotes the encryption key at frame t, $\overline{\psi}_t$ is the normalized spectral-directional tensor, $\Omega\left(\overline{\psi}_t\right)$ is the weight matrix of layer l, $\varphi'_{hybride}(z_l)$ is the derivative of the hybrid activation function and z_l is the pre-activation at layer l.

$$z_l = \Omega\left(\overline{\psi}_t\right) h_{l-1} + b_l \tag{5}$$

where $\Omega\left(\overline{\psi}_t\right)$ is an adaptive orthogonal matrix dependent on the tensor $\overline{\psi}_t$, h_{l-1} represents the outputs of the previous layer, and b_l is the bias vector of layer l.

Owing to the bounded nature of the hybrid activation derivative, we impose:

$$0 < S_l \leq \varphi'_{hybride}(Z_l) \leq \overline{S}_l \leq 1 \tag{6}$$

Ensuring that the Frobenius norm $\|J_t\|_F$ remains controlled. The lower bound S_l prevents degenerate mapping with vanishing Jacobian norm, while the upper bound $\overline{S}_l \leq 1$ avoids gradient explosion.

3.5. Training Procedure

3.5.1. Training Hyperparameters and Configuration

The network was trained using gradient descent with an adaptive learning rate initialized at $\eta_0 = 10^{-3}$ and dynamically modulated according to the spectral energy of the input tensor. The number of epochs was set to E = 50, as the proposed orthogonally constrained and Jacobian-regularized architecture exhibited rapid convergence. No mini-batching was used, as training was performed sequentially frame by frame. The composite loss weights were empirically fixed as follows: orthogonality penalty $\lambda_1 = 0.1$, inter-frame decorrelation $\lambda_2 = 0.5$, and Jacobian margin constraint $\lambda_3 = 0.2$.

3.5.2. Frame-Wise Local and Adaptive Learning

For each frame t, the network parameters θ_t are locally updated as:

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta \mathcal{L}_{unsup}\left(f_t, N_\theta(\overline{\psi}_t)\right) \tag{7}$$

where N_θ denotes the deep neural network parameterized by θ, applied to the normalized tensor $\overline{\psi}_t$ denotes the normalized input tensor at time t; f_t denotes the frame at time t; $\mathcal{L}_{unsup}(\cdot,\cdot)$ denotes the unsupervised loss function; ∇_θ denotes the gradient of the loss with respect to the parameters θ; and η_t denotes the learning rate at iteration t.

The learning rate η_t was itself modulated by the spectral energy of the tensor:

$$\eta_t = h\left(E_{spec}(\overline{\psi}_t)\right) \tag{8}$$

where $\overline{\psi}_t$ denotes the normalized input tensor at time t, and $E_{spec}(\overline{\psi}_t)$ denotes the spectral energy of $\overline{\psi}_t$.

As a result, the update becomes self-adaptive: each frame adjusts the learning rate according to its spectral content.

3.5.3. Loss Function

The unsupervised objective $\mathcal{L}_{unsup}$ enforces orthogonality, temporal decorrelation, and minimum sensitivity.

In accordance with (1), the dense layer is parameterised by a square orthogonal matrix conditioned on the spectral–directional tensor, denoted $\Omega\left(\overline{\psi}_t\right)$. After each gradient update, we enforce orthogonality by reprojecting the updated matrix onto the orthogonal manifold using a QR factorisation and retaining the Q factor, such that:

$$\Omega\left(\overline{\psi}_t\right)^T \cdot \Omega\left(\overline{\psi}_t\right) = I \tag{9}$$

To further stabilise optimisation, we add a soft orthogonality penalty:

$$L_{orth} = \left\| \Omega\left(\overline{\psi}_t\right)^T \cdot \Omega\left(\overline{\psi}_t\right) - I \right\|_F^2 \quad (10)$$

To enforce inter-frame independence, we minimize the squared Pearson correlation between successive keys:

$$L_{div} = corr(K_t, K_{t+1})^2 \quad (11)$$

where corr(·,·) is the Pearson correlation on vectorized keys, averaged over the mini-batch.

Finally, we enforce a minimum Jacobian norm to promote the avalanche effect:

$$L_{jac} = \max(0, \varepsilon - \|J_t\|_F)^2 \quad (12)$$

where $J_t = \dfrac{\partial K_t}{\partial \bar{\psi}_t}$ denotes the Jacobian of the key with respect to the normalised input tensor.

The overall loss is then given by the weighted sum:

$$\mathcal{L}_{unsup} = \lambda_{orth} L_{orth} + \lambda_{div} L_{div} + \lambda_{jac} L_{jac} \quad (13)$$

3.6. Key Generation

The normalized tensor $\overline{\psi}_t \epsilon \mathbb{R}^{M\times N\times 4}$ is used as input to an adapted designed deep neural network $\mathcal{N}_\theta$ with the following configuration.

The network takes $\overline{\psi}_t$ as input and outputs an encryption key in three channels Red, Green and Blue (RGB):

$$K_t = \mathcal{N}_\theta\left(\overline{\psi}_t\right) \epsilon\, [0,1]^{M\times N\times 3} \quad (14)$$

where $\mathcal{N}_\theta$ denotes the deep neural network parameterized by θ applied to the normalized tensor $\overline{\psi}_t$, and $[0,1]^{M\times N\times 3}$ represents the three-dimensional real space of dimensions $M \times N \times 3$.

The network generates a continuous RGB encryption key that is scaled to an 8-bit integer array; therefore, the channel index $c \in \{R, G, B\}$ is introduced in the following formula:

$$K_t^{'}(x, y, c) = \lfloor 225 \cdot K_t(x, y, c) \rfloor \epsilon \mathbb{Z}_{256} \quad (15)$$

for c $\epsilon\{R, G, B\}$

where $K_t(x, y, c)$ represents the continuous encryption key generated by the deep neural network, c denotes the R, G or B channel, and $\mathbb{Z}_{256}$ denotes the set of integers from 0 to 255.

3.7. Video Encryption

The XOR-based encryption is then performed as:

$$f_t^{(c)}(\mathrm{x}, \mathrm{y}, \mathrm{c}) = f_t^{(RGB)}(\mathrm{x}, \mathrm{y}, \mathrm{c}) \oplus K_t^{'}(x, y, c) \quad (16)$$

$$\forall \mathrm{c} \epsilon \{R, G, B\}$$

where $f_t^{(RGB)}$(x, y, c) represents the original pixel for channel c, and $\oplus$ denotes the XOR encryption operation with the key $K_t^{'}$, with c denoting the R, G, or B channel.

3.8. Video Decryption

As mentioned in [15] regarding XOR properties, if a pixel f has been encrypted with a key k, the original can be recovered by:

$$\widehat{f_t}(x, y, c) = f_t^{(c)}(x, y, c) \oplus K_t^{'}(x, y, c) \quad (17)$$

where $f_t^{(c)}$(x, y, c) denotes the encrypted pixel at time t and channel c, $K_t^{'}(x, y, c)$ refers to the key, identical to the one used during encryption, and $\widehat{f_t}(x, y, c)$;represents the decrypted pixel [16].

3.9. Algorithm

```
Begin
    For each frame f_t in the video
sequence do:
        # Preprocessing
        f_gray ← Convert(f_t, RGB → Gray)
        # Feature extraction
        (Mag, Phase) ← FFT2D(f_gray)
        (Amp, Orient)← RieszTransform
          (f_gray, order= 1)
        # Feature tensor construction
        ψ_t ← Combine(Mag, Phase, Amp,
Orient)   # dimension M × N × 4
        ψ̄_t ← Normalize(ψ_t, [0, 1])
        # Key generation via adapted
Deep Learning
        K'_t ← DeepNetwork(ψ̄_t)   # RGB
output, 8 bits
        # Encryption step
        encrypted ← XOR(f_t, K'_t)
        Save(encrypted)
        # Decryption step (validation
of reversibility)
        decrypted ← XOR(encrypted, K'_t)
        Save(decrypted)
    EndFor
End
```

3.10. Threat Model and Security Properties

Security was evaluated under standard threat models, including Ciphertext-Only Attack, Known-Plaintext Attack, and Chosen-Plaintext Attack, in accordance with Kerckhoffs' principle. The spectro-directional deep key generator produced frame-wise, content-dependent keys, preventing key reuse and minimizing temporal correlations. Adaptive learning and hybrid activation introduced strong nonlinearity, while Jacobian-constrained training and orthogonality ensured high entropy, avalanche effect, and statistical independence. Consequently, the framework provided robust security despite the use of XOR-based encryption.

4. Results

This section presents results on key quality and their effect on the security and robustness of 3D data encryption [17], [18].

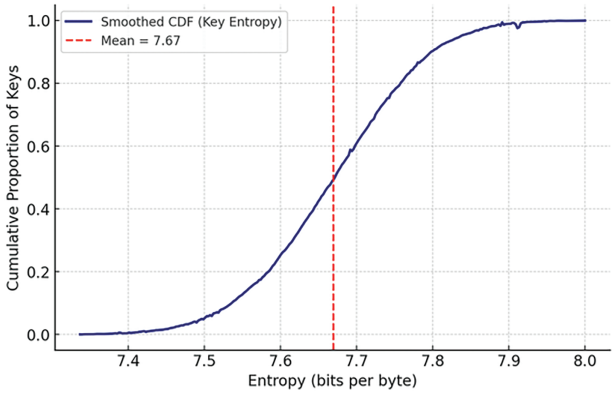

Fig. 1. Cumulative distribution function of key entropy.

4.1. *Analysis and Evaluation of Generated Keys*

Key quality and security were assessed using metrics for randomness, uniqueness, and robustness [19].

4.1.1. *Key Entropy*

Fig. 1 shows the cumulative distribution of key entropy, illustrating the keys' uniformity.

The generated keys exhibited a mean entropy of 7.67 bits per byte, with a 95% confidence interval that remained above weak-randomness thresholds, thereby confirming strong statistical randomness and cryptographic suitability [20].

4.1.2. *Avalanche Effect*

Fig. 2 shows the avalanche effect, where small input changes greatly alter the generated key [21].

The mean Hamming distance of 129.62 bits with a 95% confidence interval from 129.11 to 130.13 confirmed a balanced avalanche effect, while a one-sample t-test against 128 bits yielded $p < 0.001$, and the range 116–143 bits demonstrated strong diffusion and resistance to differential attacks [22], [23].

4.1.3. *Inter-Frame and Inter-Channel Independence*

The following analysis, as Fig. 3 illustrates, evaluates the independence of keys across frames and color channels to ensure high variability and prevent redundancy [24].

Inter-frame correlations averaged 0.002 over 300 frames with a 95% confidence interval from −0.013 to 0.017 and a p-value of 0.77, consistent with [25], confirming strong temporal independence between successive keys [26].

Fig. 4 shows the correlations between the R, G, and B channels of the generated keys, which are near zero, ranging from minus 0.04 to 0.01, indicating minimal redundancy and strong statistical independence.

According to [27], as shown in Table I, the inter-channel correlations were weak, with averages of 0.0129 for RG, −0.045 for RB, and 0.0129 for GB over 20 frames, corresponding to 95% confidence intervals of [−0.0648, 0.0906], [−0.122, 0.032], and [−0.0648, 0.0906], and p-values of 0.73, 0.23, and 0.73, respectively. These results confirm statistical independence and strong, non-redundant key variability [28].

4.2. *Evaluation of Video Encryption and Decryption Performance*

4.2.1. *Video Encryption and Decryption Results*

Original and decrypted frames are visually compared in Fig. 5 to assess encryption fidelity [29].

The Akiyo sequence (a) is unreadable after encryption (b) and fully restored after decryption (c), showing the effectiveness of the key generation method [30].

4.2.2. *Correlation between Adjacent Pixels*

In accordance with [31], the proposed encryption, as depicted in Fig. 6, significantly reduced adjacent-pixel correlation to a negligible level, confirming key effectiveness.

The original videos exhibited a pixel correlation of approximately 0.9, which dropped close to zero after encryption, confirming the effective removal of spatial redundancies.

4.2.3. *Validation of Key Effectiveness via Entropy and Directional Correlations*

Tables II and III presents the entropy and correlations before and after encryption to validate the effectiveness of the generated keys.

According to Ghouate [30], entropy near 8 bits ensures strong randomness, and our encrypted video reached

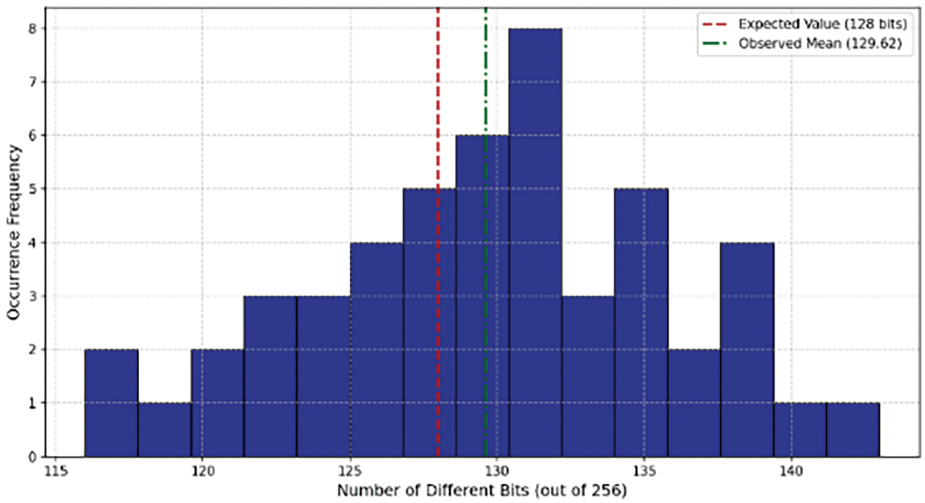

Fig. 2. Distribution of the avalanche effect (Hamming distances between 256-bit keys).

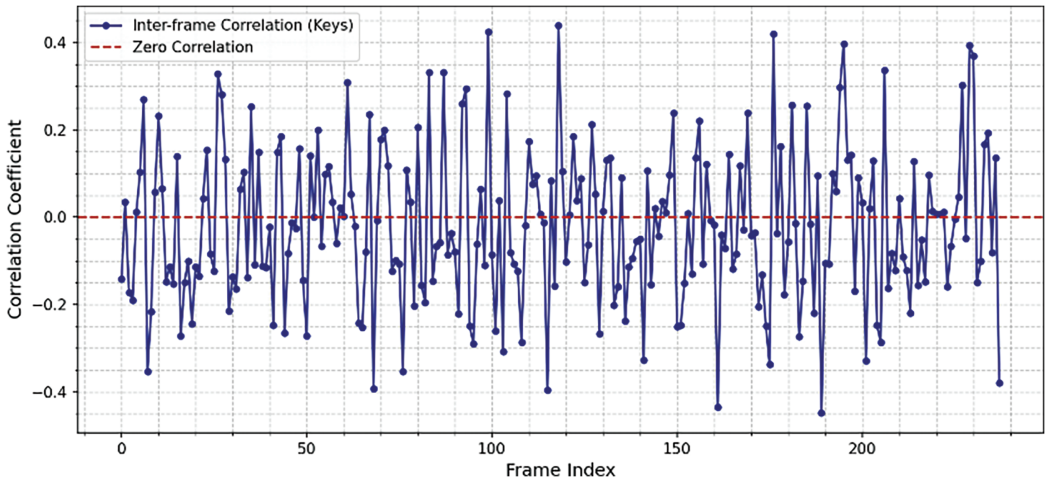

Fig. 3. Key independence between successive frames.

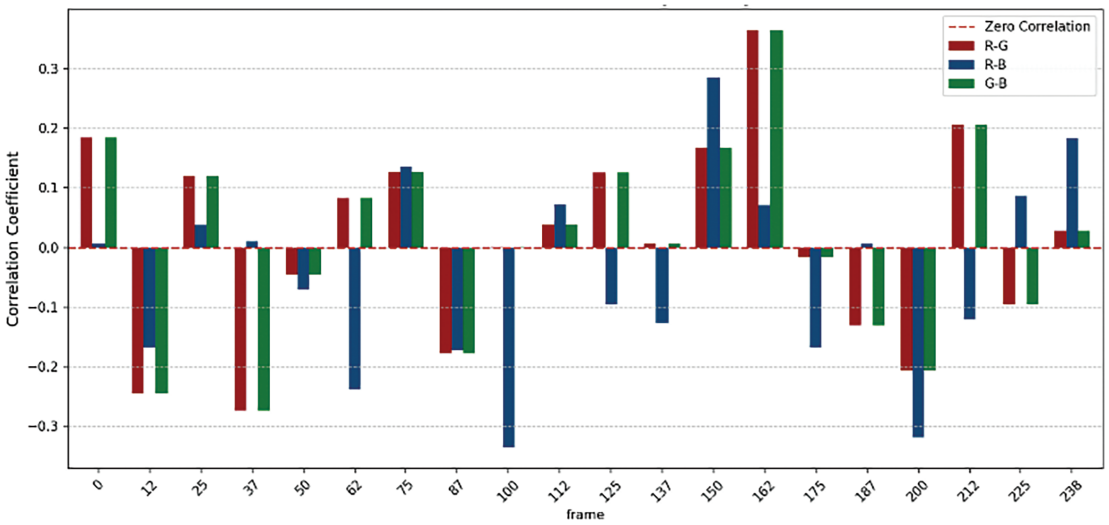

Fig. 4. Key independence across color channels (R, G, B).

TABLE I: Descriptive Statistics of Inter-Channel Correlations (R–G, R–B, G–B) over 20 Frames

	Frame	R-G	R-B	G-B
N_f	20	20	20	20
Mean	118.55	0.0129	−0.045	0.0129
Std	74.013	0.166	0.164	0.166
Min	0.00	−0.273	−0.335	−0.273
Q_1	59.00	−0.103	−0.167	−0.103
Q_2	118.50	0.0175	−0.032	0.017
Q_3	178.00	0.126	0.071	0.126
Max	238.00	0.363	0.285	0.363

between 7.96 and 7.99 bits with directional correlations averaging 0.006 over 5 frames with a 95% confidence interval from −0.008 to 0.020, demonstrating effective key generation.

4.2.4. Evaluation of System Robustness against Disturbances

Robustness was tested via PSNR after decrypting videos affected by noise, compression, or data loss.

Our keys provided robust encryption, Table IV highlights that we achieved a PSNR of 33.8 dB over 300 frames with a 95% confidence interval from 33.76 to 33.84 dB, with PSNR of 35.7 dB for JPEG compression and 32 dB for data loss, ensuring reliable visual recovery [32].

4.3. Comparison with the Existing Approaches

4.3.1. Comparative Evolution of Frame PSNR

As Fig. 7 reveals, the proposed method achieved a stable PSNR of 42 dB over 300 frames, with a 95% confidence interval of [41.94, 42.06] dB, thus ensuring high visual quality according to [33]. In contrast, Chaotic Maps, Scalable Video Coding (SVC), and Selective Video Encryption (H.264) exhibited lower PSNR values of 34.02 dB,

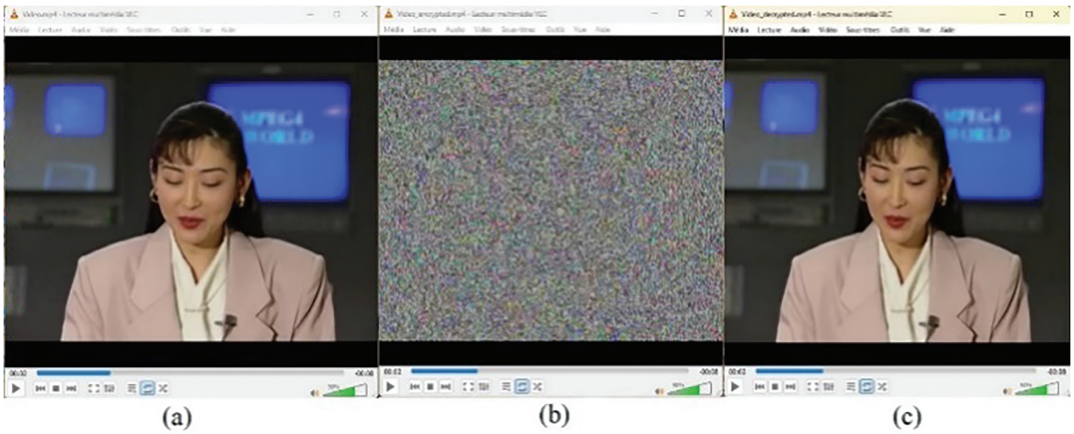

Fig. 5. Comparison of (a) original, (b) encrypted, and (c) decrypted video frames.

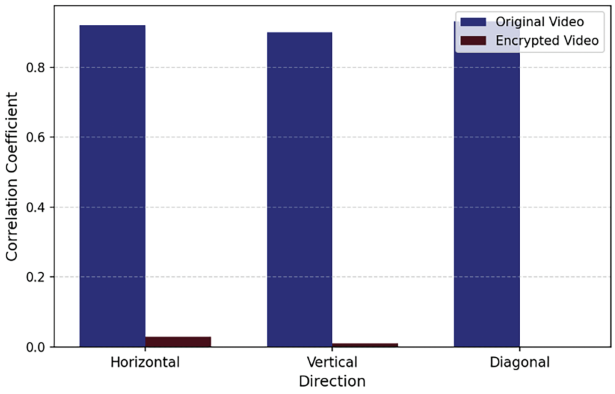

Fig. 6. Correlation between adjacent pixels.

37.95 dB, and 30.09 dB, respectively, indicating greater visual loss [34]–[37].

4.4. *Average SSIM Distribution*

Fig. 8 illustrates that the Proposed Method achieved an SSIM of 0.95 over 300 frames with a 95% confidence interval from 0.949 to 0.951, exceeding SVC at 0.92 [36] but slightly below the perfectly stable 1.0 reported in [38].

4.5. *Encrypted Frame Entropy Evolution*

Fig. 9 shows that the Proposed Method achieved about 8-bit entropy [38], comparable to Coding Characteristics at 7.89 bits [39] and Block Scrambling at 8 bits.

4.6. *Ablation Study on the Contribution of Model Components*

The ablation results, which Table V reveals, showed that combining Fourier and Riesz features with orthogonality, Jacobian adaptation, and hybrid ReLU/Sigmoid activation produces cryptographic keys of the highest quality.

TABLE II: Entropy and Horizontal/Vertical Correlations

Frame	Entropy (bits)	H-Corr (Orig)	H-Corr (Enc)	V-Corr (Orig)	V-Corr (Enc)
1	7.98	0.91	0.02	0.88	0.00
2	7.97	0.92	0.03	0.89	−0.01
3	7.99	0.93	0.01	0.90	0.02
4	7.96	0.90	0.00	0.87	0.01
5	7.98	0.91	−0.01	0.89	−0.02

TABLE III: Diagonal Correlations

Frame	D-Corr (Orig)	D-Corr (Enc)
1	0.87	0.01
2	0.88	0.00
3	0.89	−0.01
4	0.86	0.02
5	0.87	0.01

TABLE IV: Evaluation of Encryption Robustness against Different Types of Attacks

Type of attack	Average PSNR	Interpretation
Gaussian noise	∼33.8 dB	Minimal visual degradation; video remains usable.
JPEG compression (Q = 75%)	∼35.7 dB	The encryption withstands moderate compression well.
Packet loss	∼32 dB	Good robustness; video remains intelligible.

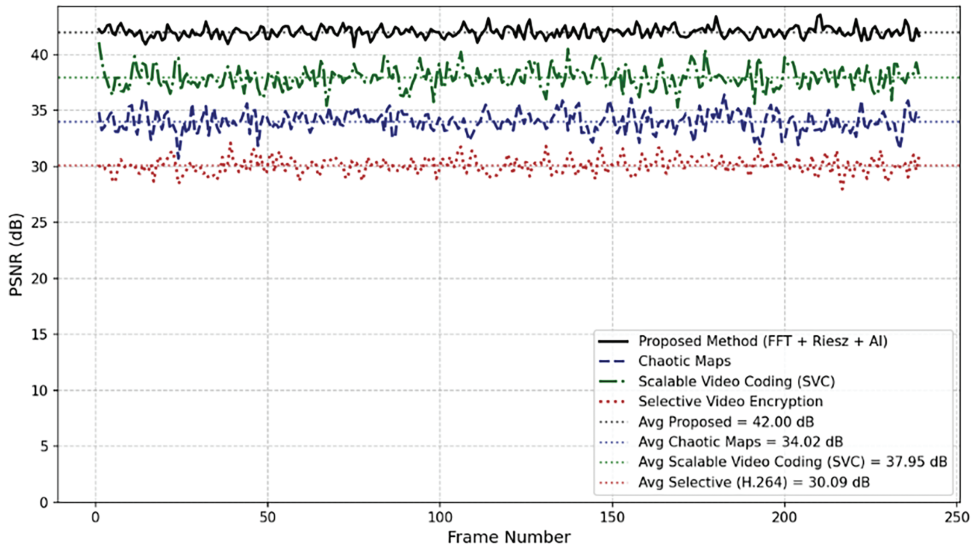

Fig. 7. PSNR performance of proposed and selected existing approaches.

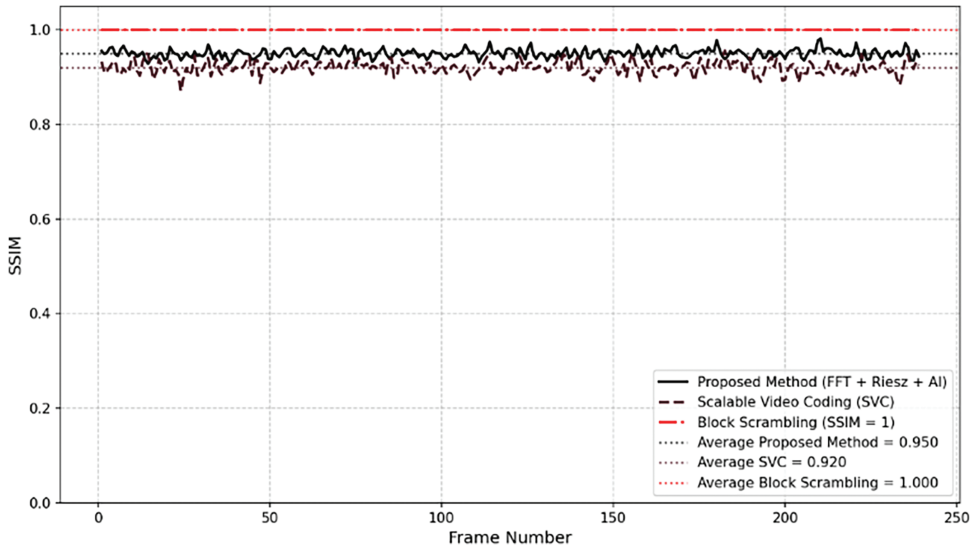

Fig. 8. Comparative evolution of frame SSIM: Proposed vs. existing methods.

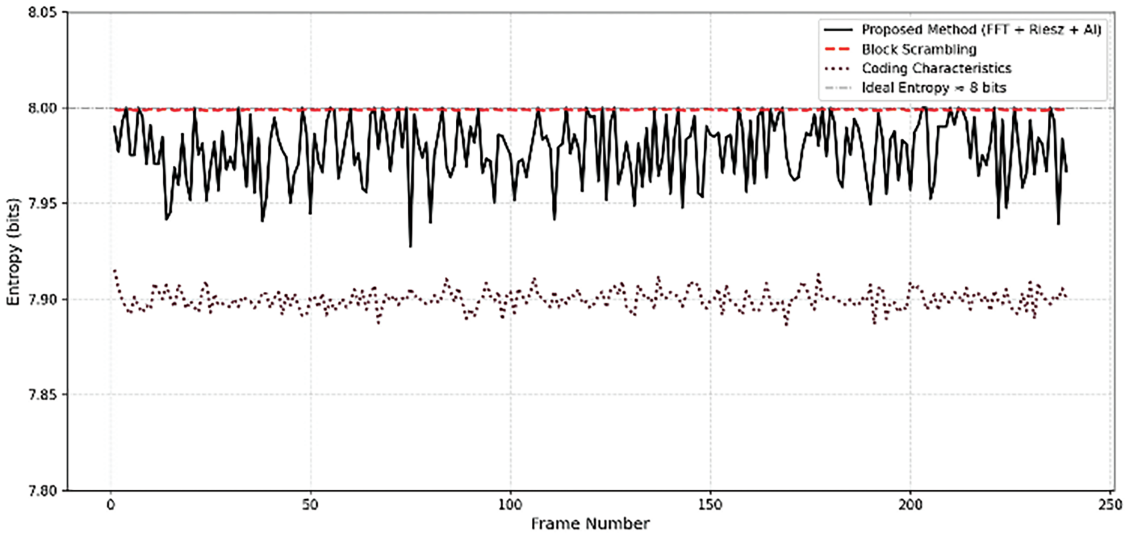

Fig. 9. Comparative evolution of encrypted frame entropy for the proposed method, coding characteristics and block scrambling.

TABLE V: Ablation Study Evaluating the Contribution of Each Component to Key Quality

Config.	Feat.	Ortho.	Jac.	Activation	Frame-wise	Ent.	Corr.	Ham.
C_1	F	No	No	ReLU	No	7.65	0.12	121.4
C_2	R	No	No	Sigmoid	No	7.62	0.15	119.8
C_3	F + R	No	No	Tanh	No	7.71	0.07	124.6
C_4	F + R	Yes	No	ReLU/Sigmoid	Partial	7.79	0.03	127.1
C_5 (proposed)	F + R	Yes	Yes	ReLU/Sigmoid	Yes	7.96	≈0.00	129.6

Note: Where Feat.: Features; Ortho.: Orthogonality; Jac.: Jacobian; Ent.: Entropy; Corr.: Correlation; Ham.: Hamming; F: Fourier features; R: Riesz features; F+R: combined Fourier and Riesz features.

4.7. Security Validation via Neural Discriminator Attack

A neural discriminator trained on 70% of 300 frames over 50 epochs achieved 49.8% ± 1.2% accuracy, showing encrypted frames are statistically indistinguishable from noise and confirming robustness against neural attacks.

4.8. Computational Performance Analysis

Training took 2.3 min on GPU and 9.8 min on CPU for 50 epochs, with an average inference time of 6.4 ms (GPU) and 28.7 ms (CPU), demonstrating near real-time processing. The theoretical complexity $O(E \times T(N^2 logN + Ld^2 + d^3))$ includes $N^2 logN$ for FFT extraction, Ld^2 for forward/backpropagation, and d^3 for QR factorization, and remains manageable due to GPU parallelization.

5. Discussion

The proposed method achieved an average key entropy of 7.67 bits, confirming proximity to the theoretical optimum of 8 bits and indicating strong randomness as well as resistance to statistical attacks, as reported in [40]. The avalanche effect produced an average Hamming distance of 129.62 bits with a 95% confidence interval ranging from 129.11 to 130.13 and a p-value below 0.001, thereby satisfying the strict diffusion criteria established in [41]. Adjacent-pixel correlations decreased from values close to 0.9 to statistically indistinguishable values from zero, in accordance with [40], [42]. Ablation analysis showed that configuration C_5 offered the best trade-off between entropy maximization, decorrelation efficiency, and nonlinear sensitivity, consistent with [42], [43].

The decrypted sequences achieved a PSNR close to 42 dB and an SSIM value around 0.95, with confidence intervals confirming stability across all frames. These results demonstrate the superiority of the method over selective encryption approaches described in [39] and chaos-based schemes presented in [44], while maintaining robustness against noise, compression, and packet loss, as shown in [40], [43], [44].

Although the deep Fourier–Riesz framework introduced additional computational cost associated with feature extraction and constrained optimization, and requires hardware acceleration for strict real-time deployment, it offers a favorable trade-off between security, reconstruction fidelity, and statistical stability.

The method also demonstrates resilience against model-extraction attacks, as the neural discriminator failed to recover exploitable patterns, and it limits side-channel leakage through entropy-preserving transformations. Remaining limitations concern computational load and sensitivity to training diversity.

6. Conclusion

This research presented a novel framework for dynamic and adaptive key generation, leveraging Fourier–Riesz features combined with deep learning. The approach produces high-entropy, decorrelated, and robust keys, ensuring strong cryptographic properties for videos. Experimental results demonstrated that deep spectro-directional features effectively capture temporal and spatial variations, providing robust and independent keys for each frame. Future work will focus on optimizing the key generation process, integrating the framework into modern codecs such as High Efficiency Video Coding (H.265/HEVC), evaluating performance on high-resolution video sequences, exploring alternative spectro-directional transformations, and developing adaptive mechanisms to enhance robustness and scalability in dynamic video scenarios.

Appendix

TABLE VI: Notation Summary

Symbol	Description
Ω	Orthogonal matrix
$\overline{\psi}_t$	Spectro-directional tensor
N_θ	Deep neural network
λ_{orth}	Orthogonality penalty
λ_{div}	Diversity penalty
λ_{jac}	Jacobian regularization
K_t	Continuous encryption key
$K_t^{'}$	Generated key

Code Availability Statement

The code used in this research is available from the corresponding author upon reasonable request.

Conflict of Interest

The authors declare that they do not have any conflict of interest.

References

[1] Shahid Z, Chaumont M, Puech W. Fast protection of H.264/AVC by selective encryption of CAVLC and CABAC for I and P frames. *IEEE Trans Circ Syst Video Technol*. 2011;21(5):565–76.

[2] Li S, Chen G, Zheng X. Chaos-based encryption for digital images and videos. *Chaos Solitons Fractals*. 2004;22(2):341–61.

[3] Lian S. Multimedia content encryption techniques: current status and challenges. *Signal Process: Image Commun*. 2008;23(3):230–47.

[4] Liu F, Koenig H. A survey of video encryption algorithms. *Comput Secur*. 2010;29(1):315.

[5] Mousavi A, Baraniuk R. Learning to invert: signal recovery via deep convolutional networks. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017.

[6] Unser M, Van De Ville D. Wavelet steerability and the higher-order Riesz transform. *IEEE Trans Image Process*. 2010;19(3):636–52.

[7] Vorontsov A, Sun X, Burda M, Turner R. Orthogonality constraints in neural networks through Lie algebra parametrization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[8] Hashem MI, Kuban KH. Key generation method from fingerprint image based on deep convolutional neural network model. *Nexo Revista Científica*. 2023;36(6):906–25.

[9] Kuznetsov O, Zakharov D, Frontoni E. Deep learning-based biometric cryptographic key generation with post-quantum security. *Multimed Tools Appl*. 2024;83(19):56909–38.

[10] Yirga TG, Yirga HG, Addisu EG. Cryptographic key generation using deep learning with biometric face and finger vein data. *Front Artif Intell*. 2025;8:1543268.

[11] Erkan U, Toktas A, Enginoğlu S, Akbacak E, Thanh DNH. An image encryption scheme based on chaotic logarithmic map and key generation using deep CNN. *Multimed Tools Appl*. 2022;81:7365–91.

[12] Wang X, Shao Z, Li B, Fu B, Shang Y, Liu X. Color image encryption based on discrete trinion Fourier transform and compressive sensing. *Multimed Tools Appl*. 2024;83(26):67701–22.

[13] Video Test Media. YUV video sequences dataset. 2019. [Online]. Available from: https://media.xiph.org/video/derf/. [Accessed: Mar 30, 2026].

[14] Jakubovitz D, Giryes R. *Improving DNN Robustness to Adversarial Attacks Using Jacobian Regularization*. Tel Aviv University, Tech. Rep.; 2018.

[15] Lizama-Pérez LA. *XOR Chain and Perfect Secrecy at the Dawn of the Quantum Era*. Universidad Técnica Federico Santa María, Tech. Rep.; 2019.

[16] Schneier B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. 2nd ed. New York, NY, USA: Wiley; 2015.

[17] Menezes A, Van Oorschot P, Vanstone S. *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press; 1996.

[18] Stallings W. *Cryptography and Network Security: Principles and Practice*. Pearson; 2017.

[19] National Institute of Standards and Technology (NIST). *Security Requirements for Cryptographic Modules*. FIPS PUB; 2001. p.140-2.

[20] Contreras-Rodríguez L, Madarro-Capó EJ, Contreras-Rodríguez L, Legón-Pérez CM, Rojas O, Sosa-Gómez G. Selecting an effective entropy estimator for short sequences of bits and bytes with maximum entropy. *Entropy*. 2021;23:561.

[21] Matsui M. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology–EUROCRYPT '93, LNCS 765*. Springer, 1994. pp. 386–97.

[22] Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems. *J Cryptol*. 1991;4(1):3–72.

[23] Daemen J, Rijmen V. *The Design of Rijndael: AES—The Advanced Encryption Standard*. Springer; 2002.

[24] Shannon CE. Communication theory of secrecy systems. *Bell Syst Tech J*. 1949;28(4):656–715.

[25] Wang X, Yu H. How to break MD5 and other hash functions. In *Advances in Cryptology–EUROCRYPT 2005*. Springer, 2005. pp. 19–35.

[26] Li C, Lin D, Lo K. Cryptanalysis of an image encryption scheme based on a compound chaotic sequence. *Signal Process: Image Commun*. 2017;52:130–9.

[27] Wu Y, Noonan JP, Agaian S. NPCR and UACI randomness tests for image encryption. *Cyber J: Multidiscip J Sci Technol*. 2011;1(2):31–8.

[28] Chen G, Mao Y, Chui CK. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Solitons Fractals*. 2004;21(3):749–61.

[29] Liu H, Wang X. Color image encryption using spatial chaotic systems. *Signal Process*. 2012;92(12):3492–501.

[30] Ghouate NE. A high-entropy image encryption scheme using optimized chaotic maps. *Sci Rep*. 2025;15(1):14784.

[31] Alexan W. A secure and efficient image encryption scheme based on a 5D hyperchaotic system. *Sci Rep*. 2025;15(1):15794.

[32] Gao S, Liu J, Iu HHC, Erkan S, Zhou S, Wu R, *et al.* Development of a video encryption algorithm for critical areas using 2D extended Schaffer function map and neural networks. *Signal Process: Image Commun*. 2024;117:103227.

[33] Kanungo A, Srivastava A, Anklesaria S, Churi P. A systematic review on video encryption algorithms: a future research. *J Auton Intell*. 2023;6(2):1–12.

[34] Salama WM, Aly MH. *Chaotic Maps Based Video Encryption: A New Approach*. Pharos University/AASTMT; 2020.

[35] Elkamchouchi H, Salama WM, Abouelseoud Y. *New Video Encryption Schemes Based on Chaotic Maps*. IET Image Processing; 2020.

[36] Wang H. A multi-level secure video encryption framework integrating scalable video coding with joint source-channel cryptography. *Proceedings of the CONF-MPCS Symposium*, 2025.

[37] Goyal D, Hemrajani N. Novel selective video encryption for H.264 video. *Int J Inform Secur Sci*. 2014;3(4):5161.

[38] Hosny KM, Zaki MA, Lashin NA, Hamza HM. Fast colored video encryption using block scrambling and multi-key generation. *Vis Comput*. 2023;39(12):6041–72.

[39] Cheng S, Wang L, Ao N, Han Q. A selective video encryption scheme based on coding characteristics. *Symmetry*. 2020;12(3):332.

[40] Das S, Jagan L, Singh GK, Kumar S, Rout J, Soni A, *et al.* Multilayered digital image encryption approach to resist cryptographic attacks for cybersecurity. *PeerJ Comput Sci*. 2025;11:e3260.

[41] Castro JCH, Sierra JM, Seznec A, Izquierdo A, Ribagorda A, *et al*. The strict avalanche criterion randomness test. *Math Comput Simul*. 2005;68(1):17.

[42] Panwar K, Kukreja S, Singh A, Singh KK. Towards deep learning for efficient image encryption. *Procedia Comput Sci*. 2023;218:644–50.

[43] Wang M, Fu X, Yan X, Teng L. A new chaos-based image encryption algorithm based on discrete Fourier transform and improved Joseph traversal. *Mathematics*. 2024;12(5):638.

[44] Xu H, Tong XJ, Zhang M, Wang Z, Peng J. Dynamic video encryption algorithm for H.264/AVC based on a spatiotemporal chaos system. *J Opt Soc Am A*. 2016;33(6):1166–74.