CHAPTER 6

HYBRID ANOMALY DETECTION WITH HIERARCHICAL MULTISCALE CNN FUSION WITH LSTM

6.1 INTRODUCTION

Existing video anomaly detection (VAD) methods have limitations in low image resolution, noise interference, computational facilities, overfitting and diminishing reliability and accuracy. To resolve these limitations a hybrid model comprises Hierarchical Fusion of Multiscale-Convolutional Neural Network (M-CNN) and Long Short-Term Memory (LSTM). Noise reduction can be eliminated by Bilateral-Wave Denoise Technique. To enhance the feature extraction by reducing overfitting uses a Spatial Pyramid Pooling (SPP) with the Hierarchical M-CNN. The accurate classification is obtained by using temporal pattern recognition using LSTM along with Global Average Pooling (GAP).

6.2 BILATERAL-WAVE DENOISE TECHNIQUE

The Bilateral wave Denoise technique (Zhang & Gunturk, 2008) integrates the bilateral filter with the wavelet transform to enhance image denoising efficiency, Wavelet thresholding in the wavelet domain primarily identifies and processes the noise components in an image. Applying a threshold to the image coefficients in the wavelet domain produces an optimal estimate of the input image with reduced noise.

The fundamental steps in wavelet thresholding for image denoising are as follows:

- **Wavelet Decomposition:** The noisy input image is decomposed to extract wavelet coefficients, as expressed in Equation (6.1):

$$w=W(X+N) \tag{6.1}$$

where:

— X is the original noise-free image.

— N represents the noise component affecting the image.

— W is the wavelet transform function, responsible for decomposing the image into different frequency components.

— w denotes the computed wavelet coefficients, which capture both signal and noise information.

- **Wavelet Thresholding:** The coefficients are processed using a wavelet thresholding rule to obtain an optimal estimate, as given in Equation (6.2):

$$W' = \delta_\lambda(w) \tag{6.2}$$

where:

— *W'* represents the refined wavelet coefficients after thresholding.

— $\delta_\lambda(w)$ is the wavelet thresholding function that selectively removes noise components.

— λ is the threshold value, determining which coefficients are suppressed or retained.

- **Reconstruction:** The denoised image is attained by applying the inverse wavelet transform on the refined coefficients, as shown in Equation (6.3):

$$X' = W^{-1}w' \tag{6.3}$$

where:

— *X'* is the final denoised image.

— W^{-1} is the inverse wavelet transform, reconstructing the image from the threshold coefficients.

Wavelet transform is highly effective in reducing noise in low-resolution images by decomposing them into different frequency components, allowing selective noise suppression.

- **Bilateral Filtering**

Bilateral filtering for noise reduction smooths images while preserving edges, making it widely applicable in image denoising and enhancement. The bilateral filter operates by replacing every pixel with a weighted average of its neighbors, where the weights rely on both spatial distance and photometric similarity. The Equation (6.4) depicts the combined approach of bilateral filtering and wavelet denoising.

$$h(x) = \frac{1}{k(x)} \iint_{-\infty}^{\infty} W^{-1}\delta_\lambda\left(W\left(f(\xi)\right)\right) c(\xi, x)s\left(f(\xi), f(x)\right)d\xi \tag{6.4}$$

where:

— $h(x)$ is the output filtered intensity at pixel x.

— $\left(W\left(f(\xi)\right)\right)$ represents the wavelet-transformed intensity at pixel ξ.

— $\delta_\lambda\left(W\left(f(\xi)\right)\right)$ applies wavelet thresholding to remove noise.

— W^{-1} is the inverse wavelet transform reconstructing the denoised image.

— $c(\xi, x)$ is the spatial weighting function based on geometric proximity.

— $s\left(f(\xi), f(x)\right)$ is the range weighting function based on intensity similarity.

— $k(x)$ is the normalization constant ensuring the sum of weights is 1.

The bilateral filtering effectively suppresses noise while retaining edge details. In smooth areas, the bilateral filter operates much like a conventional domain filter, averaging minor, weakly correlated differences in pixel values caused by noise, where pixels within a small neighborhood share similar values.

6.3 HIERARCHICAL MULTISCALE–CNN (M–CNN)

The Hierarchical M–CNN architecture is designed for image processing tasks, incorporating four convolutional blocks that extract deep features while mitigating the vanishing gradient problem. This structure enhances its effectiveness in complex environments. The model removes the final two fully connected layers to optimize performance and reorganizes them into two distinct components. The first component, the Feature Extraction Network, integrates the initial three convolutional blocks to capture essential features. The second component, the Anomaly Detection Block, processes these extracted features and employs Spatial Pyramid Pooling (SPP) to accommodate varying input dimensions, thereby improving anomaly detection accuracy. Figure 6.1 illustrates the architecture of Hierarchical M–CNN.

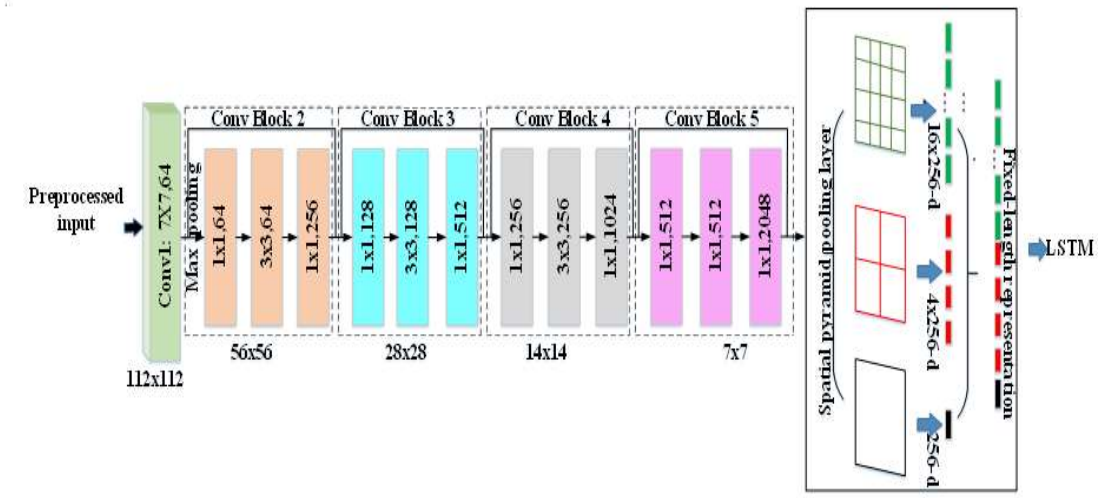

Figure 6.1 Architecture of Hierarchical M-CNN

- **Convolutional Structure**

The architecture begins with the first Convolutional layer (Conv1), which processes 512×512-pixel input images using a 7×7 kernel to generate 64 feature maps. A stride of 2 ensures down sampling of the feature maps. Subsequently, a max-pooling layer (Pool1) with a 3×3 window and stride of 2 maintains the 64 channels while reducing the spatial dimensions to 256×256. The network then undergoes a series of convolutional blocks, progressively increasing feature depth while reducing spatial dimensions:

- **Block 1:** Down samples to 128×128 with 256 channels.
- **Blocks 2 & 3:** Reduce dimensions to 64×64, with Block 2 reaching 512 channels, while Block 3 maintains this size but doubles the channels to 1024.
- **Block 4:** Further deepens the channels to 2048 with a stride of 2, resulting in 7×7 feature maps.

Each block contains multiple convolutional layers following a convolutional group structure, allowing the network to capture increasingly complex features. This hierarchical design efficiently extracts meaningful representations for downstream tasks such as object recognition or classification.

➢ **Atrous Convolution for Multiscale Feature Extraction**

Atrous convolution (or dilated convolution) is employed to control feature map filters effectively. It extends traditional convolutional methods, facilitating multiscale feature extraction. The nonlinear function ReLU is represented by Equation (6.5).

$$F = W_{2\sigma}(W_{1x}) \tag{6.5}$$

where,

— *F:* Non-linear function output after applying ReLU.

— W_1, W_2: Weight matrices used in the convolutional layers.

— σ: Non-linear activation function (ReLU).

— x: Input to the convolutional layer.

The output y is computed using a shortcut connection and a second ReLU and is expressed in Equation (6.6).

$$y = F(x, \{W_i\}) + x \tag{6.6}$$

Here,

— *y:* Output after applying residual learning.

— *x:* Input feature map.

— $F(x,\{W_i\})$: Function learned using weight set W_i.

— W_i: Weights of the network.

For a given input *x,* the network learns a hierarchical function representation, denoted by Equation (6.7).

$$y = F_{(x[W_i])} + W_S x \tag{6.7}$$

where,

— *y*: Output after residual connection.

— $F_{(x[Wi])}$: Function learned from input x and weights W_i.

— W_S: Scaling factor in the residual connection.

➢ **Feature Map Generation in a Unidimensional Signal**

Considering a unidimensional signal, the output feature map *y[a]* is determined by an input signal *x[a]* and a filter *q[k]* of length *k* is expressed by Equation (6.8):

$$y[a] = \sum_k x[a + r.k]q[k] \tag{6.8}$$

— *y[a]*: Output feature map at position *a*.

— *x[a]*: Input signal at position *a.*

— *q[k]*: Filter applied to the signal.

— *k*: Filter length.

— *r*: Atrous rate (controls dilation in atrous convolution).

The atrous rate (*r*) determines the stride of the sampling signal, modifying the filter's field of view. If *r=1*, the operation is standard convolution; for *r>1*, the receptive field expands, enabling multiscale feature extraction.

➢ **Optimization using Depthwise Separable Convolution**

To reduce computational complexity, the encoder module utilizes depthwise separable convolution, splitting standard convolution into:

1. **Depthwise convolution** – Performs spatial convolutions independently for each input channel.
2. **Pointwise convolution** – Merges the outputs of the depthwise convolutions.

A decoder module then refines feature representation by enhancing edge segmentation accuracy, linking low-level and high-level extracted features.

➢ **Loss Function for Model Optimization**

The overall loss function $l^S_{overall}$ for the final model is given in Equation (6.9):

$$l^S_{overall} = l^S_{(j_i,k_i)} + l_R \tag{6.9}$$

where,

— l_R is the regularization loss function.

— $l^S_{(j_i,k_i)}$ is the Softmax cross-entropy loss function.

The Softmax cross-entropy loss function is computed using Equation (6.10):

$$l^{S}_{(j_i,k_i)} = -\sum_i^M j(x) log k(x) \quad (6.10)$$

where,

— *M*: Number of images in the training dataset.

— *j(x):* Labeled pixel visual function.

— *k(x):* Expected pixel visual function.

The regularization loss function is computed using Equation (6.11).

$$l_R = \mu \sum_i^M (x_i)^2 \quad (6.11)$$

— l_R : Regularization loss function.

— μ : Regularization criterion.

— x_i : Input image pixel value at the *i*-th training image.

Regularization loss helps prevent overfitting, ensuring the model learns generalized features rather than memorizing the training set.

- **Spatial Pyramid Pooling Network (SPP)**

To accommodate deep networks processing images of arbitrary dimensions, the last pooling layer was substituted with a SPP layer. SPP aggregates responses from all feature map channels within multiple spatial bins, generating a fixed-length *kM*-dimensional vector, where:

— *k* is the count of feature map channels (or filters) in the last convolutional layer before SPP.

— *M* is the total number of spatial bins across all pyramid levels in the SPP layer.

— *kM* represents the final dimensionality of the output feature vector from SPP.

Traditional deep learning models often require fixed-dimensional inputs, limiting their ability to handle variable-sized images. SPP overcomes this constraint by allowing input images of arbitrary dimensions while maintaining consistent feature extraction. This flexibility ensures effective handling of different aspect ratios and scales without the need for cropping distortion.

SPP further enhances feature extraction by dividing the feature map into multiple spatial bins at different levels. Instead of relying on a single fixed-size feature map, it adaptively pools feature into a fixed- length representation, ensuring compatibility with downstream layers such as fully connected layers or LSTMs.

To improve anomaly detection, global and local modular features were integrated. The final pooled feature maps from the SPP block were merged with high-resolution features extracted from deeper network layers. The combined features were processed through a *1* x *1* convolutional layer, producing a structured output suitable for detecting anomalies across different scales and resolutions.

6.4 HIERARCHICAL M-CNN WITH LSTM MODEL

The challenges of VAD can be tacked by integrating advanced preprocessing in the proposed model. A Bilateral-Wave Denoising techniques is used for reducing noise and image enhancing, a Hierarchical M-CNN for spatial feature recognition and an LSTM for modelling temporal patterns in video sequence. The hybridization of the model helps to enhance the accuracy and generalization of VAD making it appropriate for surveillance systems. The proposed Hierarchical M-CNN with LSTM is illustrated in Figure 6.2.

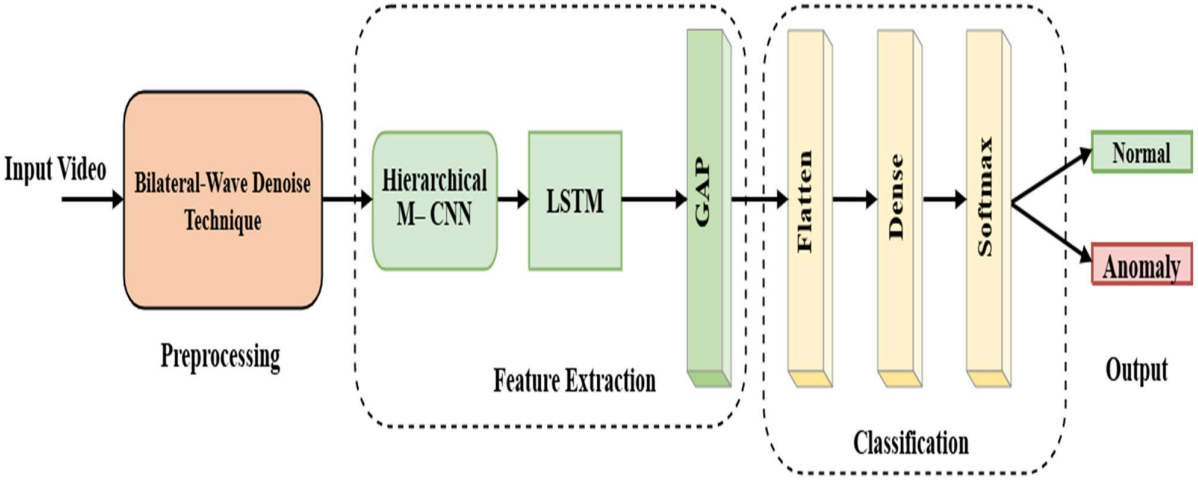

Figure 6.2 Architecture of Hierarchical M-CNN with LSTM

- **Preprocessing**

During preprocessing stage, a Bilateral-Wave Denoise technique is used to boost the image quality of video frames. This preprocessing technique is a hybridized model for

enhancing the low-resolution images by combining the Bilateral Filter and Wavelet Transformation.

The Bilateral filter can smoothen the images while preserving critical edge details. Simultaneously the images are decomposed into frequency components using Wavelet transform that enables the noise removal in specific bands. This dual approach helps to reduce noise effectively meanwhile preserves important image details, enabling efficient and accurate feature extraction of data.

- **Feature Extraction**

The integration of Hierarchical Multiscale -CNN(M-CNN) along with Long Short-Term Memory (LSTM) and Global Average Pooling (GAP) helps to identify the spatial and temporal patterns during feature extraction section for effective Anomaly Detection. The convolutional blocks in a M-CNN extracts multiscale spatial features and the adaptability of varying input dimensions are performed using Spatial Pyramid Pooling (SPP). The sequential dependencies of video are processed by identifying long-term patterns in the data using an LSTM network. The spatial dimensions in the input are reduced by averaging feature maps in the GAP layer thereby minimize overfitting and increasing generalization.

- **Hierarchical M-CNN**

The multiple convolutional blocks in a Hierarchical M-CNN extracts spatial features from the input images. These multiple blocks process images at different levels to capture the complex patterns in detail. For improving the computational efficiency and generalization ability the final pooling layer is substituted with the SPP layer. This layer helps to aggregate the features across multiple scales enabling the model to adapt the input images of varying dimensions. The atrous convolution is used in addition to expand the receptive field which empowers the capturing of large-scale features without increasing the computational complexity.

- **Long Short-Term Memory (LSTM)**

The identified spatial information is processed using LSTM model in order to identify the temporal patterns and the long-term dependencies. The LSTM layer consist of memory cells for the selection, retaining or forgetting the information over time making them effective for time series analysis and event prediction tasks. The input, forget and output gates are used to control the data flow and update the hidden state enabling the model to

concentrate of the processing of relevant temporal patterns at each step. The information shared across time steps and linear regression layer helps to forecast the outputs based on sequential data and minimize the prediction errors.

- **Global Average Pooling (GAP)**

The spatial dimensions of data are decreased by averaging all values with each feature map using a GAP layer. This step enables the simplification of data, avoid overfitting, reduce the number of arguments and improves the generalization capability of the model. The sequential data is analyzed using LSTM and patterns are classified over time by combining The LSTM with GAP. The GAP enables the feature maps to be compressed into a manageable format and thereby facilitates the model's capability to classify the input appropriately.

- **Classification**

The extracted features are transformed into meaningful predictions during the classification section using Flatten, Dense and Softmax Activation Function. The multidimensional feature maps are transformed into a one-dimensional array using Flatten layer enabling them compactible with fully connected layers.

The extracted features are aggregated using Dense layer, where each neuron represents an output class, enabling the final classification. The final Softmax Activation function converts raw output values into probability distributions, ensuring each input assigned to as class having the confidence score.

- **Flatten Layer**

For reducing the multidimension input into a one-dimensional array is performed in a Flatten layer. This layer is used for optimizing the process, making it suitable for fully connected layers in neural networks. The spatial information is preserved, enabling the data to be processed by Dense layer for classification or regression layers.

- **Dense Layer**

The Dense layer contains interconnected neurons, where each neuron is linked to all the other neurons in the preceding layer. The features learned by earlier layers are aggregated in this layer to generate the final output. In a classification task, the dense layer typically has as many neurons as there are output classes, with each neuron corresponding to a specific class.

- **Softmax Activation Function**

The Softmax activation function transforms raw output values into probabilities, with each probability indicating the chance that an input belongs to a given category. It simplifies probabilistic interpretation by guaranteeing that the total probability distribution across all classes sums to one. Algorithm 6.1 illustrates the Pseudocode for the Hierarchical M–CNN and LSTM for VAD.

Algorithm 6.1: Pseudocode for Hierarchical M–CNN with LSTM for VAD

Step 1: Load Video Frames

— Read and load the video frames from the input surveillance video. Process each frame sequentially.

Step 2: Apply the Bilateral-Wave Denoise Technique

— Apply bilateral-wave denoise to each frame to reduce noise while preserving edges. Decompose frames into frequency components using wavelet transform. Save the denoised frames for further processing.

— Decompose images into different frequency components.

— Save the denoised frames for further processing.

Step 3: Initialize Hierarchical M– CNN with SPP

Hierarchical M–CNN: Extracts multiscale spatial features.

— Replace the final pooling layer with an SPP layer to decrease the parameters count and increase feature extraction.

Step 4: Temporal Feature Extraction with LSTM

Extract Temporal Features

— Use LSTM layers to obtain temporal features from the SPP output. The LSTM captures sequential patterns and temporal dependencies in the video frames.

Step 5: Apply GAP

— Apply Global Average Pooling (GAP) to reduce dimensionality.

Step 6: Classification using Flatten, Dense and Softmax activation

— Flatten the pooled output and pass it through Dense layers with Softmax activation for classification into normal or anomalous categories.

Step 7: Model Integration

Combine All Components into a Unified Model

— Integrate the Multiscale CNN, Temporal LSTM and Anomaly Classifier into a single model.

6.5 RESULTS AND DISCUSSIONS

The outcomes of the Hierarchical M-**C**NN with LSTM for VAD are presented through performance evaluation and comparative analysis. Figure 6.3 illustrates the Original and pre-processed images after the application of filtering techniques.

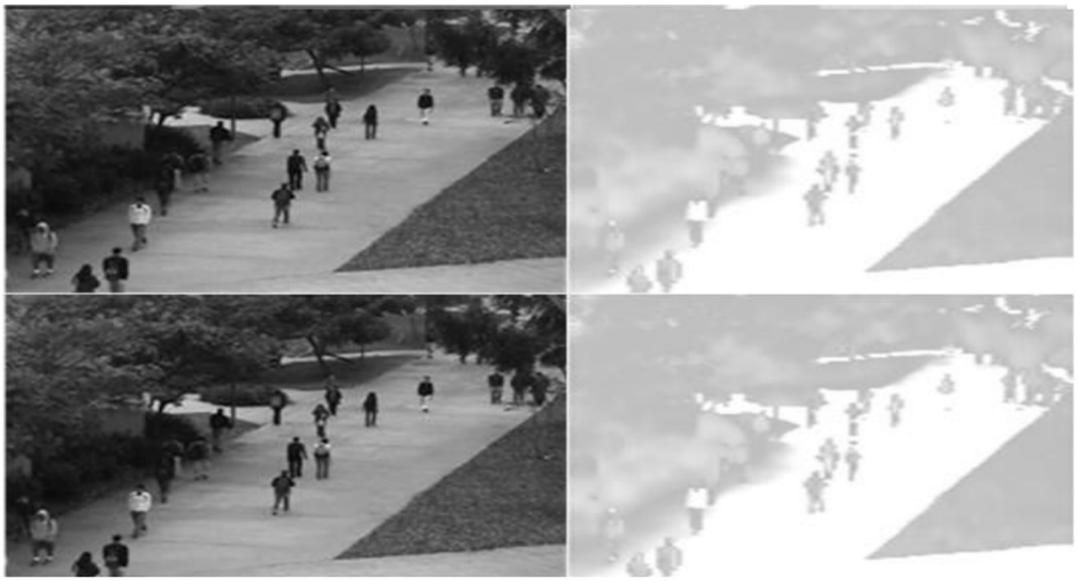

Figure 6.3 Original and pre-processed images

During the preprocessing stage, the bilateral filter technique is employed to enhance the quality of images extracted from video frames. This step ensures improved clarity and reduced noise, which are critical for effective anomaly detection. Figure 6.4 depicts the final classification results for VAD, showing the distinction among normal and anomalous images. The Figure 6.4 compares normal and anomalous events in a surveillance scene,

where anomalies are highlighted using green bounding boxes. These anomalies include unusual behaviors such as irregular movement patterns or individuals engaging in activities that deviate from normal pedestrian behavior.

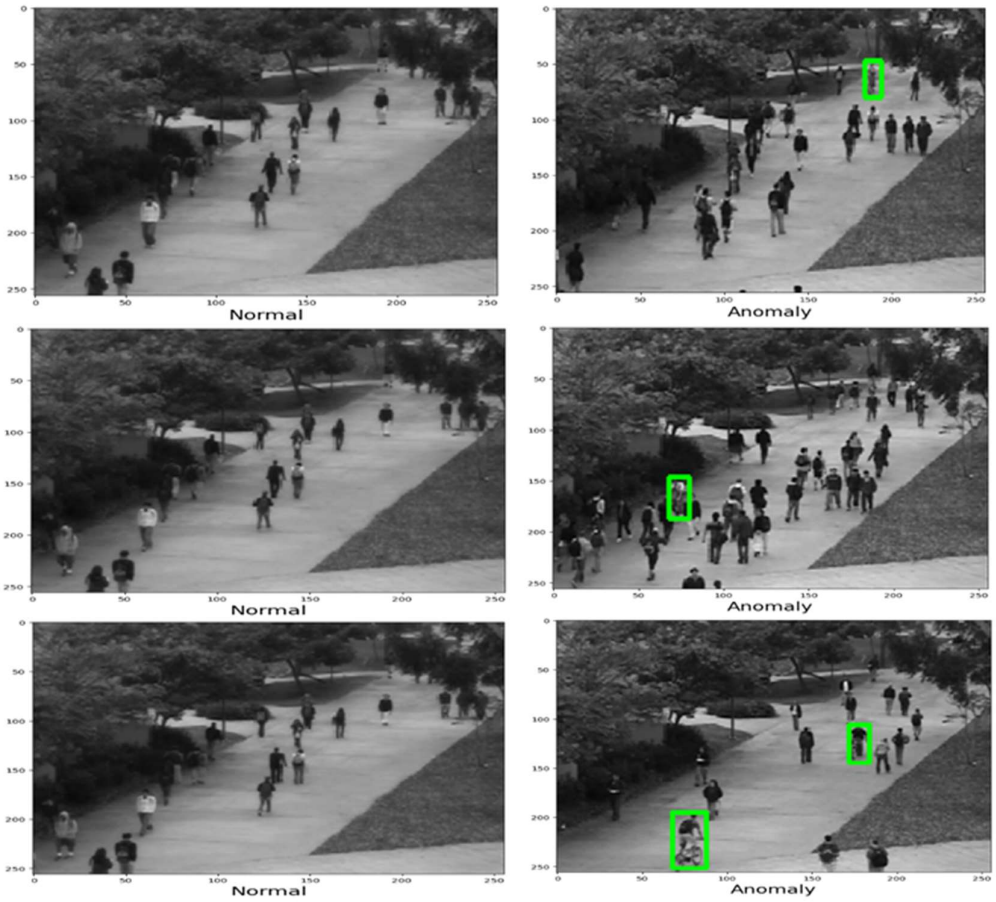

Figure 6.4 Results of Normal and Anomalous Image

The proposed method demonstrates high efficiency in predicting anomalies by accurately localizing them. The Hierarchical M-CNN enhances the anomaly detection capabilities in video frames by replacing the conventional pooling layer with a SPP layer that enables to diminish the number of parameters and increase the computational efficiency. This alteration enhances the feature extraction capabilities for effective detection of anomalies.

The presence of LSTM layer enables the model to capture temporal dependencies within video sequences for precise anomaly calculations. To process the extracted features and classification of the model the Flatten, Dense and Softmax Activation Function are

incorporated. The results given in Figure 6.4 emphasized the model's capacity to discriminate among normal and anomalous frames effectively.

6.5.1 Performance Evaluation

The performance of the Hierarchical M-CNN with LSTM model for multiscale AD is discussed in this section. The model's classification accuracy and loss across 25 training epochs and the model's convergence period is also evaluated. The Figure 6.5 depicts the training accuracy of the model.

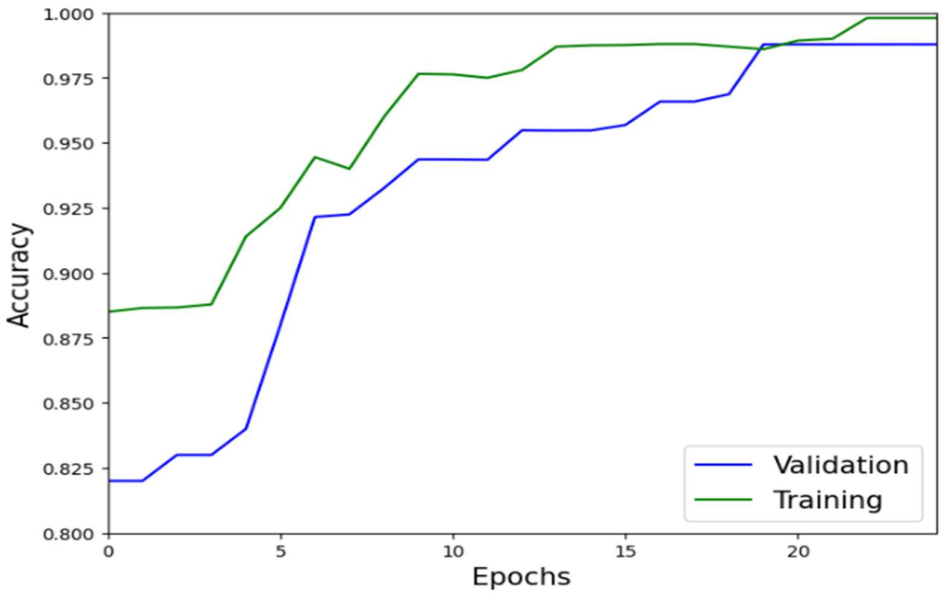

Figure 6.5 Training Accuracy of the Hierarchical M-CNN with LSTM

The Hierarchical M-CNN with LSTM model undergoes training and validation for multiscale AD over 25 epochs. The renowned Adam optimizer helps in adaptive learning rate an the frame work achieved an impressive accuracy of 99.35% demonstrating the exceptional capacity detect and analyze data patterns effectively. The model's reliability in consistently identifying normal and anomalous frames is highlighted by the Precision and make the model well-suited for accurate anomaly detection applications.

Figure 6.6 presents the training and validation loss of the proposed method. The model demonstrated a rapid reduction in loss metric with an average loss of 4.4% over the training period of 25 epochs. This efficient loss reduction indicates the optimizer's ability to adaptively update parameters, enabling the model to converge to the optimal solution while

avoiding overfitting by 1.01%. The model's increased accuracy and the loss indicates a balance of computational efficiency and generalizability.

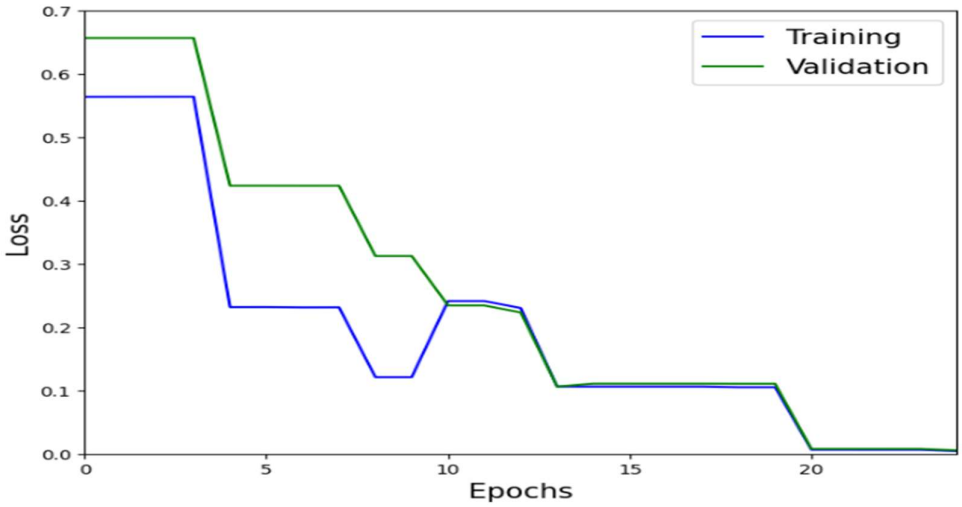

Figure 6.6 Training Loss of the Hierarchical M-CNN with LSTM

The results further confirm the efficiency and reliability of the framework for practical anomaly detection tasks. Table 6.1 provides the performance metrics of the Hierarchical M – CNN Fusion with LSTM, reinforcing its capability to maintain high accuracy and robust performance throughout the training process.

Table 6.1 Performance of Hierarchical M–CNN with LSTM Model

Performance Metrics	Hierarchical M–CNN with LSTM
Accuracy	99.35 (%)
Precision	99.35 (%)
Recall	99.35 (%)
F1 Score	99.35 (%)
AUC	0.9985
PSNR	42.18 (dB)
EER	5.4 (%)

Figure 6.7 to Figure 6.13 illustrate the performance comparison of the Hierarchical M–CNN with LSTM model against IUNet-CSWT and ResNet-LSTM. Figure 6.7 compares Accuracy, demonstrating that the Hierarchical M–CNN with LSTM model achieves 99.35%,

outperforming IUNet-CSWT at 99% and ResNet-LSTM at 96.5%. The 0.35% increase over IUNet-CSWT and 2.85% improvement over ResNet-LSTM highlight its superior classification capability in anomaly detection.

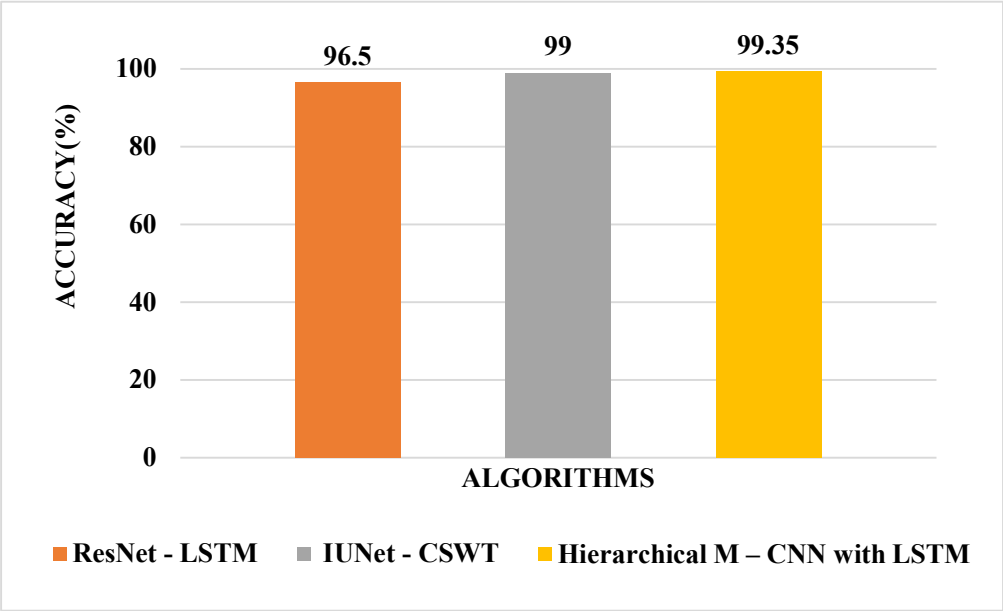

Figure 6.7 Performance Comparison of Accuracy

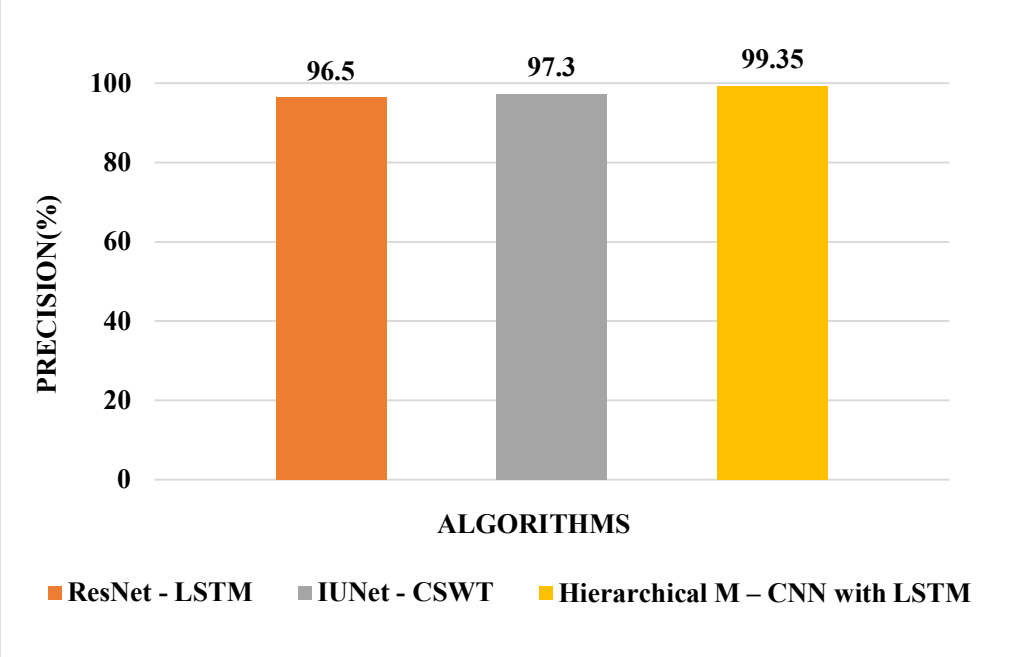

Figure 6.8 Performance Comparison of Precision

Figure 6.8 illustrates Precision, where the Hierarchical M–CNN with LSTM model attains 99.35%, exceeding IUNet-CSWT at 97.3% and ResNet-LSTM at 96.5%. The 2.05% increase over IUNet-CSWT demonstrates improved accuracy in correctly identifying anomalies.

Figure 6.9 presents Recall, with the Hierarchical M–CNN with LSTM model reaching 99.35%, compared to 97.5% for IUNet-CSWT and 96.5% for ResNet-LSTM. The 1.85% improvement over IUNet-CSWT highlights enhanced sensitivity in detecting true anomalies.

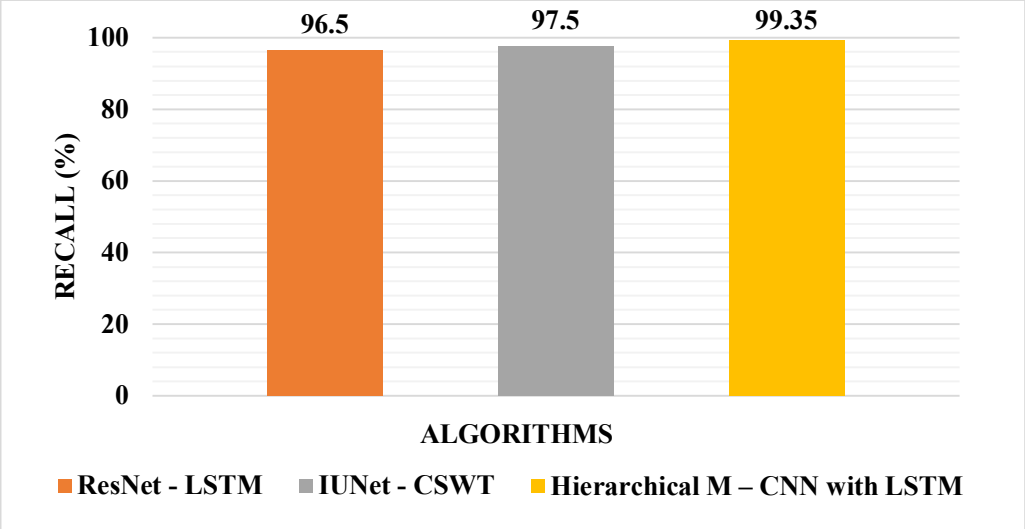

Figure 6.9 Performance Comparison of Recall

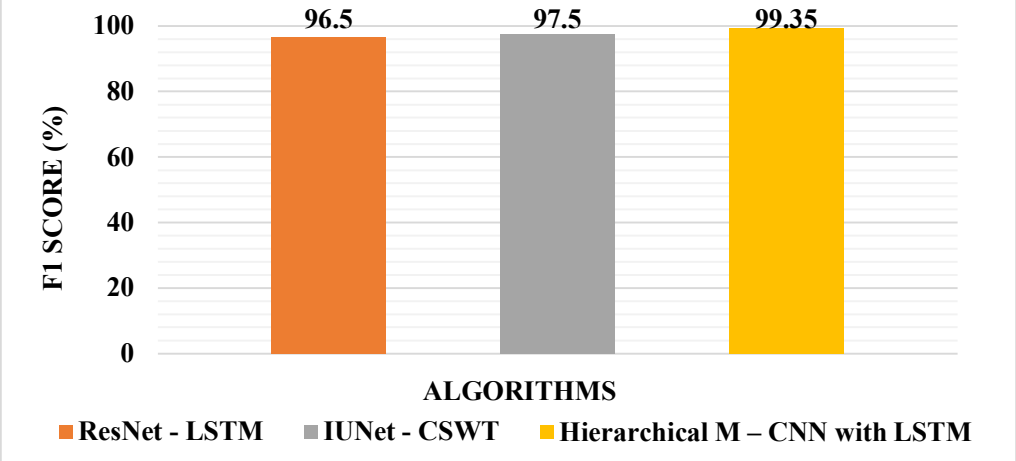

Figure 6.10 Performance Comparison of F1 Score

Figure 6.10 displays the F1 Score, where the Hierarchical M–CNN with LSTM model attains 99.35%, surpassing IUNet-CSWT at 97.5% and ResNet-LSTM at 96.5%. This 1.85% increase over IUNet-CSWT reflects a improved balance between Precision and Recall, ensuring more reliable anomaly classification.

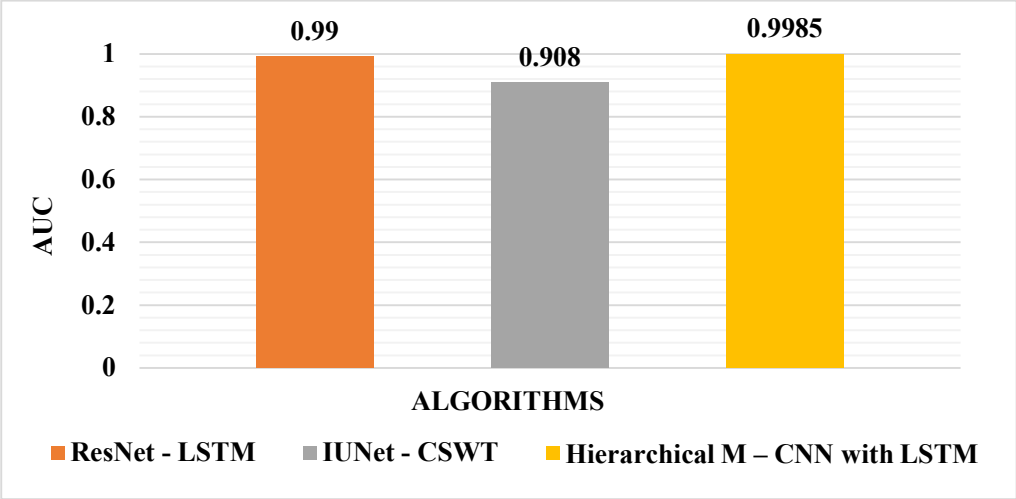

Figure 6.11 Performance Comparison of AUC

Figure 6.11 compares the Area Under the Curve (AUC), showing that the Hierarchical M–CNN with LSTM model records 99.85%, outperforming ResNet-LSTM at 99% and IUNet-CSWT at 90.8%. The 9.05% higher value compared to IUNet-CSWT suggests a stronger capability to differentiate normal and anomalous events.

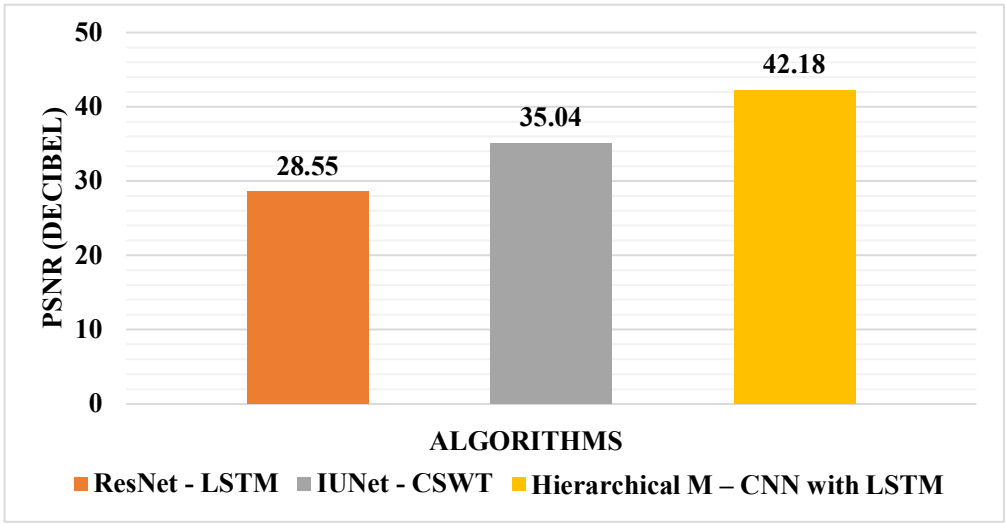

Figure 6.12 Performance Comparison of PSNR

Figure 6.12 evaluates the Peak Signal-to-Noise Ratio, with the Hierarchical M–CNN with LSTM model achieving 42.18dB, compared to IUNet-CSWT at 35.04dB and ResNet-LSTM at 28.55dB. The 7.14dB improvement over IUNet-CSWT and 13.63dB increase over ResNet-LSTM signifies superior video reconstruction quality, which is essential for accurately detecting anomalies.

Figure 6.13 highlights the Equal Error Rate (EER), where the Hierarchical M–CNN with LSTM model achieves 5.4%, outperforming IUNet-CSWT at 10.9% and ResNet-LSTM at 14.5%. The 5.5% reduction in error rate from IUNet-CSWT and 9.1% decrease from ResNet-LSTM indicate significantly fewer misclassifications, improving overall model reliability.

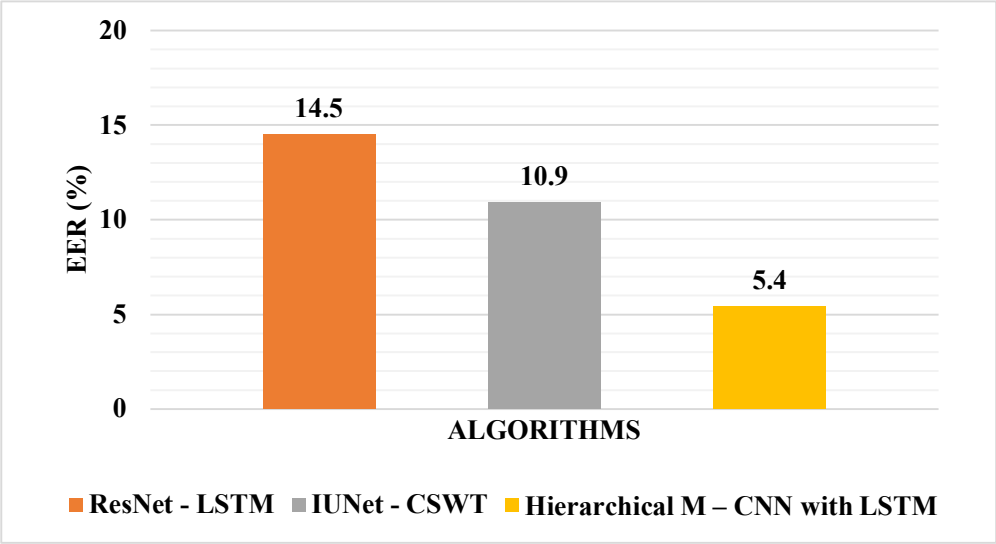

Figure 6.13 Performance Comparison of EER

6.6 SUMMARY

This model introduces an innovative approach for automated AD in surveillance videos, enhancing public security by reducing false positives which are common in traditional methods. Bilateral-Wave Denoise for noise reduction and Hierarchical Fusion of Multiscale CNN with LSTM mitigated overfitting and captured temporal features for precise classification. This hybrid model achieved exceptional Accuracy, Precision, Recall and F1 Score amounting to 99.35%, an AUC of 0.9985 and a low EER of 4.4%. These results authenticate its effectiveness and potential for advancing public security systems.