.

CHAPTER 5

ENHANCING VIDEO ANOMALY DETECTION WITH IMPROVED UNET AND CASCADE SLIDING WINDOW TECHNIQUE

5.1 INTRODUCTION

The analysis of video sequence for an efficient Video Anomaly Detection (VAD) requires processing of spatial and temporal information. Even though the model discussed in Chapter 4 effectively identifies anomalous events, the model's performance is limited to overfitting problem. In order to resolve the overfitting issue, the video sequences are preprocessed using Wiener filter and a feature extraction model focusing on image segmentation is suggested in Chapter 5. The model utilized a hybrid methodology using an improved U-Shaped Network (UNet) along with Cascade Sliding Window Technique (CSWT). This hybrid model successfully calculates the Anomaly Score (AS) from multiple frames and depending on the AS the model classifies the input video into normal or anomalous.

5.2 UNET ARCHITECTURE FOR SEGMENTATION

The UNet is an image segmentation model in Deep Learning (DL). The model contains a contracting (encoder) and expanding (decoder) path. The basic UNet architecture is displayed in Figure 5.1 where the arrows represent the flow of processing.

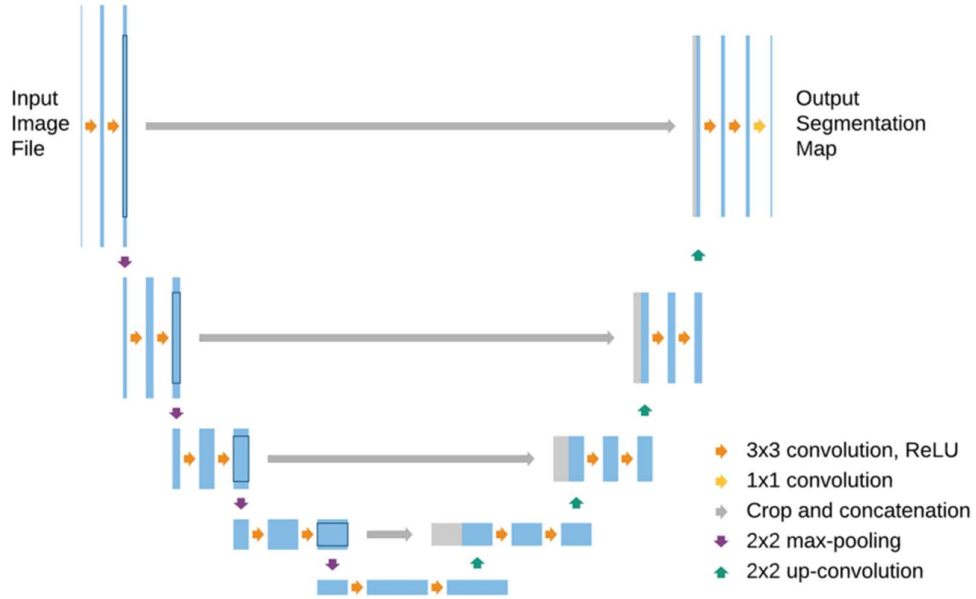

Figure 5.1 UNet Architecture (Siddique et al., 2021)

The blue colored box represents feature mapping and the grey color denotes the feature maps from the contracting path. The contracting path extracts features while reducing spatial dimensions using consecutive *3×3* convolutions with ReLU activation, trailed by max-pooling for down sampling. The expansive path reconstructs the image resolution and localizes features by combining up-convolution layers with cropped features from the encoder through skip connections. This process integrates high-resolution spatial details with contextual information for improved segmentation accuracy. The final output layer uses a *1×1* convolution to generate the segmented image with the desired amount of classes. UNet's design ensures accurate feature extraction, localization and segmentation by combining hierarchical feature learning and precise spatial information.

5.3 CONVOLUTIONAL LSTM (ConvLSTM)

Convolutional Long Short-Term Memory (ConvLSTM) is an improved Fully Connected LSTM(FC-LSTM) designed to model spatiotemporal dependencies more effectively (Shi et al., 2015). Unlike FC-LSTM, which process inputs as one-dimensional vectors, ConvLSTM represents them as three-dimensional tensors, preserving spatial relationships. By integrating convolutional processes in input-to-state and state-to-state transitions, it efficiently captures spatial correlations while maintaining the capability to learn long-term temporal dependencies. ConvLSTM employs a gated structure with input, forget and output gates, replacing traditional matrix multiplications with convolutions to enhance computational efficiency and reduce redundant connections. It follows an encoder-forecaster architecture, where the encoder extracts meaningful spatiotemporal features and the forecaster predicts future sequences. This structured learning approach allows the model to distinguish complex motion patterns, enabling it compatible for applications like video analysis, anomaly detection and activity recognition. By bridging convolutional feature extraction and sequential modeling, ConvLSTM provides a robust feature extraction and sequential modeling, ConvLSTM provides a robust framework for processing dynamic spatiotemporal data.

The Equations (5.1) to (5.6) provides a summary of the ConvLSTM unit's formulation.

Forget Gate (f): Decides which part of previous memory to retain.

$$f_t = \sigma(w_f * (h_{i-1}, x_t, C_{t-1}) + b_f \tag{5.1}$$

— x_t *:* Input at the ongoing time step.

— h_{t-1}*:* Hidden state from the preceding time step.

— C_{t-1}*:* Previous cell state storing long-term memory.

— W_f : Learnable convolutional kernels for the forget gate.

— b_f : Bias for the forget gates.

— σ : Sigmoid activation function, producing values between 0 and 1 to regulate information flow.

— f_t : Forget gate activation, determining how much of C_{t-1} is retained.

Input Gate *(i_t):* Regulates how much new data to incorporate.

$$i_t = \sigma(w_i * (h_{i-1}, x_t, C_{t-1}) + b_i \tag{5.2}$$

— i_t : Input gate activation, controlling new information storage.

— W_i Learnable convolutional kernels for the input gate.

— b_i : Bias terms for the input gate.

Candidate Cell State (C'_t): Generates new information.

$$C'_t = tanh\ (W_c * (h_{t-1}, x_t) + b_c \tag{5.3}$$

— *tanh*: Hyperbolic tangent function, mapping values between -1 and 1 for memory updates.

— C_t' : Candidate cell state, representing new information to be added.

— W_c: Learnable convolutional kernels for the candidate gate.

— b_c : Bias terms for the candidate gate.

Cell State Update (C_t): Merges old and new memory.

$$C_t = f_t \times C_{t-1} + i_t \times C'_t \tag{5.4}$$

— C_t : Updated cell state, combining past and new information.

Output Gate *(o_t):* Controls what information flows to the next time step.

$$o_t = \sigma(w_o * (h_{i-1}, x_t, C_{t-1}) + b_0 \tag{5.5}$$

— W_o: Learnable convolutional kernels for the output gates.

— b_o : Bias terms for the gates.

Hidden State Update *(h_t):* Generates the new hidden state, influencing future predictions.

$$h_t = o_t \times tan\, h(C_t) \tag{5.6}$$

This formulation allows ConvLSTM to efficiently capture spatial and temporal dependencies in video sequences, making it ideal for VAD and other spatiotemporal tasks.

5.4 CASCADE SLIDING WINDOW TECHNIQUE

The Cascade Sliding Window Technique (CSWT) is an object detection method adapted for VAD to identify anomalous behaviors or events in video streams. It operates through a multi-stage process, scanning frames at various locations and scales using a sliding window to detect irregularities. At each stage, a classifier analyzes the data within the window to classify normal and anomalous patterns.

The cascade structure consists of multiple layers of classifiers. During the initial stages simpler and faster classifiers are used to quickly filter out non-anomalous area there by reducing the computational load. And in the later stages a more sophisticated classifier is used in order to refine the anomaly score of the VAD and for improved Precision.

The Anomaly Score (AS) generated using CSWT represents the probability of anomaly in a specific frame. The frames containing anomalous behavior are identified and analyzed based on the thresholds of the AS. The CSWT enhances the accuracy and efficiency of Anomaly Detection (AD), offering a structured VAD model.

The cascade approach is utilized according to the Algorithm 5.1 that apply a progressive increase of AS and can optimize detection based for the enhanced classification. This helps to filter out the non-anomalous data in the earlier stages, to maintain reliable and precise anomaly identification in later stages.

Input: Actual present frame F_actual and F_anticipated, frame size R, window size R, window decrease size v.

Result: score S for anomalies.

Algorithm 5.1: Pseudocode for CSWT

Step 1. Initialize Coordinates:

- Set x = 0 and y = 0.

Step 2. Calculate Initial Difference Image:

- Compute the square of the differences between F_{actual} and $F_{anticipated}$ to get I.

Step 3. Slide Window Vertically:

- While y is less than R:
  - Check whether the window fits within the frame height:
  - If $y+R_{window}$ is less than or equal to R:
    - **Slide Window Horizontally:**
      - While x is less than R:
        - Check if the window fits within the frame width:
          - If $x+R_{window}$ is less than or equal to R:
          - Calculate the average intensity of the current window.
          - Else:
            - Calculate the average intensity for the window at the edge.
        - Move the window horizontally by incrementing x by R_{window}.
  - Else:
    - **Slide Window Horizontally:**
      - While x is less than R:
        - Check if the window fits within the frame width:
          - If $x + R_{window}$ is less than or equal to R:
            - Compute the average intensity for the window at the edge.
          - Else:
            - Compute the average intensity for the window at the corner.
        - Move the window horizontally by incrementing x by $R_{window.}$
  - Reset x to 0.
  - Move the window vertically by incrementing y by $R_{window.}$
  - Decrease R_{window} by v.

Step 4. Arrange Anomaly Scores:

- Sort the calculated window scores in ascending order.

Step 5. Compute Final Anomaly Score:

- Calculate the average of the sorted window scores to get the final AS S.

Step 6. Return Anomaly Score:

- Output the AS S.

In Algorithm 5.1, the parameter represents the height and width of the frame, while R_{window} specifies the sizes of the sliding window. The process begins by computing a different image *I*, which is obtained by squaring the pixel-wise differences among the actual frame F_{actual} and the anticipated frame $F_{anticipated}$. This approach proves advantageous as it amplifies pixel intensities in anomalous regions of *I*, enhancing the detection of anomalies compared to using a simple difference between the actual and anticipated frames.

The sliding window process begins at coordinates $x=0$ and $y=0$, with the window moving right by R_{window} until it reaches the edge of the frame. When the window reaches the edge, it moves down by R_{window} and the window size R_{window} is decreased by *v*. If the window cannot fully fit within the frame due to the remaining space being less than R_{window}, it adjusts to fit the available space, either along the right side or at the top corner of *I.*

The average intensity P_k is computed for each window during the sliding window process. This value is similar to the Mean Squared Error (MSE) calculated among the actual frame and the predicted frame. Once the sliding window has traversed the entire frame, the P_k scores are arranged in growing order. The final AS *S* is then derived by averaging the top *n* scores from the sorted list. The decreasing window size *v*, as shown in Algorithm 5.1, is essential as it accounts for the diminishing size of objects as they move further from the camera.

5.4.1 Anomaly Detection (AD)

The CSWT is utilized to determine the AS for each frame. In the output of the model, the AS spans from 0 to a maximum value, with the framework's color depth configured at 257. This results in the anomaly scores ranging over a wide range of values, making it unsuitable for setting a consistent threshold for anomalous frames. To address this issue, the

anomaly scores are normalized to a range between 0 and 1, ensuring consistency and suitability for thresholding using the Equation (5.7).

$$S'(t) = 1 - \frac{s(t) - min_t s(t)}{max_t s(t) - min_t s(t)'} \tag{5.7}$$

Where *S'(t)* is the normalized AS and *S(t)* is the AS for frame *t*. Videos have two AS: $min_t S(t)$ and $max_t S(t)$ for maximum and minimum AS, respectively. However, when acquiring a new frame in the real-world, the $max_t S(t)$ and $min_t S(t)$ values could change. This leads to a reevaluation of the threshold and the associated ASs calculated using Equation (5.8). Scaling the AS to a range of 0 to 1 effectively addresses this issue.

$$S'(t) = \frac{s(t)}{colordepth^2} \tag{5.8}$$

Here, color depth represents the color depth of the output frame. This normalization method eliminates the need to recalculate the threshold and anomaly scores, even if the maximum and minimum ASs vary.

5.5 IMPROVED UNET-CSWT MODEL

VAD and segmentation are computationally demanding, often resulting in inefficiencies, reduced accuracy and overfitting due to their reliance on precise training data. The UNet architecture addresses these challenges effectively by handling datasets and delivering precise results. The architecture of Improved UNet-CSWT (IUNet-CSWT) is given in Figure 5.2.

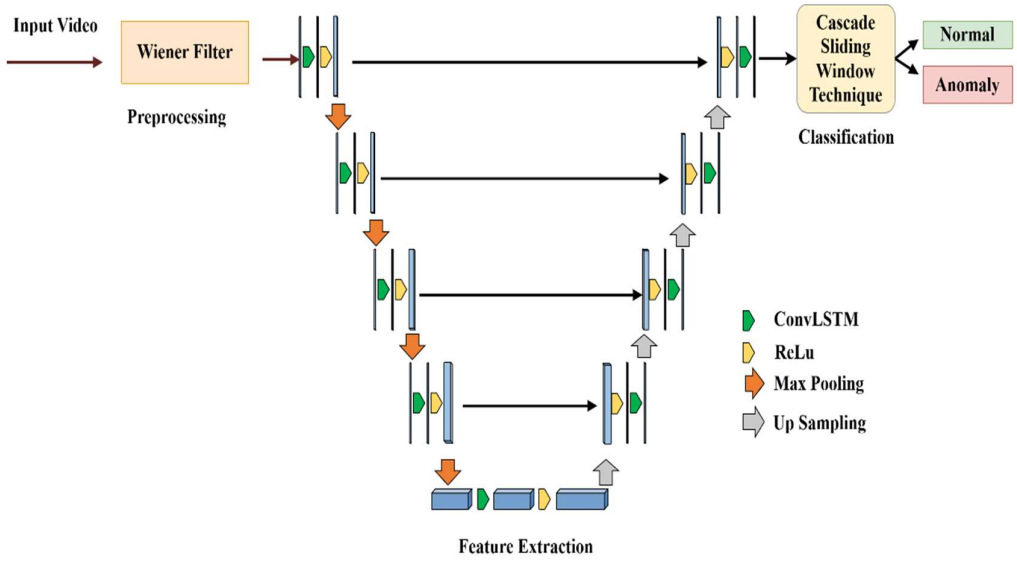

Figure 5.2 Architecture of IUNet - CSWT Model

The Improved-UNet (IUNet) enhances anomaly detection in video frames by integrating a contracting path (encoder) for capturing temporal context and performing initial segmentation, along with an expanding path (decoder) for refining segmentation and restoring spatial resolution.

The input UCSD video is transformed into frames and is further preprocessed using a Wiener filter for noise reduction. The features are identified using an IUNet model and CSWT calculated the Anomaly Score. Depending upon summation of the Anomaly Score the IUNet-CSWT model classifies the video input.

- **Preprocessing**

The presence of noise contents in the UCSD dataset will results in the Precision of the VAD model. The noise in the video is occurs due to the environmental variations, background clutters, subtle anomalies, artifacts and fixed camera setup. The presence of noise in the input video complicated the accurate VAD and also reduced the performance of the detection algorithms. To rectify this issue, the Wiener filter is utilized to each frame during preprocessing and there by filter out the noise. This step enhances the input data and improves the effectiveness of the VAD.

- **Wiener Filter**

The Wiener filter (Dos Santos et al., 2020) utilizes Linear Time-Invariant (LTI) filtering to evaluate a random process from a noisy observation, assuming consistent signal and noise spectrum with known additive noise. The Wiener filter generates an output approximation by statistically filtering a similar input signal, minimizing the Mean Square Error (MSE) among the assessed and actual random processes. As an adaptive filter, it adjusts the degree of smoothing based on local variation. It applies more smoothing in regions of low variation and less in areas of high variation, using the neighborhood's mean and variance for computation. The filter minimizes the variance comparing the actual and estimated signals. The Equation (5.9) represents the error measure between an original image and a processed image:

$$e^2 = E\{(f - f')^2\} \tag{5.9}$$

where,

— Error Measure *(e^2)* – Represents the squared error between the original image and the processed image.

— Original Image *(f)* – The input image before any processing.

— Processed Image (*f'*) – The output image after processing.

— E {.} symbolizes the predicted value of the parameter.

As a result, minimizing the quadratic error function is essential for generating an approximated image. This is accomplished in the frequency domain, under the assumptions that the intensity levels in the predicted image are reduced by a linear function, the image and noise have zero mean and there is no correlation between them. The error function reaches its minimum under specific conditions and is represented in Equation (5.10).

$$F(u,v) = \left[\frac{H*(u,v)s_f(u,v)}{s_f(u,v)|H(u,v)|^2+s_f(u,v)}\right] G(u,v) \tag{5.10}$$

where:

— $F(u,v)$ represent the predictable image in the frequency domain.

— $H(u,v)$ is the transformation of the degradation function.

— $G(u,v)$ denotes the transformation of the corrupted image.

— $H^*(u,v)$ is the complex conjugate of $H(u,v)$.

— $S_f(u,v) = |F(u,v)|^2$, the power spectrum of the original, non-degraded image.

The general principle of the filter relies on the magnitude squared of a complex value, derived from multiplying the value by its conjugate. This results in an alternate form as given in Equation (5.11).

$$F(u,v) = \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2+S_n(u,v)/s_f(u,v)}\right] G(u,v) \tag{5.11}$$

where:

— $S_n(u,v) = |N(u,v)|^2$ represents the power spectrum of noise.

— The ratio $S_n(u,v)/ S_f(u,v)$ is often substituted by a constant K, as the power spectrum of the non-degraded image is rarely known.

The Wiener filter helps reduce noise in digital image processing caused by continuous power additive noise. The key parameters of the Wiener filter are the neighborhood size and noise power.

- **Feature Extraction**

The preprocessed image is provided to the Improved-UNet, where ConvLSTM layers are used instead of traditional convolutional networks within the U-Net framework to detect abnormalities in the denoised video frames, enhancing detection Precision. This approach

leverages spatial and temporal information for anomaly detection. The fusion of ConvLSTM into the model allows for the analysis of spatial features within discrete video frames and temporal relationships between successive frames, ensuring a comprehensive understanding of the content.

- **Classification**

In the postprocessing stage, the model employs the CSWT to compute an AS. After analyzing the video frames, the CSWT allocates a score that reflects the likelihood of an abnormality. This score is then utilized to assess whether a given frame exhibits an anomaly or remains in normal range. The pseudocode for the model is presented in Algorithm 5.2.

Algorithm 5.2 Pseudocode for IUNet - CSWT model

Step 1: **Input video and convert it into frames**

Step 2: **Preprocessing**

Load video frames

Apply Wiener filter for noise removal

Use a wiener filter to denoise each frame.

Save the denoised frames for further processing.

Step 3:Initialize and configure the model

Initialize IUNet with ConvLSTM to enhance spatial and temporal feature learning.

Extract spatial features from each frame using the encoder.

Capture Temporal Information

Use ConvLSTM to identify the temporal dependencies among frames.

Reconstruct Frames to preserve spatial-temporal consistency

Use the decoder to reconstruct frames from the extracted features and temporal information.

Denoise and refine features

Denoise and refine features using the encoder to enhance representation.

Step 4: Anomaly Scoring

Compute Anomaly Scores

Apply the CSWT to compute an AS for each reconstructed frame.

Step 5: Anomaly Detection

Classify Frames as Normal or Anomalous

Compare the anomaly scores against a predefined threshold to classify frames.

5.6 RESULTS AND DISCUSSIONS

The proposed model, Spatial-Temporal Anomaly Identification with IUNet, effectively addresses the challenge of overfitting in AD within video frames. This segment presents the results of the IUNet-CSWT model. Figure 5.3 illustrates the original and preprocessed images, where the Wiener filter is applied during preprocessing to remove noise from the frames.

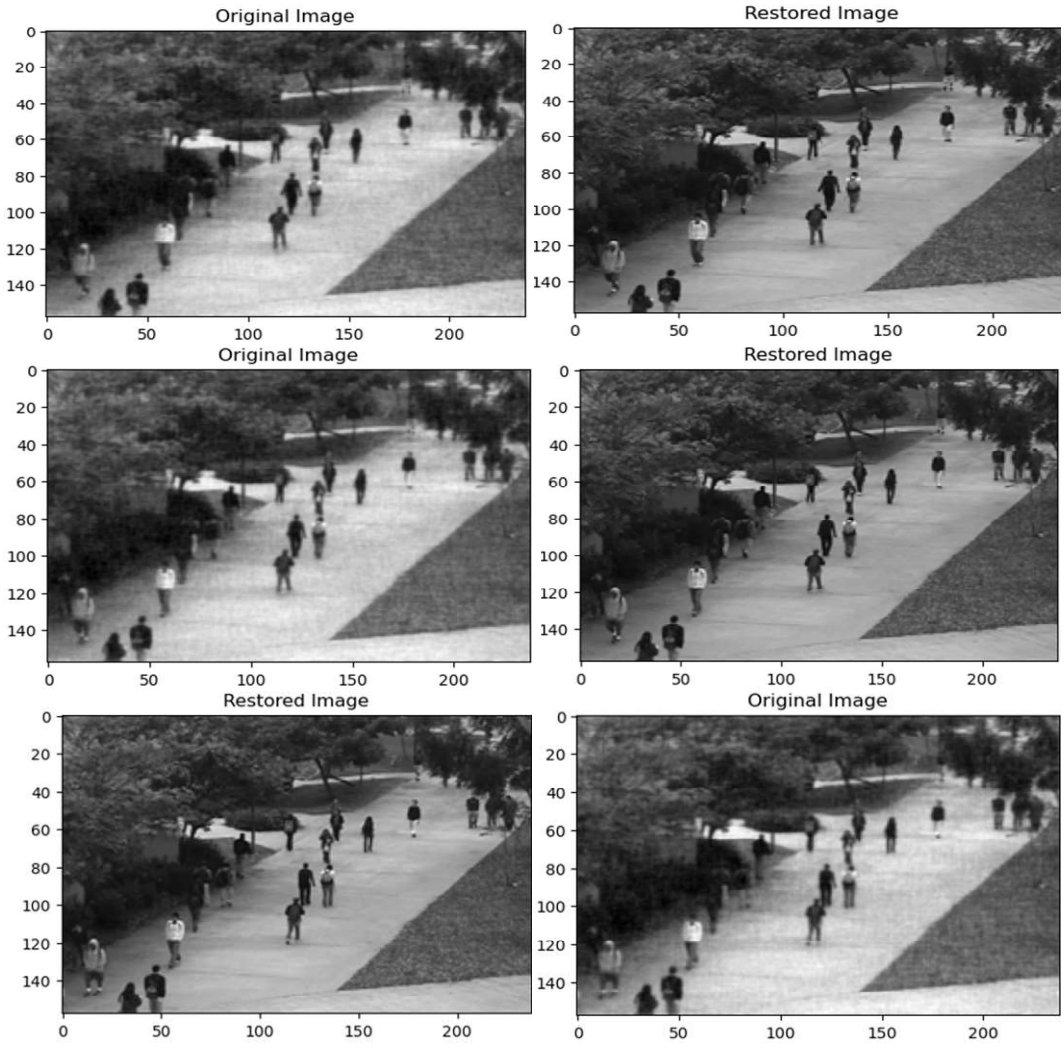

Figure 5.3 Original and Pre-Processed Images

The Figure 5.3 presents a comparison between original and restored surveillance images, demonstrating the effectiveness of the Wiener filter in noise removal and image restoration, enhancing clarity for improved anomaly detection. Figure 5.4 highlights the classification of Video Anomaly Detection (VAD) into normal and anomalous categories.

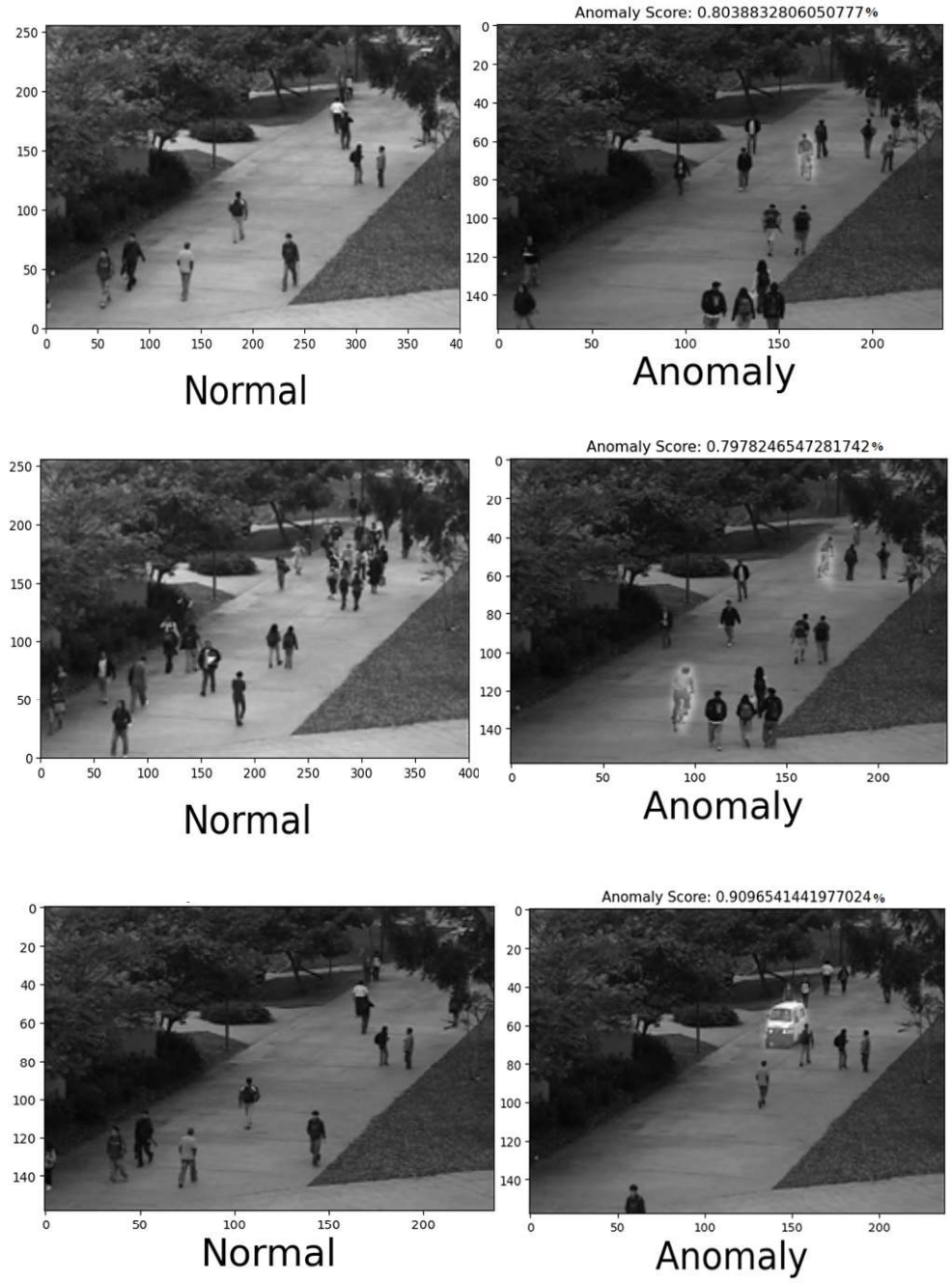

Figure 5.4 Results of Normal and Anomalous Image

The Figure 5.4 compares normal and anomalous events in a surveillance scene, where normal events depict typical pedestrian movement. The anomalous events are highlighted with higher anomaly scores. These anomalies include unusual activities such as a vehicle on the pedestrian walkway and a boy riding a bicycle in an area designated for pedestrians.

5.6.1 Performance Evaluation

This section highlights how the proposed approach accurately classifies the input frames as normal or anomalous. Figure 5.5 illustrates the training process of the IUNet-CSWT model over twenty-five epochs.

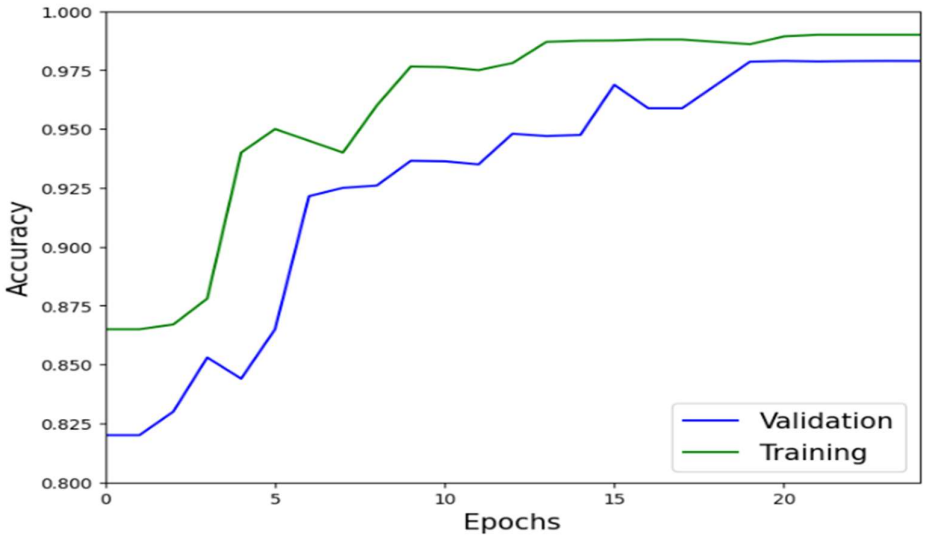

Figure 5.5 Training and Validation Accuracy

The outcomes indicate that the model grabs high accuracy while maintaining effectiveness, with rapid convergence during training. The model demonstrates its ability to mitigate overfitting, achieving improved generalization and maintaining accuracy across approximately 1% of the data. The model utilized the Adam optimizer to achieve a consistent 99% accuracy. The Adam optimizer, a widely used tool in deep learning models, enhances optimization by adapting learning rates for individual parameters, effectively managing complex training dynamics and ensuring efficient convergence. This exceptional level of accuracy underscores the reliability and efficiency of the IUNet-CSWT in accurately detecting and analyzing data patterns. The model's performance highlights its capability or tasks requiring high Precision and dependability.

Figure 5.6 demonstrates the training process of the IUNet-CSWT model using the Adam optimizer, with validation and training loss metrics tracked over 25 epochs. From the initial to the eighth epoch, the model consistently decreases loss levels, highlighting its ability to minimize discrepancies among predicted outputs and actual targets. Achieving an average loss of approximately 1.53%, the model demonstrated high prediction accuracy. This effective reduction in loss is likely attributed to the adaptive learning rates and parameter updates facilitated by the Adam optimizer, enabling the model to converge efficiently to an optimal solution.

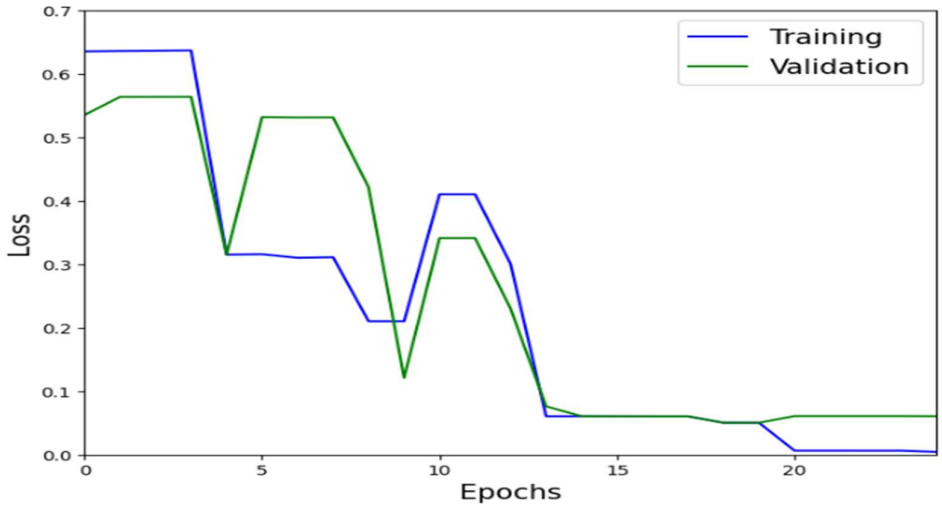

Figure 5.6 Training and Validation Loss

Table 5.1 offers the performance metrics and anomaly scores for the proposed VAD using IUNet - CSWT model with respect of Accuracy, Recall, Precision, F1 Score, AUC, PSNR and EER. These values underscore the model's capability in detecting video anomalies.

Table 5.1 Performance of the IUNet - CSWT Model

Performance Metrics	IUNet - CSWT
Accuracy	99 (%)
Precision	97.3 (%)
Recall	97.5 (%)
F1 Score	97.5 (%)
AUC	0.908
PSNR	35.04 (dB)
EER	10.9 (%)

Figures 5.7 to 5.13 provides a detailed performance comparison of IUNet - CSWT model. Figure 5.7 compares the accuracy, demonstrating that IUNet-CSWT attains 99% of accuracy, while ResNet-LSTM achieves 96.5%. IUNet-CSWT outperforms ResNet-LSTM by 2.59%, showcasing its superior classification capability in VAD.

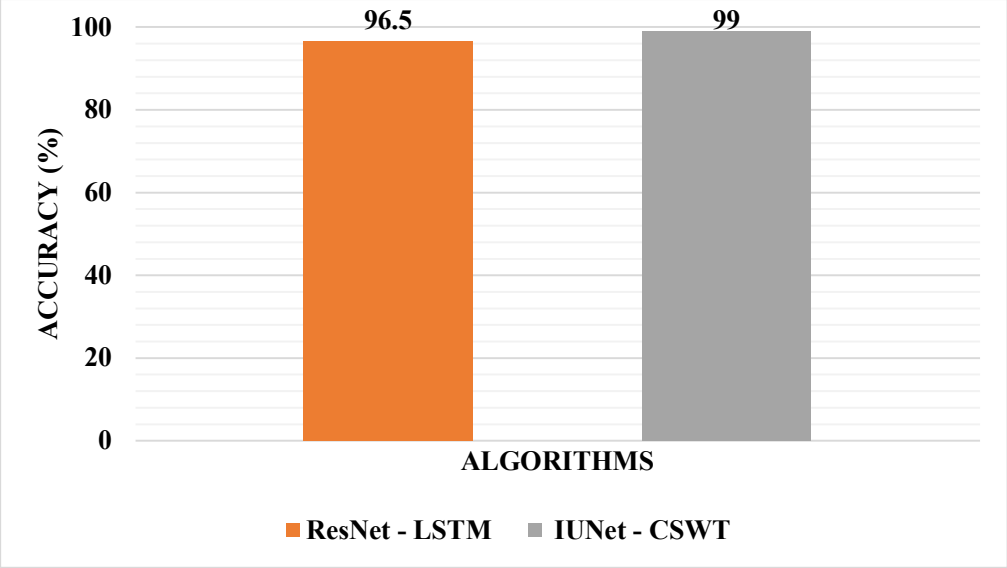

Figure 5.7 Performance Comparison of Accuracy

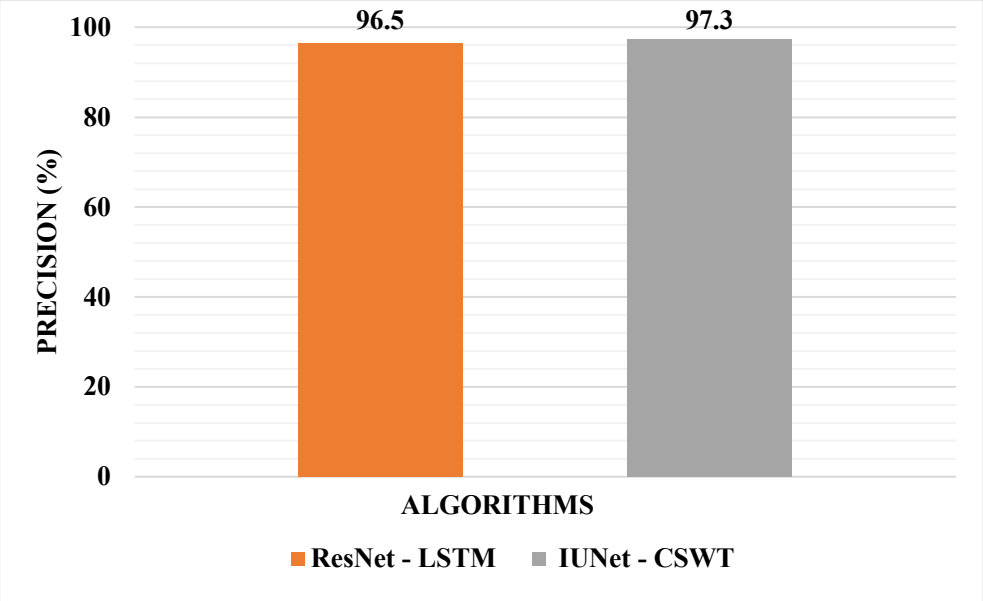

Figure 5.8 Performance Comparison of Precision

Figure 5.8 illustrates the Precision, where IUNet-CSWT achieves 97.3%, surpassing ResNet-LSTM at 96.5%. The 0.8% improvement highlights the model's enhanced capability to suitably detect anomalies while reducing false positives.

Figure 5.9 and Figure 5.10 presents the performance of Recall and F1 Score, with IUNet-CSWT attaining 97.5%, exceeding ResNet-LSTM, which records 96.5%. This improvement of 1% indicates a higher success rate in detecting true anomalies. This high Recall and F1 Score ensures a high detection rate, minimizing the risk of false negatives.

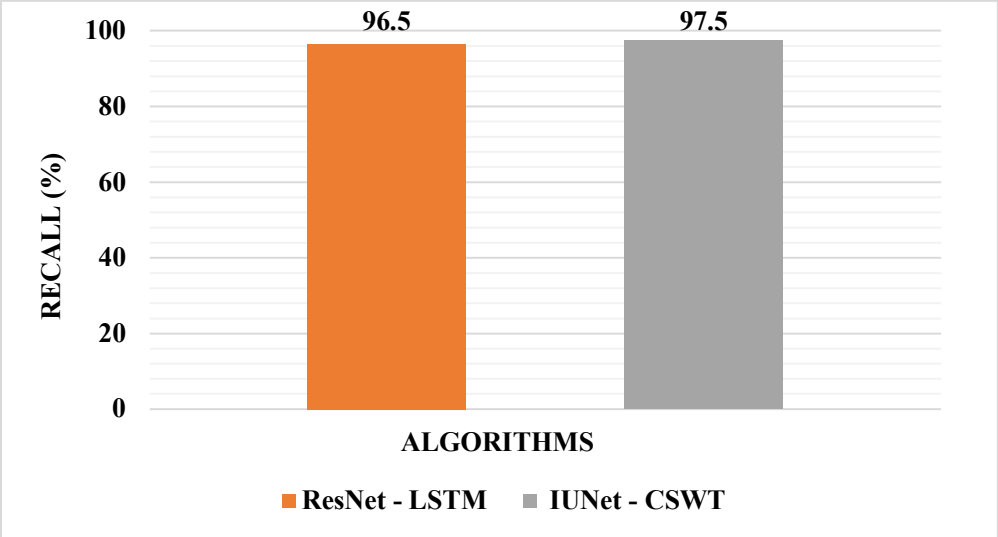

Figure 5.9 Performance Comparison of Recall

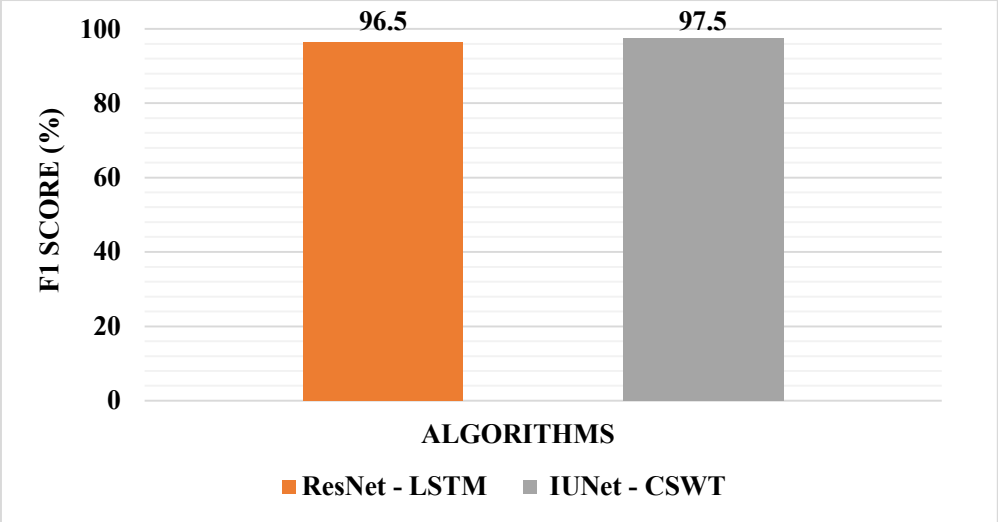

Figure 5.10 Performance Comparison of F1 Score

Figure 5.11 compares the AUC, showing that ResNet-LSTM achieves 0.99, while IUNet-CSWT records 0.908. The 0.082 lower AUC indicates that IUNet-CSWT has a slightly reduced capability to distinguish between normal and anomalous events. However, the AUC value of 0.908, as shown in Figure 5.11, demonstrates the model's strong capability to differentiate anomalies from normal behavior. A high AUC signifies that the model is well-calibrated, ensuring reliable performance in Anomaly Detection (AD) tasks.

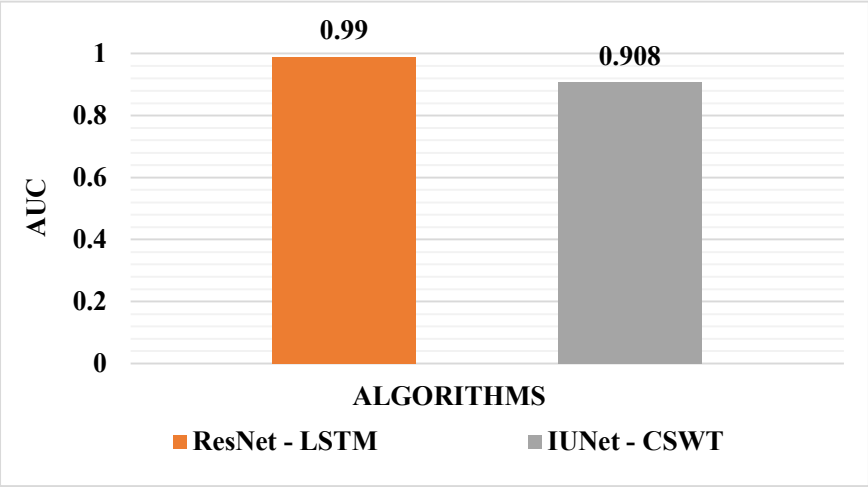

Figure 5.11 Performance Comparison of AUC

Figure 5.12 evaluates PSNR, with IUNet-CSWT attaining 35.04dB, compared to 28.55dB for ResNet-LSTM. The 6.49dB improvement indicates superior video reconstruction quality, essential for anomaly detection.

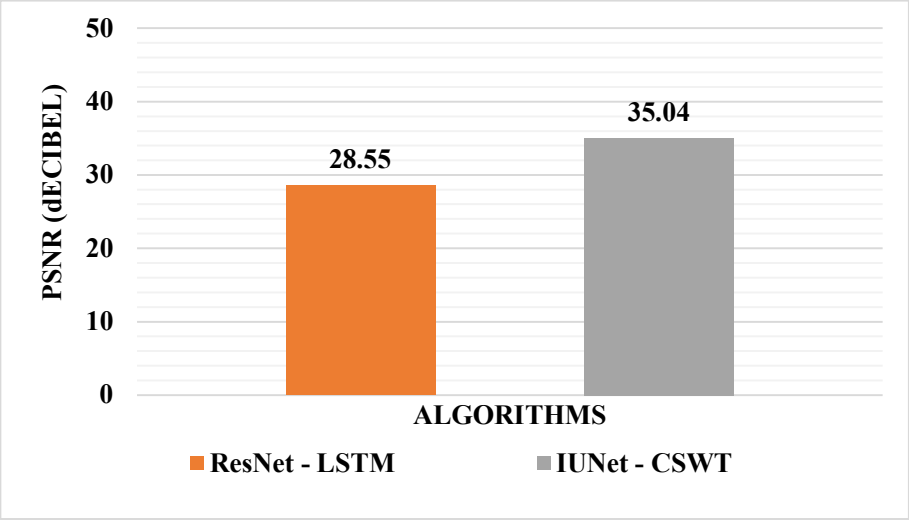

Figure 5.12 Performance Comparison of PSNR

The PSNR of 35.04dB highlights the quality of the reconstructed images and videos. Higher PSNR values indicate minimal distortion, ensuring better representation of detected anomalies. Figure 5.13 highlights the Equal Error Rate (EER), where IUNet-CSWT achieves 10.9%, outperforming ResNet-LSTM at 14.5%. The fewer misclassifications and increased overall reliability are provided and is identified using 3.6% of reduction in EER.

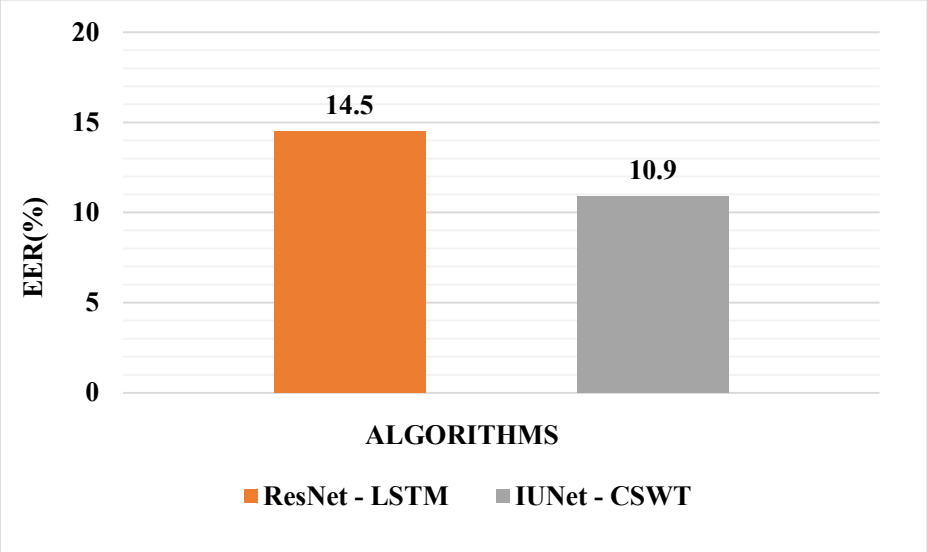

Figure 5.13 Performance Comparison of EER

The lower EER of 10.9% indicate the IUNet-CSWT models capability of balancing between false positive and false negative rates. The EER value specifies the model's ability to obtain enhanced overall performance by manging the misclassification.

The efficiency of the IUNet-CSWT model is evaluated using various performance metrics and these measures analyze the model's ability to deliver accurate classification results and attain benchmark performance. The improved encoder-decoder using ConvLSTM mechanism attain improved accuracy for the model and helps to capture the spatial and temporal features effectively. The anomalies in the video sequences over time are accurately detected by analyzing the complex patterns using ConvLSTM. The performance of the model is enhanced by the use of CSWT that can generate the anomaly scores of the individual frames needed for the determination of anomalous events. The CSWT analyzes the consecutive frames and thereby enhances the VAD model in AD along with the ConvLSTM. The spatial and temporal relations are combined effectively and the IUNet-CSWT model enhances the anomaly prediction using the addition of anomaly scores.

5.7 SUMMARY

The IUNet-CSWT model discussed in this chapter provides a remedy for the overfitting challenges in VAD. The noise removal of the input video is performed using Wiener filter during preprocessing. The replacement of convolution layer with ConvLSTM layer in the UNet model effectively gather the spatial and temporal information and thereby enhances feature extraction capability of the IUNET-CSWT model. The CSWT calculated the anomaly scores in sequential video and there by enhances the classification capability of the model. The Accuracy of 99%, AUC of 0.908 and EER of 10.09% proves the model's efficiency in identifying anomalous events and high precision explains the effectiveness of the model.