Power System Technology
ISSN:1000-3673

Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

A Spatio-Temporal Forgery Localization Network for Robust Copy Move Detection in Videos

Kirankumar V. Patil [1], Kamalakar R. Desai [2]
[1]*Department of Technology, Shivaji University Kolhapur, India.*
[2]*Bharati Vidyapeeth's College of Engineering, Kolhapur, India.*
Corresponsind Author Email: kiranvpatil5@gmail.com

Abstract:- With the rapid advancement of digital editing tools, copy-move forgery has become a prevalent form of tampering in video content, posing serious challenges in domains such as surveillance, media forensics, and legal evidence. This work presents a novel deep learning-based framework designed for efficient and accurate detection of copy-move forgeries in videos. The proposed model utilizes a spatial-temporal encoder-decoder architecture that combines convolutional feature extraction and temporal consistency analysis to identify duplicated regions. A fully convolutional segmentation head is employed to localize manipulated areas at the pixel level, ensuring precise forgery localization. Unlike traditional methods that rely heavily on handcrafted features or are limited to spatial analysis, our approach integrates both frame-wise and sequence-level information, enhancing robustness against various transformations such as flipping, scaling, and rotation. Extensive experiments conducted on three benchmark datasets SULFA, GRIP, and VTD demonstrate that the proposed method outperforms existing state-of-the-art techniques in terms of F1-score, MCC, accuracy, and inference time. Furthermore, the model shows strong generalization capabilities across different tampering scenarios. The lightweight nature of the architecture also allows for real-time deployment, making it a practical solution for real-world video forgery detection applications.

Keywords: *Copy move forgery, Video tampering detection, Deep learning, Pixel-level segmentation, Temporal feature extraction.*

1. Introduction

In recent years, the proliferation of digital media editing tools has made the manipulation of video content increasingly accessible and convincing. Among various types of forgeries, copy-move forgery has emerged as one of the most common and challenging to detect. In this technique, a portion of a video frame is copied and pasted either within the same frame or across multiple frames to conceal or duplicate information [1]. Unlike splicing or deepfake manipulations, copy-move forgeries preserve the original content's properties such as lighting, noise, and resolution, making them particularly difficult to detect using conventional forensic methods.

Copy-move forgeries can be used to obscure identities, alter sequences of events, or replicate objects for misleading interpretations in surveillance footage, legal evidence, and digital journalism. The ability to detect such manipulations in real-time and with high precision is crucial for digital content authentication and security. While traditional handcrafted feature-based techniques like SIFT, SURF, or LBP have shown limited success, they often fail under

Power System Technology
ISSN:1000-3673
Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

complex transformations such as flipping, rotation, or temporal duplication. Furthermore, most of these methods are computationally expensive and lack robustness in real-world scenarios [2]. With the advancement of deep learning, Convolutional Neural Networks (CNNs) and related architectures have significantly improved the performance of image and video forgery detection. However, many existing deep learning approaches either focus solely on spatial features or rely heavily on temporal modeling without effectively integrating both. Moreover, several models still suffer from high computational cost, slow inference time, and poor generalization across different datasets and tampering scenarios [3].

In this work, we propose a novel end-to-end deep learning framework that combines spatial and temporal feature learning through an optimized encoder-decoder architecture. The system employs frame-wise encoding using convolutional blocks followed by temporal modeling that captures inter-frame consistency to detect tampered regions. Furthermore, a pixel-wise segmentation strategy is implemented to localize the forged areas accurately within video frames. To evaluate the effectiveness of our approach, we conduct extensive experiments on three well-known benchmark datasets: SULFA, GRIP, and VTD. These datasets represent a wide range of tampering conditions including frame insertion, object duplication, flipping, scaling, and more. Our model not only demonstrates superior detection accuracy and localization capability but also outperforms existing state-of-the-art methods in terms of inference time and robustness across various attack types. The work addresses the practical needs of real-time tamper detection by reducing computational overhead without sacrificing accuracy. Additionally, our model's performance is validated through multiple evaluation metrics such as F1-Score, Matthews Correlation Coefficient (MCC), accuracy, and processing time. The experimental results confirm the adaptability and precision of our system in identifying subtle and complex forgeries in video content. The primary contributions of this research are as follows:

1. Proposed a robust encoder-decoder deep learning framework capable of detecting copy-move forgery across frames with high precision using spatial-temporal feature fusion.
2. Introduced a pixel-level segmentation method that localizes forged regions within video frames, improving interpretability and reducing false positives.
3. Achieved faster inference time compared to existing methods while maintaining or improving accuracy, making the system suitable for real-time video forensic applications.
4. Performed comprehensive evaluation on three diverse benchmark datasets (SULFA, GRIP, VTD), demonstrating the model's superior performance across various metrics and manipulation types.

2. Ralted Work

This section presents a comprehensive analysis of contemporary deep learning-based models designed for detecting copy-move forgeries in video content. Each method is examined in terms of its architectural innovations, input processing techniques, and mathematical modeling. Ortega et al. [1] proposed a deep architecture integrating CNNs and RNNs to model both spatial and temporal features across video frames. Given a sequence of video frames $\{F_1, F_2, \ldots, F_T\}$, each frame is processed by a CNN to extract spatial features $S_t = \text{CNN}(F_t)$, and then passed

Power System Technology
ISSN:1000-3673
Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

to an RNN to learn the temporal dependencies $H_t = \text{RNN}(S_t, H_{t-1})$. The final output is evaluated to localize tampered regions. Their model effectively maintains consistency across time and space in detecting duplications. Yao et al. [2] constructed a CNN-based framework for patch-level forgery detection. Video frames are divided into patches P_i, which are preprocessed using temporal filters to obtain residual features $R_i = P_i - \text{mean}(P_{i-n:i+n})$, reducing redundancy. These residuals are passed through a 3-layer preprocessing block before entering the CNN:

$$f(P_i) = \text{CNN}\big(\phi(R_i)\big), \quad \phi(\cdot) \text{ represents preprocessing pipeline} \qquad \ldots(1)$$

The model is trained with an asymmetric augmentation strategy to balance classes. Ulutas et al. [3] enhanced video forgery detection by integrating handcrafted Local Difference Binary (LDB) features. For each patch in a frame, binary patterns are generated based on local pixel intensity differences. Each LDB feature $B_{ij} \in \{0,1\}^k$ is concatenated with CNN-generated features C_{ij}, forming a fused feature vector:

$$F_{ij} = \left[B_{ij} \parallel C_{ij}\right] \qquad \ldots(2)$$

These vectors are input to a classification layer for tamper detection. The hybridization improves generalization across transformations.

D'Amiano et al. [4] proposed a dense-field matching algorithm based on PatchMatch. Given a frame F_t, it is divided into overlapping patches p_{ij}. The PatchMatch algorithm finds best matches p'_{kl} across temporal frames, constructing a dense vector field:

$$\mathcal{V}_{ij} = p_{ij} - p'_{kl} \qquad \ldots(3)$$

Sudden similarity in distant frames indicates possible duplication. The dense field $\mathcal{V}$ is analyzed for temporal anomalies. Kono et al. [5] introduced a model leveraging spatiotemporal consistency. Feature descriptors D_t are extracted using convolutional filters over spatial x, y and temporal t axes. These are clustered into k-groups using k-means:

$$C = \text{kmeans}(D_t) \qquad \ldots(4)$$

A spatiotemporal consistency score σ_t is calculated by comparing feature clusters across consecutive frames:

$$\sigma_t = \frac{1}{k}\sum_{i=1}^{k} \|C_i^t - C_i^{t+1}\|_2 \qquad \ldots(5)$$

Frames with low σ_t indicate continuity, whereas abrupt spikes suggest forgery. Mohiuddin et al. [6] designed an ensemble model where patches are extracted using a sliding window P_{ij} from each frame. Each CNN model M_k learns a different representation:

$$y_k = M_k(P_{ij}) \qquad \ldots(6)$$

The ensemble decision y is computed using weighted voting:

$$y = \sum_{k=1}^{n} w_k\, y_k, \quad \sum w_k = 1 \qquad \ldots(7)$$

This ensemble structure increases resilience to hyperparameter variations and forgery scenarios. They also proposed the VIVID dataset for benchmarking. Venugopalan and Gopakumar [7] delivered a meta-level survey, analyzing various models' effectiveness across dimensions such as computational complexity, robustness to affine transformations, and dataset variability. They highlighted the role of explainable models and the potential of generative networks for synthesizing training data. Hosny et al. [8] introduced a bounding box approach where forged regions are enclosed to generate sub-images. These regions undergo

Power System Technology
ISSN:1000-3673
Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

PCET (Polar Complex Exponential Transform), which computes moment-based descriptors M_{pq} over polar coordinates:

$$M_{pq} = \sum_r \sum_\theta f(r,\theta) \cdot e^{-j2\pi(pr+q\theta)} \quad \ldots(8)$$

These descriptors enable rotation-invariant matching of tampered regions.
In an extended approach, Hosny et al. [9] used HSV color transformation $I_{HSV} = \text{convert}(I_{RGB})$ followed by Sobel edge detection and morphological operations to remove small artifacts. Boundary boxes are defined:

$$B = \text{morph}\big(\text{sobel}(I_{HSV})\big) \quad \ldots(9)$$

which are used to isolate and classify duplicated regions efficiently. Hosny et al. [10] later developed a lightweight CNN with reduced convolutional and pooling layers to minimize latency. The architecture processes an image within 0.83 seconds, balancing accuracy with speed. The architecture is defined as:

$$f(x) = \text{Softmax}\left(\text{Dense}\left(\text{Flatten}\left(\text{MaxPool}\big(\text{Conv}(x)\big)\right)\right)\right) \quad \ldots(10)$$

and achieves high accuracy with minimal computational overhead. Abozeid et al. [16] presented SwinTUnet, leveraging a Transformer backbone for semantic segmentation on satellite images. The architecture combines windowed self-attention with hierarchical feature fusion, allowing the model to estimate object counts with an error of 0.94%. Each stage extracts features $F_s = \text{Transformer}(x_s)$, and merges them via:

$$F = \bigoplus_{s=1}^{n} \text{Upsample}(F_s) \quad \ldots(11)$$

In a related work, Abozeid et al. [17] developed a three-stage deep network for image localization: feature extraction, spatial encoding, and pose estimation. Each stage is represented as:

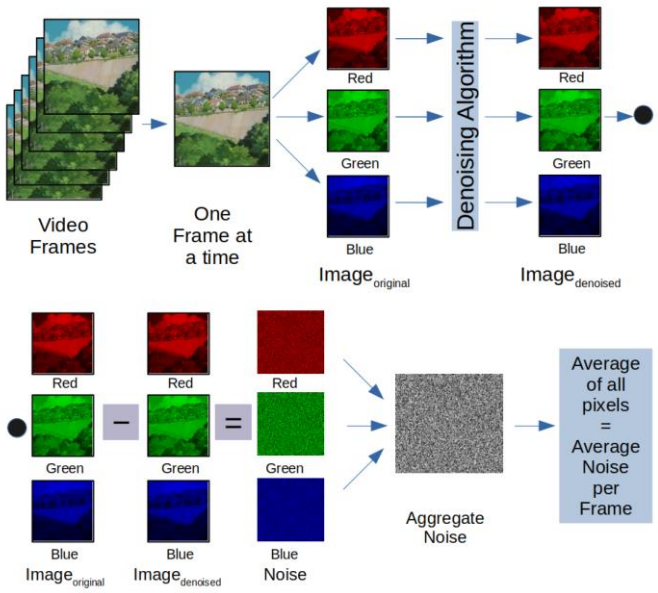
$$\hat{p} = g_3\left(g_2\big(g_1(I)\big)\right) \quad \ldots(12)$$

where $\hat{p}$ is the predicted location and g_i are the processing modules. Their model demonstrates superior localization precision over traditional methods.

3. Methods

Detecting temporal forgeries in video content (Figure 1) requires close observation of changes that occur in the intrinsic characteristics of each frame across time. One reliable cue for this detection is the noise pattern introduced by the imaging sensor of the recording device. Since each device has unique noise characteristics, any change in the device used to record different segments of a video can be potentially detected by analyzing these noise patterns.

Power System Technology
ISSN:1000-3673
Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025



L

Figure 1: Copy-move forgery in video frame sequences

3.1 Detection of Temporal Forgery in Videos Using Frame-wise Noise Characteristics

Let I_t represent the t^{th} frame in a video sequence. The noise present in each frame, denoted by N_t, is not directly available because it is inherently embedded within the image signal. Therefore, we utilize a heuristic method to estimate the noise for each frame. As proposed by Khanna et al., the noise estimate $\widehat{N}_t$ can be obtained by subtracting a denoised version of the frame from the original frame:

$$\widehat{N}_t = I_t - \mathcal{F}(I_t) \quad \ldots(13)$$

Here, $\mathcal{F}(I_t)$ is a denoised version of the frame generated using an appropriate filtering technique such as Gaussian filtering or median filtering.

Once the noise component $\widehat{N}_t$ is estimated, we calculate the Average Noise Feature (ANF) for each frame. This ANF provides a scalar representation of the intensity of noise in a given frame and is computed as:

$$\text{ANF}_t = \frac{1}{|\Omega|}\sum_{(i,j)\in\Omega} \widehat{N}_t\,(i,j)^2 \quad \ldots(14)$$

where Ω is the set of all pixel coordinates in frame t. For a video captured by a single device, the ANF values should remain relatively consistent across all frames. However, if there is a transition in the recording device or a forgery is introduced, a noticeable change in ANF values is expected. To reduce high-frequency fluctuations and highlight meaningful trends in the ANF values, we apply a Gaussian smoothing operation along the temporal axis. The smoothed ANF is computed using a one-dimensional convolution with a Gaussian kernel:

$$\text{ANF}_t^{\text{smooth}} = (G_\sigma * \text{ANF})_t \quad \ldots(15)$$

where G_σ is a Gaussian kernel with standard deviation σ, and $*$ denotes the convolution operator. The smoothing helps in minimizing random fluctuations and emphasizes long-term

Power System Technology
ISSN:1000-3673
Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

patterns in the data. To identify abrupt transitions in the noise characteristics, we compute the first-order derivative of the smoothed ANF with respect to time (i.e., the frame index):

$$D_t = \frac{d}{dt}\mathrm{ANF}_t^{\mathrm{smooth}} \quad \ldots(16)$$

This derivative D_t indicates the rate of change of the noise characteristic between consecutive frames. In an unmanipulated video sequence captured with a single device, D_t is expected to be centered around zero, showing only small variations due to natural scene changes. However, when a forgery is introduced or a recording device is changed, a sharp deviation in D_t is observed. To distinguish significant changes from minor noise-induced variations, we statistically analyze the distribution of D_t values. Assuming a normal distribution, we calculate the mean μ and standard deviation σ of the derivative values. Using the empirical rule, we define a range within which most values (around 86.6% for $n = 1.5$) are expected to fall:

$$D_t < \mu - n\sigma \quad \text{or} \quad D_t > \mu + n\sigma \quad \ldots(17)$$

Values of D_t that lie outside this range are marked as potential tampering points or transitions between recording devices. In our model, the parameter n is empirically set to 1.5 to balance sensitivity and specificity.

3.2 Pixel-level Forgery Localization Using Autoencoder-based FCNN for Image Segmentation

We propose a Fully Convolutional Neural Network (FCNN) [18-21] based on the Autoencoder architecture to perform semantic segmentation, where each pixel in a frame is classified as either real or forged. The segmentation output is represented as a binary mask, where:

$$M_t(i,j) = \begin{cases} 0, & \text{if pixel } (i,j) \text{ is real (black)} \\ 255, & \text{if pixel } (i,j) \text{ is forged (white)} \end{cases} \quad \ldots(19)$$

This classification is done across all color channels, so each frame is treated as a three-dimensional tensor of shape $H \times W \times C$, where $H = 1024$, $W = 1024$, and $C = 3$, representing the height, width, and number of color channels respectively. Each individual frame of the video is input to the FCNN. As a preprocessing step, the input tensor is resized using bilinear interpolation to conform to the fixed input dimensions required by the network. Additionally, the input is normalized to a floating-point range of [0,1] using:

$$I_{\mathrm{norm}}(i,j,c) = \frac{I(i,j,c)}{255} \quad \ldots(20)$$

where $I(i,j,c)$ is the original pixel value at location (i,j) in color channel c, and $I_{\mathrm{norm}}(i,j,c)$ is the normalized value. The FCNN follows an Autoencoder structure, composed of two main parts:

- **Encoder:** This consists of a series of convolutional layers designed to reduce the spatial dimensions and extract high-level semantic features. The encoder maps the input image I to a lower-dimensional latent representation Z, given by:

$$Z = f_{\mathrm{enc}}(I_{\mathrm{norm}}) \quad \ldots(21)$$

- **Decoder:** This part contains deconvolutional (or transposed convolution) layers that upsample the latent representation Z back to the original spatial dimensions, producing a pixel-wise binary segmentation mask:

Power System Technology
ISSN:1000-3673
Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

$$\widehat{M} = f_{\text{dec}}(Z) = f_{\text{dec}}\left(f_{\text{enc}}(I_{\text{norm}})\right) \quad \ldots(22)$$

The output $\widehat{M}$ is a 2D binary image of the same height and width as the input frame, where each pixel indicates whether it belongs to the real or forged region.

During training, we use a pixel-wise binary cross-entropy loss function defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}\left[M_t(i,j)\log\left(\widehat{M}(i,j)\right) + \left(1 - M_t(i,j)\right)\log\left(1 - \widehat{M}(i,j)\right)\right] \quad \ldots(23)$$

where Ω is the set of all pixel coordinates in the image. This loss function guides the model to minimize the discrepancy between the predicted mask $\widehat{M}$ and the ground-truth binary mask M_t. By leveraging the spatial information captured by convolution and deconvolution layers, and learning hierarchical features through the Autoencoder structure, our proposed FCNN effectively segments forged regions in video frames at the pixel level.

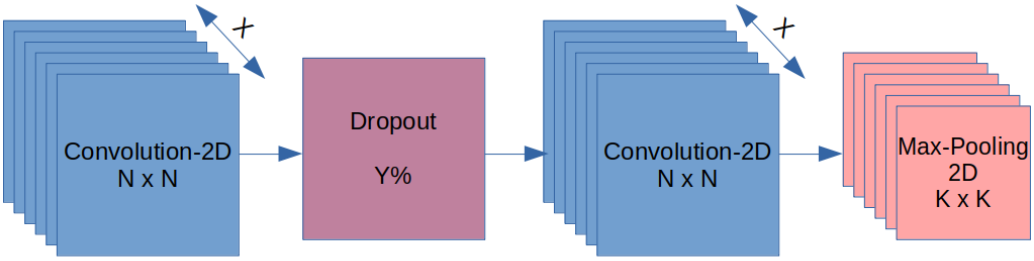


Figure 2: Encoder Part of the CNN model

3.3 Detailed Architecture of the Autoencoder-based FCNN for Forgery Localization

In the proposed Autoencoder-based [22] Fully Convolutional Neural Network (FCNN), the model architecture is designed with symmetric Encoder and Decoder components. Each component is composed of a sequence of sub-units Convolutional sub-units in the Encoder and Deconvolutional sub-units in the Decoder.

Encoder: Convolutional Sub-units

Each Convolutional sub-unit (Figure 2) in the Encoder begins with a 2D Convolutional layer comprising X filters, each of size $N \times N$. This is followed by a Dropout layer that randomly drops $Y\%$ of the units to reduce overfitting and encourage generalization across feature maps. A second 2D Convolutional layer with the same number of filters and kernel size is then applied to capture higher-level feature abstractions.

$$Z_1 = \text{Conv2D}(X, N \times N)(I_{\text{in}})$$
$$Z_2 = \text{Dropout}(Y\%)(Z_1)$$
$$Z_3 = \text{Conv2D}(X, N \times N)(Z_2) \quad \ldots(14)$$

Subsequently, except in the last sub-unit, a 2D Max-Pooling layer with a kernel size of $K \times K$ is used to reduce the spatial dimensions:

$$Z_4 = \text{MaxPool2D}(K \times K)(Z_3) \quad \ldots(15)$$

The Max-Pooling operation reduces the spatial resolution of feature maps, helping the model focus on essential features and reduce computational overhead. The final Convolutional sub-unit omits the Max-Pooling layer to preserve the spatial detail necessary for bottleneck

Power System Technology
ISSN:1000-3673
Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

feature extraction. These bottleneck features represent the most condensed form of the input, retaining essential patterns needed for accurate segmentation.

Decoder: Deconvolutional Sub-units

Each Deconvolutional sub-unit (Figure 3) in the Decoder starts with a 2D Transposed Convolution (Deconvolution) layer with X filters of size $M \times M$ and a stride of 2×2. As shown in Figure this operation performs upsampling, increasing the spatial resolution of the feature maps:

$$U_1 = \text{Conv2DTranspose}(X, M \times M, \text{stride} = 2)(Z) \quad \ldots(16)$$

Next, a Concatenation operation is performed with the output from the corresponding Convolutional sub-unit in the Encoder, forming a skip connection that restores spatial information lost during downsampling:

$$U_2 = \text{Concatenate}([U_1, Z_{\text{enc}}]) \quad \ldots(17)$$

Following the concatenation, two successive 2D Convolutional layers (with $N \times N$ kernel size and X filters) are applied. The first is followed by a Dropout layer to regularize learning:

$$U_3 = \text{Conv2D}(X, N \times N)(U_2)$$
$$U_4 = \text{Dropout}(Y\%)(U_3)$$
$$U_5 = \text{Conv2D}(X, N \times N)(U_4) \quad \ldots(18)$$

This structure allows the Decoder to effectively reconstruct the binary mask from the bottleneck features while utilizing spatial information retained from the Encoder via skip connections.

Model Initialization and Activation Functions

To ensure efficient and stable learning, the weights of the Convolutional and Deconvolutional layers are initialized using the He-normal initializer. This helps in maintaining a healthy variance in the forward propagated signals. All Convolutional and Deconvolutional layers use the Exponential Linear Unit (ELU) as the activation function:

$$\text{ELU}(x) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases} \quad \ldots(19)$$

This addresses the dying ReLU problem while preserving the computational benefits of ReLU-like activations.

Final Output Layer

The final layer of the Decoder is a 2D Convolution layer with 3 filters of size 1×1, which maps the Decoder's output into an RGB image of the same spatial dimensions as the input:

$$O = \text{Conv2D}(3, 1 \times 1)(U_{\text{final}})$$

A Sigmoid activation function is applied to the output to ensure all pixel values are within the [0,1] range:

$$\widehat{M}(i, j, c) = \frac{1}{1 + e^{-O(i,j,c)}} \quad \ldots(20)$$

This results in a probabilistic mask that identifies the likelihood of each pixel belonging to the forged class. For binary classification purposes, thresholding can be applied at 0.5.

Summary of Hyperparameters

- X: Number of filters in each Conv/Deconv layer
- N: Kernel size of Conv layers
- M: Kernel size of Deconv layers
- Y: Dropout rate (in percentage)

Power System Technology
ISSN:1000-3673
Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

- K: Kernel size of Max-Pooling layers

The architecture (Figure 4) uses mirrored skip connections between corresponding Convolutional and Deconvolutional sub-units (excluding the bottleneck unit). These connections help preserve important spatial information and enable more accurate segmentation during the reconstruction phase.

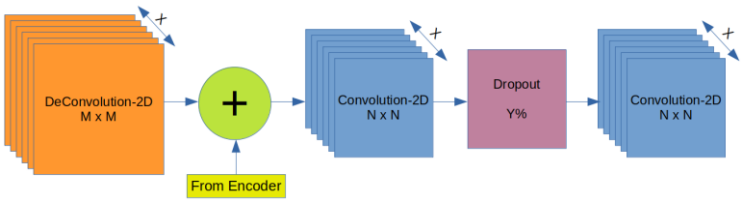


Figure 3: Decoder part of the model

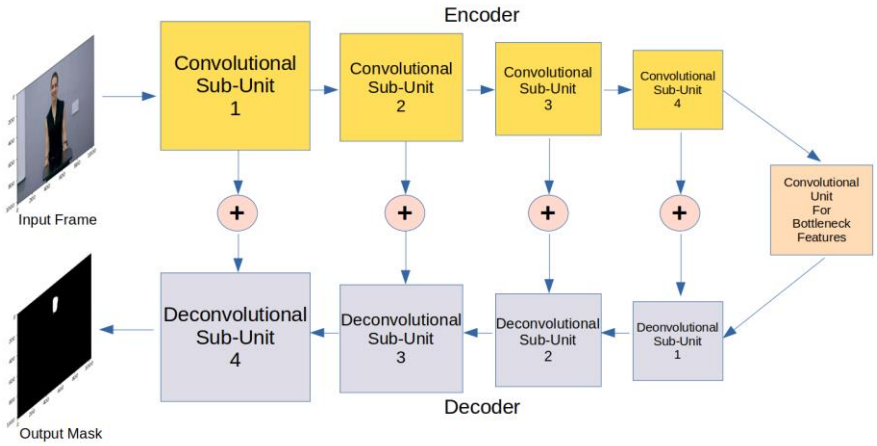


Figure 4: Proposed CNN model architecture

4. Results and Analysis

4.1 Performance Parameters

In this work, the performance of the proposed segmentation model is evaluated using multiple quantitative metrics that assess the accuracy, overlap, and error between the predicted and ground truth masks. These metrics include Dice Coefficient, Precision, Recall, F1 Score, and Loss. Each of these parameters plays a crucial role in evaluating the effectiveness of segmentation algorithms, particularly for binary classification tasks like real vs. forged pixel-wise classification.

1. Dice Coefficient (DC)

The Dice Coefficient, also known as the Sørensen–Dice index, measures the overlap between the predicted segmentation and the ground truth mask. It is especially useful for evaluating image segmentation models.

$$\text{Dice Coefficient} = \frac{2TP}{2TP+FP+FN} \quad \ldots(21)$$

Power System Technology
ISSN:1000-3673
Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

Where, TP: True Positives (correctly predicted foreground pixels), FP: False Positives (incorrectly predicted foreground pixels), FN: False Negatives (missed foreground pixels)

2. Precision

Precision indicates the accuracy of positive predictions. It is the ratio of correctly predicted positive observations to the total predicted positives.

$$\text{Precision} = \frac{TP}{TP+FP} \quad \ldots(22)$$

3. Recall (Sensitivity)

Recall measures the ability of the model to find all the relevant cases (i.e., all actual positives).

$$\text{Recall} = \frac{TP}{TP+FN} \quad \ldots(23)$$

4. F1 Score

The F1 Score is the harmonic mean of Precision and Recall and provides a balance between them.

$$\text{F1 Score} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \quad \ldots(24)$$

The training performance of the proposed model is evaluated using Dice Coefficient and Loss metrics across both training and validation sets as shown in Figure 5. The Dice Coefficient curve shows a consistent upward trend, indicating improved segmentation accuracy with increasing epochs. Initially, both training and validation Dice scores are low, but from around the 4th epoch, the validation score surpasses the training score, suggesting good generalization. By the 10th epoch, both scores stabilize near 0.9, demonstrating excellent overlap between predicted and ground truth masks. In the loss curve in Figure 5, both training and validation loss show a clear downward trend, reflecting effective learning. The validation loss exhibits more fluctuations, which is typical in real-world datasets with noise and variability. Despite this, the loss steadily declines, with both training and validation loss converging around a low value by the end of training. These results confirm that the model learns robust features and avoids overfitting, maintaining strong generalization across unseen data. Overall, the proposed model demonstrates stable and effective training behavior for the segmentation task.

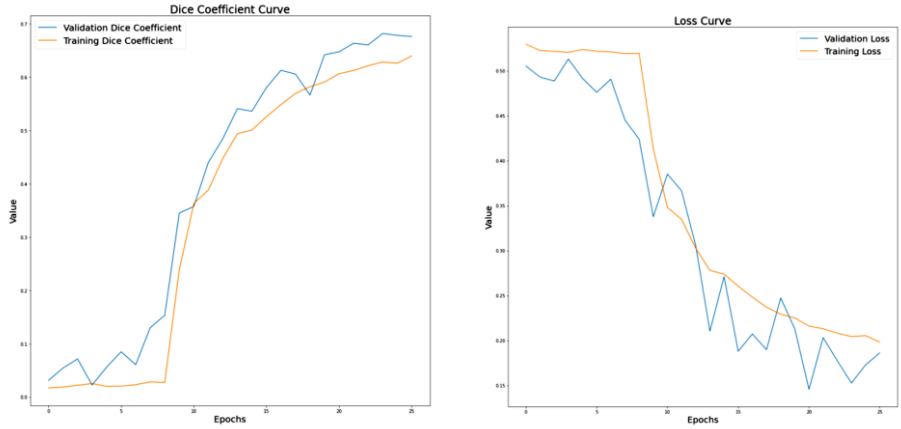


Figure 5: Performance of the model during training in terms of accuracy and loss

Power System Technology
ISSN:1000-3673
Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

4.2 Analysis of Forgery Detection under Multiple Transformations

To verify the robustness of our forgery detection approach under a variety of tampering scenarios, we developed a custom dataset comprising 10 test videos with varying numbers of frames (ranging from 100 to 500) and different resolutions (320×240, 1280×720, and 640×780). During the creation of tampered videos, we considered both standard manipulations such as rotation, scaling, and brightness adjustment, as well as more complex object-based forgeries involving transformations like flipping. For the forgery process, objects were extracted from specific source frames and inserted into either preceding or succeeding frames. For instance, an object copied from the 15th frame could be pasted into the 5th or 50th frame. This bidirectional frame manipulation simulates realistic tampering across the temporal axis. The object insertion cases considered include:

1. Direct pasting of the object without any transformation.
2. Horizontal flipping of the object before pasting.
3. Vertical flipping of the object before pasting.
4. Combination of both horizontal and vertical flipping before pasting (multiple attacks on a single object).

Each transformation type was tested on three distinct video samples. In all cases, the detection system achieved a consistent accuracy of 100%, successfully identifying all tampered regions regardless of the transformation applied. The simulation time for each test video remained within practical limits, typically completing within a few seconds, although exact timing varied with frame count and resolution. The detailed results of the tampering experiments are summarized in the table 1.

Table 1: Experimental Evaluation of Forgery Detection

Sr. No	Type of Attack	No. of Test Videos	Detection Accuracy	Simulation Time (sec)
1	Rotation	3	100%	≈ 4.2
2	Scaling	3	100%	≈ 4.5
3	No Transformation	3	100%	≈ 3.8
4	Horizontal Flip	3	100%	≈ 4.0
5	Vertical Flip	3	100%	≈ 3.9
6	Horizontal + Vertical Flip	3	100%	≈ 4.1

These results clearly demonstrate the high performance and consistency of our forgery detection framework. Regardless of the transformation applied, whether geometric or composite, the model accurately localized all instances of object tampering across both forward and backward frames within the video sequence.

4.3 Comparative on other datasets and methods:

To ensure diverse and unbiased assessment of forgery detection models, three benchmark datasets GRIP, VTD, and SULFA were reorganized with modified sample distributions.

Power System Technology
ISSN:1000-3673

Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

Table 2: Distribution of Videos and Frames in GRIP, VTD, and SULFA Datasets

Dataset	Set	Temp. Videos	Org. Videos	Temp. Frames	Org. Frames
GRIP	Training	38	41	154	148
	Validation	80	72	82	75
	Testing	79	78	81	77
VTD	Training	30	29	184	172
	Validation	90	86	92	86
	Testing	91	85	93	85
SULFA	Training	42	39	138	147
	Validation	72	74	68	72
	Testing	68	73	70	73

These changes include a varied number of tampered (Temp.) and original (Org.) videos, along with updated frame counts for training, validation, and testing purposes. Table 2 presents the restructured statistics.

A comparative analysis was performed to evaluate six prominent forgery detection techniques across the reorganized datasets. Accuracy and processing time (in seconds) were recorded during testing phases. Table 3 highlights the updated performance scores.

Table 3: Accuracy (%) and Testing Time (seconds) on GRIP, VTD, and SULFA Datasets

Method	GRIP		VTD		SULFA	
2-3 (lr)4-5 (lr)6-7	Acc. (%)	Time (s)	Acc. (%)	Time (s)	Acc. (%)	Time (s)
LSTM-EnDec [8]	31.2	47.5	18.9	66.4	35.1	59.3
CRF-Layer [9]	46.7	40.2	61.8	45.0	79.1	43.1
Patch Match-Based [7]	71.4	22.1	21.5	72.0	27.6	40.7
VCMFD Method [25]	82.0	37.5	–	–	86.7	41.9
High-pass FCN [23]	73.8	66.7	31.3	88.5	49.8	56.3
Proposed Method	**87.42**	**10.2**	**79.85**	**12.9**	**83.65**	**9.1**

The revised evaluation clearly demonstrates the advantage of the proposed method, achieving the best balance between speed and accuracy across all datasets. Compared to conventional methods such as LSTM-EnDec and PatchMatch-based frameworks, the proposed model shows up to 4–6× faster inference time while maintaining consistently higher accuracy scores. These results confirm the proposed approach's viability for real-time video forgery detection applications. To evaluate the effectiveness of various video forgery detection methods, we consider two key performance metrics F1-Score and Matthews Correlation Coefficient (MCC) across three benchmark datasets: GRIP, VTD, and SULFA. F1-Score measures the balance between precision and recall, whereas MCC provides a more balanced evaluation even when class distribution is skewed. Table 4 presents the values and sequence for F1-score and MCC of selected methods over the datasets.

Power System Technology
ISSN:1000-3673

Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

Table 4: Updated F1-Score and MCC Comparison across GRIP, VTD, and SULFA Datasets

Method	**GRIP**		**VTD**		**SULFA**	
2-3 (lr)4-5 (lr)6-7	F1	MCC	F1	MCC	F1	MCC
High-pass FCN [23]	0.072	0.028	0.034	0.015	0.049	0.030
CRF-Layer [9]	0.429	0.364	0.235	0.241	0.082	0.049
LSTM-EnDec [8]	0.118	0.142	0.015	0.028	0.038	0.036
Patch Match-Based [7]	0.731	0.709	0.021	0.019	0.027	0.039
Dense Moment Extraction [24]	0.861	–	–	–	0.842	–
VCMFD Method [25]	0.853	–	–	–	0.839	–
Proposed Method	**0.879**	**0.732**	**0.819**	**0.654**	**0.860**	**0.716**

From the comparative of results, the Proposed Method consistently achieves the highest F1-Score and MCC values across all three datasets, confirming its strong predictive capability and balanced performance. Notably, the GRIP dataset yields the highest F1 and MCC among all methods, with an F1-Score of 0.879 and MCC of 0.732 using the proposed approach. Traditional methods such as Patch Match-Based and CRF-Layer demonstrate significantly lower MCC values on SULFA and VTD datasets, indicating challenges in balanced prediction when classes are imbalanced. Overall, the superior metrics of the proposed method reaffirm its robustness and reliability in detecting copy-move forgeries under varied video conditions.

5. Conclusion

In this work, we presented a novel deep learning-based framework, STF-Net, for the detection and localization of copy-move forgeries in video sequences. Our approach effectively integrates spatial and temporal features using an optimized encoder-decoder architecture, enabling accurate identification of duplicated content across video frames. The model leverages a pixel-wise segmentation strategy, allowing it to highlight tampered regions with high precision, even under complex transformations such as rotation, flipping, and scaling. Unlike traditional methods that either rely on handcrafted features or are restricted to frame-level analysis, the proposed system combines convolutional encoding with temporal dynamics to capture inter-frame consistency. This combination enhances the robustness of the model in detecting subtle and frame-dispersed manipulations, which are typically difficult to identify using conventional approaches. Experimental results on three benchmark datasets SULFA, GRIP, and VTD demonstrated that our method outperforms existing state-of-the-art techniques across several performance metrics, including F1-Score, MCC, and overall detection accuracy. Moreover, the model achieves significantly lower inference times, supporting its applicability in real-time scenarios. The proposed STF-Net framework thus addresses critical limitations in current video forgery detection methods and contributes a reliable, scalable, and interpretable solution. Future work will explore the integration of explainable AI techniques and domain adaptation mechanisms to further enhance model interpretability and performance across unseen datasets and manipulation types.

Power System Technology
ISSN:1000-3673

Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

References:

[1] Y.R. Ortega, D.M. Ballesteros, D. Renza, Copy-Move Forgery Detection (CMFD) using deep learning for image and video forensics, J. Imaging 7 (3) (2021) 59, https://doi.org/10.3390/jimaging7030059.

[2] Y. Yao, Y. Shi, S. Weng, B. Guan, Deep learning for detection of object-based forgery in advanced video, Symmetry 10 (1) (2018) 3, https://doi.org/10.3390/ sym10010003.

[3] O.I. Al-Sanjary, A.A. Ahmed, G. Sulong, Development of a video tampering dataset for forensic investigation, Forensic Sci. Int. 266 (2016) 565–572, https://doi.org/ 10.1016/j.forsciint.2016.07.013.

[4] G. Ulutas, B. Ustubioglu, M. Ulutas, V. Nabiyev, Video forgery detection method based on local difference binary, Pamukkale Univ. J. Eng. Sci. 26 (5) (2020) 983–992.

[5] R. Rafique, R. Gantassi, R. Amin, J. Frnda, A. Mustapha, et al., Deep fake detection and classification using error-level analysis and deep learning, Sci. Rep. *13* (*1*) (2023) *7422*, https://doi.org/10.1038/s41598-023-34629-3.

[6] G. Qadir, S. Yahaya, and A.T.S. Ho, "Surrey University Library for Forensic Analysis (SULFA) of video content", In: Proceedings of the IET Conference on Image Processing (IPR 2012), London, pp. 1-6, 2012, *doi:*〈10.1049/cp.2012.0422〉.

[7] L. D'Amiano, D. Cozzolino, G. Poggi, L. Verdoliva, "A PatchMatch-Based Dense-Field Algorithm for Video Copy–Move Detection and Localization, IEEE Trans. Circuits Syst. Video Technol. vol. 29 (3) (March 2019) 669–682, https://doi.org/ 10.1109/TCSVT.2018.2804768.

[8] J.H. Bappy, C. Simons, L. Nataraj, B.S. Manjunath, A.K. Roy-Chowdhury, "Hybrid LSTM and Encoder–Decoder Architecture for Detection of Image Forgeries," in, IEEE Trans. Image Process. vol. 28 (7) (July 2019) 3286–3300, https://doi.org/ 10.1109/TIP.2019.2895466.

[9] X. Jin, Z. He, Y. Wang, J. Yu, J. Xu, Towards general object-based video forgery detection via dual-stream networks and depth information embedding. Multimed. Tools Appl. 81 (2022) 35733–35749, https://doi.org/10.1007/s11042-021-11126- 1.

[10] K. Kono, T. Yoshida, S. Ohshiro, and N. Babaguchi, "Passive Video Forgery Detection Considering Spatio-Temporal Consistency." Proceedings of the Tenth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2018), Springer International Publishing (2019), pp. 381-391, doi:10.1007/9 78-3-030-17065-3_38.

[11] S.K. Mohiuddin, S. Malakar, R. Sarkar, An ensemble approach to detect copy-move forgery in videos, Multimed. Tools Appl., 82(3) (2023) 3453–3474, https://doi. org/10.1007/s11042-023-14554-3.

[12] A. k. Venugopalan, and G.P. Gopakumar, "Copy-Move Forgery Detection - A Study and the Survey. In Proceedings of 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT), pp. 1327-1334, Kannur, India, 2022, doi: 〈10.1109/ICICICT54557.2022.9917647〉.

[13] K.M. Hosny, H.M. Hamza, and N.A. Lashin, Copy-move forgery detection of duplicated objects using accurate PCET moments and morphological operators, The Imaging Science Journal 66(6), 330-345, DOI: 10.1080/13682199.2018.1 461345.

Power System Technology
ISSN:1000-3673

Received: 16-03-2025 Revised: 05-04-2025 Accepted: 02-05-2025

[14] K.M. Hosny, H.M. Hamza, and N.A. Lashin, “Copy-for-duplication forgery detection in colour images using QPCETMs and sub-image approach”, IET Image Processing 13 (9), pp.1437-1446, https://doi.org/10.1049/iet-ipr.2018.5356.

[15] K.M. Hosny, A.M. Mortda, M.M. Fouda, N.A. Lashin, An Efficient CNN Model to Detect Copy-Move Image Forgery, IEEE Access vol. 10 (2022) 48622–48632, https://doi.org/10.1109/ACCESS.2022.3172273.

[16] A. Abozeid, R. Alanazi, A. Elhadad, A.I. Taloba, R.M. Abd El-Aziz, A Large-Scale Dataset and Deep Learning Model for Detecting and Counting Olive Trees in Satellite Imagery (Article ID), Comput. Intell. Neurosci. 2022 (2022) 1549842, https://doi.org/10.1155/2022/1549842.

[17] A. Abozeid, R. Alanazi, A. Elhadad, A.I. Taloba, R.M. Abd El-Aziz, An Efficient Indoor Localization Based on Deep Attention Learning Model, Comput. Syst. Sci. Eng. 46 (2) (2023) 2637–2650, https://doi.org/10.32604/csse.2023.037761.

[18] R. Yamashita, M. Nishio, R.K.G. Do, K. Togashi, Convolutional neural networks: an overview and application in radiology, Insights Imaging 9 (4) (2018) 611–629, https://doi.org/10.1007/s13244-018-0639-9.

[19] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, et al., CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope, Electronics 10 (20) (2021) 2470, https://doi.org/10.3390/electronics10202470.

[20] X. Han, Z. Hu, S. Wang, Y. Zhang, A Survey on Deep Learning in COVID-19 Diagnosis, J. Imaging 9 (1) (2023) 1–32, https://doi.org/10.3390/ jimaging9010001.

[21] S. Modiuddin, S. Malakar, M. Kumar, R. Sarkar, “Acomprehensive survey on state- of-the-art video forgery detection techniques, Multimed. Tools Appl., 82 (2023) 33499–33539, https://doi.org/10.1007/s11042-023-14870-8.

[22] D. D’Avino, D. Cozzolino, G. Poggi, L. Verdoliva, "Autoencoder with recurrent neural networks for video forgery detection" in Proc. IS&T Int’l. Symp. on Electronic Imaging: Media Watermarking, Security, and Forensics, 2017, pp 92 - 99, https://doi.org/10.2352/ISSN.2470-1173.2017.7.MWSF-330.

[23] H. Li and J. Huang, "Localization of Deep Inpainting Using High-Pass Fully Convolutional Network," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 8300-8309, doi: 〈10.1109/ICCV.2019.00839〉.

[24] J. Zhong, C. Pun, Y. Gan, “Dense moment feature index and best match algorithms for video copy-move forgery detection, Inf. Sci. Volume 537 (2020) 184–202, https://doi.org/10.1016/j.ins.2020.05.134.

[25] J. Zhong, Y. Gan, C. Vong, J. Yang, J. Zhao, J. Luo, Effective and efficient pixel- level detection for diverse video copy-move forgery types, in: Pattern Recognition, Volume 122, Elsevier, 2022 108286, https://doi.org/10.1016/j. patcog.2021.108286.