IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 August 2023, accepted 25 September 2023, date of publication 4 October 2023, date of current version 13 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3321801

RESEARCH ARTICLE

SwinAnomaly: Real-Time Video Anomaly Detection Using Video Swin Transformer and SORT

ARPIT BAJGOTI[1], RISHIK GUPTA[1], PRASANALAKSHMI BALAJI[2], (Member, IEEE), RINKY DWIVEDI[1], MEENA SIWACH[3], AND DEEPAK GUPTA[4]

[1]Department of Computer Science and Engineering, Maharaja Surajmal Institute of Technology, New Delhi 110058, India
[2]Department of Computer Science, College of Computer Science, King Khalid University, Abha 61421, Saudi Arabia
[3]Department of Information Technology, Maharaja Surajmal Institute of Technology, New Delhi 110058, India
[4]Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology, Delhi 110086, India

Corresponding author: Prasanalakshmi Balaji (drsana@ieee.org)

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Small group Research Project under grant number RGP1/325/44.

ABSTRACT Detecting anomalous events in videos is a challenging task due to their infrequent and unpredictable nature in real-world scenarios. In this paper, we propose SwinAnomaly, a video anomaly detection approach based on a conditional GAN-based autoencoder with feature extractors based on Swin Transformers. Our approach encodes spatiotemporal features from a sequence of video frames using a 3D encoder and upsamples them to predict a future frame using a 2D decoder. We utilize patch-wise mean squared error and Simple Online and Real-time Tracking (SORT) for real-time anomaly detection and tracking. Our approach outperforms existing prediction-based video anomaly detection methods and offers flexibility in localizing anomalies through several parameters. Extensive testing shows that SwinAnomaly achieves state-of-the-art performance on public benchmarks, demonstrating the effectiveness of our approach for real-world video anomaly detection. Furthermore, our proposed approach has the potential to enhance public safety and security in various applications, including crowd surveillance, traffic monitoring, and industrial safety.

INDEX TERMS Video anomaly detection, SORT, GAN-based autoencoder, real-time tracking.

I. INTRODUCTION

Video anomaly detection is a technique used in computer vision to automatically detect unusual or anomalous events in video data. This can be useful for a wide range of applications, including surveillance, security, and traffic monitoring. However, this task has become increasingly challenging in recent times due to diversely distributed data. One method that has been employed to tackle this problem is one-class classification, where only normal videos are shown without any anomaly, and at testing the pre-trained model is required to distinguish between normal and anomalous events in complex video data.

Over the years, several methods such as reconstruction-based methods and prediction-based methods have achieved significant progress in anomaly detection. Reconstruction methods work on the principle that normal events can be reconstructed accurately whereas abnormal events will have a higher reconstruction error and hence can be detected as an anomaly. The prediction-based method on the other hand works on the principle that normal events can be accurately predicted, while abnormal events would not be correctly predicted.

Convolutional Neural Networks (CNNs) are necessary for prediction and reconstruction-based methods for their pattern recognition capabilities. However, they suffer from challenges such as the inability to find unexpected anomalies that deviate significantly from the training data since CNNs are designed to identify patterns in data. Another limitation of

The associate editor coordinating the review of this manuscript and approving it for publication was Laxmisha Rai.

CNNs is the extensive computational power that they require which can be time-consuming for larger datasets such as video data.

Transformers in recent times have shown exceptional capabilities in Natural Language Processing (NLP) tasks such as language translation, text summarization, and sentiment analysis. Unlike CNNs which process input sequences sequentially, transformers process the entire sequence of input data in go. To achieve this, a mechanism called self-attention is used. Following the success of transformers in NLP, several advancements have been made in computer vision employing the use of transformers. Firstly, being introduced in Vision Transformer (ViT) [1] which takes 16 × 16 image patches for image classification. It was later adapted into Video Vision Transformer (ViViT) [2], which explored the application of ViT in video classification. The main idea behind using transformers in computer vision is to apply the self-attention mechanism to the spatial dimensions of the input image, which is achieved using a technique called spatial self-attention.

A. SWIN TRANSFORMER

Swin Transformer is a type of transformer architecture initially used for image classification tasks, first introduced by Microsoft Research Asia [3] in 2021. In the Swin Transformer, the input image is first divided into small patches, and then these patches are further divided into feature maps. This method allows the Swin Transformer to capture both global and local features of the image. The use of shifted windows enables the transformer to capture more fine-grained details and better handle small objects in the image. In the context of autoencoder-based methods for video anomaly detection, Swin Transformers offer several advantages such as better feature extraction as it uses self-attention to capture long-range dependencies in the input sequence, reduced computational cost, and improved reconstruction quality by reducing distortion in the reconstructed frames. The Swin Transformer supports upsampling and downsampling, unlike ViViT which makes it the optimal choice to be used as an encoder and decoder in a Generative Adversarial Network (GAN) for image generation tasks.

B. GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Networks (GANs) have been successful in generating realistic images [4]. Recently, there has been a growing interest in combining parallel processing of transformers and realistic image generation of GANs for image segmentation tasks. In this approach, a generator network, which is a transformer-based architecture, takes an input image and produces a segmentation map. The discriminator network, which is typically a CNN-based architecture, takes both the input image and the generated image and tries to distinguish between real and fake images. Vision Transformer GAN (ViTGAN) is one such model that was proposed by Lee et al. [5] which uses a transformer-based generator and a CNN-based discriminator and has achieved state-of-the-art results. Similar to VitGAN, our model SwinAnomaly uses a swin transformer-based generator and discriminator which utilizes the benefits of using a swin transformer.

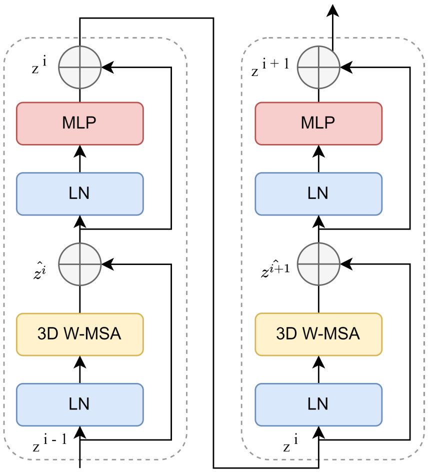

FIGURE 1. Two successive video swin transformer blocks.

In recent years, several different architectures of GANs such as Deep Convolutional GAN (DCGANs) [6], [7] and PatchGAN [8] were introduced for image synthesis tasks. The CNN-based discriminator network type DCGAN processes the input image using many layers of convolutional and pooling procedures to produce a scalar probability score representing the realism of the image. DCGANs can have difficulty producing sharp edges and fine-grained features, but they are good at producing high-quality images with realistic textures and details. PatchGAN on the other hand concentrates on assessing the realism of local image patches. The input image is divided into several, non-overlapping patches using the PatchGAN, and each patch's level of realism is assessed independently. The discriminator can capture fine-grained information in the image and produce higher-quality photos with more realistic textures by employing this method. While PatchGANs can provide images that are clearer and more detailed than DCGANs, they may also take longer to train. The results of using both PatchGAN and DCGAN in the proposed approach have been discussed in the paper.

Although many machine learning-based anomaly detection methods have been introduced in recent years, they still suffer from some limitations. The current video anomaly detection methods struggle to generalize well to unseen or novel scenarios, performing well on the datasets they were trained on but fail to detect anomalies in new and diverse environments. Some methods, particularly those based on neural networks can be computationally intensive and require significant computational resources, making real-time deployment challenging.

In this paper, we propose SwinAnomaly, a novel architecture that aims to enhance anomaly detection in video datasets, also allowing the model to be used in real-life scenarios. The encoder of the generator in the PatchGAN adopts the Video Swin Transformer architecture, while the decoder utilizes 2D Swin-transformer blocks to predict future frames from a sequence of video frames. Real-time anomaly detection is achieved by integrating an object detection model and the SORT algorithm, enabling the detection of anomalies using reconstruction error patches. Experimental results validate the superiority of SwinAnomaly over other prediction-based methods, highlighting its potential to enhance anomaly detection systems for more accurate and efficient real-world applications. The main contributions of the paper are as follows:

- We propose a model that combines Swin-transformers' patchwise feature extraction with the reconstruction capabilities of convolution-based autoencoders.
- SwinAnomaly operates on the current set of frames, without requiring a specific frame accumulation, allowing for the definition and continuous tracking of anomalies.
- We develop a technique that uses an object detection model such as YOLOv7 along with the SORT algorithm to identify and track anomalies in real-time.

II. RELATED WORKS

Several works have been done in the field of anomaly detection in the past decade. Generally, unsupervised video anomaly detection methods can be categorized into reconstruction-based methods and prediction-based methods.

A. RECONSTRUCTION-BASED METHODS

Reconstruction-based methods are a popular approach for video anomaly detection. The idea behind this method is to learn a model of normal behavior from a training set of videos, and then use this model to identify anomalies in new videos. It assumes that frames having anomalies will have higher reconstruction errors than frames without any anomalies. The most popular model used is the autoencoder, consisting of a convolutional encoder [11]. Other approaches by Fan et al. [12] and Li and Chang [13] used Variational Auto-Encoders (VAEs) to reconstruct the input frames, and the distribution difference of the latent representations was utilized to calculate the regularity scores. However, training VAEs was found to be difficult, particularly if the dataset was uneven or limited. Furthermore, the inability of VAEs to explicitly express temporal relationships may limit their ability to accurately capture the dynamics of a video sequence. To overcome this, Chong et al. [14] and Luo et al. [15] combined Convolutional Long Short-Term Memory (ConvLSTM) with Convolutional Auto-Encoder to extract more temporal information from input sequences. Incorporating the temporal information into the model helped improve its accuracy by allowing it to learn the dynamics of normal behavior over time. To find anomalies, certain reconstruction-based techniques also take advantage of the latent representational differences between normal and abnormal samples. However, using ConvLSTM had several challenges such as high computational cost, and overfitting, and it could not handle complex scenes with multiple objects and occlusions. In recent years, optical flow-based motion constraints have also been used in combination with Convolutional Auto-Encoder [16]. However using optical flow comes with its limitations such as high computational cost, sensitivity to noise, and lack of semantic understanding. Chang et al. [17] employed two autoencoders to individually leverage the spatial and temporal information of videos to get around the high computational cost of optical flow. While the temporal autoencoder recorded the movement data of the objects, the spatial autoencoder encoded the scenes and objects.

B. PREDICTION-BASED METHODS

Prediction-based methods for video anomaly detection involve training a model to predict future frames or events in a video sequence and detecting anomalies as deviations from this prediction. The basic assumption is that the normal event is predictable whereas the abnormal one is unpredictable [18]. These methods typically use Recurrent Neural Networks (RNNs) or Convolutional Neural Networks (CNNs) to learn the temporal dynamics of the video sequence and predict future frames or events. One common method is to predict upcoming frames in the video sequence based on previous frames using a sequence-to-sequence model, such as an encoder-decoder architecture with a Conv-LSTM [19] or Gated Recurrent Unit (GRU) [20] based RNN. The model can produce precise frame predictions under typical circumstances after being trained on a collection of typical video sequences. When there are departures from this prediction, such as abrupt changes in the projected frame or mistakes in the predicted motion, anomalies are recognized. In recent years, U-Net has been used as a generator for predicting the next frame, and a patch discriminator is used to distinguish the generated frames. Liu et al. [18] proposed a similar network architecture where U-net was used as a generator, and the model computed a regularity score with only the prediction error without reconstruction. However, using U-Net can be computationally expensive, they are not well suited for detecting anomalies that occur over longer time scales. In certain instances, reconstruction is combined with prediction models. For instance, Ye et al. [21] presented a Predictive Coding Network that, initially, predicts future frames using a ConvLSTM with predictive coding. Then, reconstruction techniques are used to reduce prediction errors. Lastly, to improve prediction performance, the predicted frames are updated with refined errors. This method still uses prediction error to get the regularity score while additionally considering reconstruction difference. In addition, Li et al. [22] replaced the original U-Net generator with a spatial-temporal U-Net that added three ConvLSTM layers between the U-Net layers to extract more temporal information. Tang et al. [23] on the other hand followed a hybrid approach by combining both prediction and reconstruction-based methods by connecting two U-Net blocks, the first one for predicting the future frame and the second for reconstructing the future frame.

Although Convolutional-based methods work well, they require high computational power to generate reconstructed images. In our proposed approach, we use transformers to significantly reduce the processing cost of the individual frames which is achieved because of the self-attention mechanism and parallel processing capabilities of transformers.

C. OBJECT LEVEL DETECTION

The methods discussed above operate on entire video frames. This might prove to be difficult for anomaly detection as the video frames are complex, contain variations, and contain a large number of objects. More recent methods such as Object-level video anomaly detection aim to identify anomalies at the level of individual objects or regions within a video sequence [24], [25], [26]. They first extract object boundaries using object detection algorithms. Then the tracked objects are checked to be anomalous. This is an easier task since individual objects contain much less variation than the whole frame. The need for precise and effective object detection and tracking algorithms is one difficulty in object-level video anomaly detection. Occlusion, motion blur, and changes in lighting or background can all affect how accurate these algorithms are. In addition, these algorithms may be difficult to implement in real-time applications due to their computational cost.

D. TRANSFORMER BASED METHODS

Following the success of transformers in the field of NLP [27], many attempts have been made to replicate this success for computer vision tasks. Due to their capacity to represent long-term dependencies and capture intricate spatiotemporal correlations in video sequences, transformers have demonstrated promise in the field of video anomaly identification. Transformers are more effective at simulating temporal dynamics than more conventional approaches like ConvLSTM and can minimize processing time. Deshpande et al. [28] used a three-stage model consisting of a pre-trained videoswin model, an attention layer, and lastly an RFTM model for anomaly detection. TransUNet is a hybrid of U-Net with a transformer encoder, as suggested by Chen et al. [29]. TransUNet, a hybrid CNN-Transformer architecture, makes use of both the global context stored by the transformer encoder and the finely detailed, high-resolution spatial information from CNN features. Yuan et al. [30] proposed a similar architecture that combined U-Net and ViViT to capture richer temporal information and more global contexts. The ViViT was modified to make it capable of video prediction. Another strategy is to combine the usage of transformers with other machine learning methods, like object detection and tracking algorithms. This can entail projecting the future trajectory of certain objects inside a video sequence using a transformer-based model and then analyzing the behavior and motion of the items to spot anomalies.

While these methods have demonstrated impressive performance and yielded favorable outcomes across various datasets, their reliance on a substantial number of frames during inference poses certain limitations. Moreover, these approaches are primarily designed to detect local anomalies, meaning their ability to identify global anomalies is restricted. In cases where the chosen set of frames contains a considerable number of anomalies, the distinction between the minimum and maximum Peak signal-to-noise ratio (PSNR) becomes indiscernible, leading to a smooth anomaly curve. As a consequence, the effectiveness of these methods in accurately capturing and differentiating anomalies is diminished.

III. PROPOSED WORK

The overall architecture of the SwinAnomaly model is based on a Generative Adversarial Network where the generator is a Swin-UNET [31] based Autoencoder containing a 3D Encoder with Video Swin transformer layers and a decoder based on Swin-UNet decoder layers, The generator takes 3D input in the shape of (c, t, w, h) as shown in Fig. 4(a) and outputs a 2D predicted frame with dimensions (c, w, h), the generator tries to predict new frame $\hat{f}_{t+1}$ and the discriminator tries to differentiate between actual frames in the frame sequence and the predicted frame from the generator in an adversarial way, as shown in Fig. 2. The detailed overview of the proposed GAN constituents is shown in Fig. 3 and is discussed as follows:

A. FRAME ENCODER

The Encoder takes the input in the form $f_1, f_2, \ldots, f_T$ where each matrix f is a video frame of size $W \times H \times C$ and T is the number of frames taken together and concatenated to give a size of $T \times W \times H \times C$ to the input. The input is passed to a 3D patch partition layer which divides the input into $\frac{T}{t} \times \frac{W}{4} \times \frac{H}{4}$ patches where t is the number of frame partitions in the input data as shown in Fig. 4(b). The value of t is a small integer value where $kt = T$ as the frame size must be justified for input frames to be converted from a 3D input to a 2D patch where each patch contains the information content of the corresponding pixels of the frames in sequential order as shown in Fig.4(a). After the patch partition, a linear embedding layer is applied to project the patches into E-dimensional feature vectors, in the Swin-T-based architecture the embedding dimensions are taken as E = 96. The multi-head attention layer (MSA) is replaced with 3D windowed MSA and the consecutive block with cyclic shifted windowed MSA with shift space of (2, 2, 2) tokens as shown in Fig. 4(c), each encoder block's output is resized and propagated through an MLP layer to downsample the 3D input to 2D embedding which is passed to the adjacent decoder block as a skip connection to the decoder. Each encoder block is also down-sampled using a patch-merging

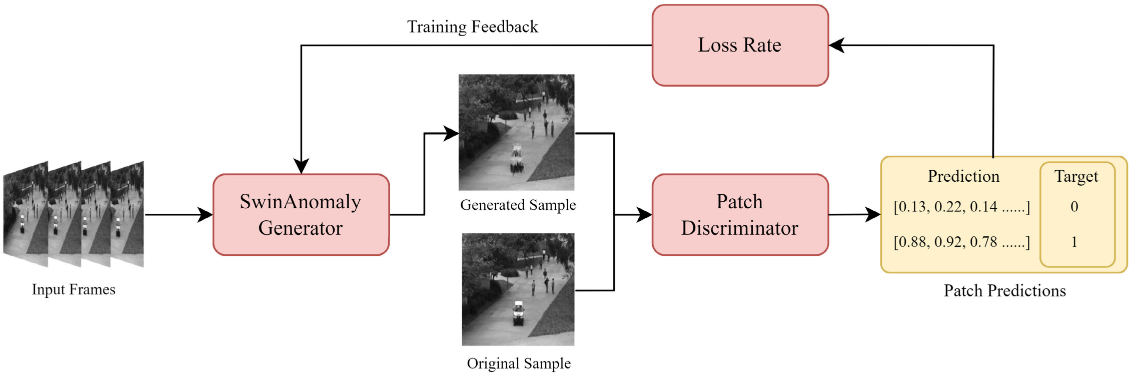

FIGURE 2. **Conditional PatchGAN-based workflow of proposed approach.**

layer, where the patches are reduced by two times i.e. ($T \times \frac{W}{2} \times \frac{H}{2} \times 2C$), the consecutive blocks in each video swin transformer is shown in Fig. 1. It can be noticed that the channels and the image dimensions are altered to extract the features but the temporal frame patch size t is not reduced to maintain the integrity of our model during skip connections and upsampling [32].

1) 3D SHIFTED WINDOW BASED MSA

As the multi-head self-attention mechanism is used only within each non-overlapping 3D window, there are no connections between windows, which may reduce the architecture's capability for representation. As a result, the shifted 2D window mechanism of the Swin Transformer is converted to 3D windows to introduce cross-window connections, which is achieved by the circular shift of the patched frames as proposed by Liu et al. [32] while preserving the effective calculation of non-overlapping window-based self-attention.

$$\begin{aligned}
\hat{z}^{l} &= 3DW - MSA(LN(z^{l-1})) + z^{l-1}, \\
z^{l} &= MLP(LN(\hat{z}^{l})) + \hat{z}^{l}, \\
\hat{z}^{l+1} &= 3DSW - MSA(LN(z^{l})) + z^{l}, \\
z^{l+1} &= MLP(LN(\hat{z}^{l+1})) + \hat{z}^{l+1} \qquad (1)
\end{aligned}$$

where $\hat{z}^{l}$ and z^{l} represent the outputs of the $(S)W - MSA$ module and the MLP module of the l_{th} block, respectively.

2) 3D RELATIVE POSITION BIAS

The addition of a relative position bias B significantly improves the performance of swin-transformers in feature extraction. For incorporating 3D patches, a new dimension P is added which represents patches in the 3rd dimension. The Attention mechanism of the attention layer in Swin block 3D is governed by the following equation:

$$Attention(Q, K, V) = SoftMax\left(\frac{QK^{T}}{\sqrt{d}} + B\right)V \qquad (2)$$

where the query, key, and value matrices are $Q, K, V \in \mathbb{R}^{PM^{2} \times d}$, d is the dimension of the query and key features, and PM^{2} is the total number of tokens in a 3D window. A smaller-sized bias matrix, $\hat{B} \in \mathbb{R}^{(2P-1)\times(2M-1)\times(2M-1)}$ is parameterized, and values in B are picked from $\hat{B}$ because the relative position along each axis falls within the range of $[-P+1, P-1]$ (temporal) or $[-M+1, M-1]$ (height or width).

3) PATCH MERGING LAYER

The patch merging layer separates the input patches into 4 parts, which are then concatenated. The feature resolution will be two times down-sampled with this processing. Additionally, because the concatenate operation causes the feature dimension to increase by 4×, to reduce the complexity of the model, a linear layer is applied to the concatenated features to bring the feature dimension back to the original 2× dimension.

B. FRAME DECODER

The symmetric decoder in Swin Transformer is constructed based on Swin Transformer blocks, similar to the encoder. However, the decoder applies attention to 2D patches instead of 3D. Unlike the patch merging layer used in the encoder, the decoder utilizes a patch-expanding layer to up-sample the extracted deep features. This layer reshapes the adjacent dimension feature maps into a higher resolution feature map with 2× up-sampling while reducing the feature dimension to half of the original dimension.

1) PATCH EXPANDING LAYER

Let's take the first patch expanding layer as an example. Before up-sampling, a linear layer is applied to the input features ($\frac{W}{32} \times \frac{H}{32} \times 8C$) to increase the feature dimension to 2x the original dimension ($\frac{W}{32} \times \frac{H}{32} \times 16C$). Then, a rearrange operation expands the input feature resolution to 2x the original resolution and reduces the feature dimension to a quarter of the input dimension ($\frac{W}{32} \times \frac{H}{32} \times 16C \rightarrow \frac{W}{16} \times \frac{H}{16} \times 4C$).

2) 2D SHIFTED WINDOW MSA AND POSITIONAL BIAS

The decoder uses 2D patches unlike encoders hence the equations 1 and 2 can be easily modified to sample

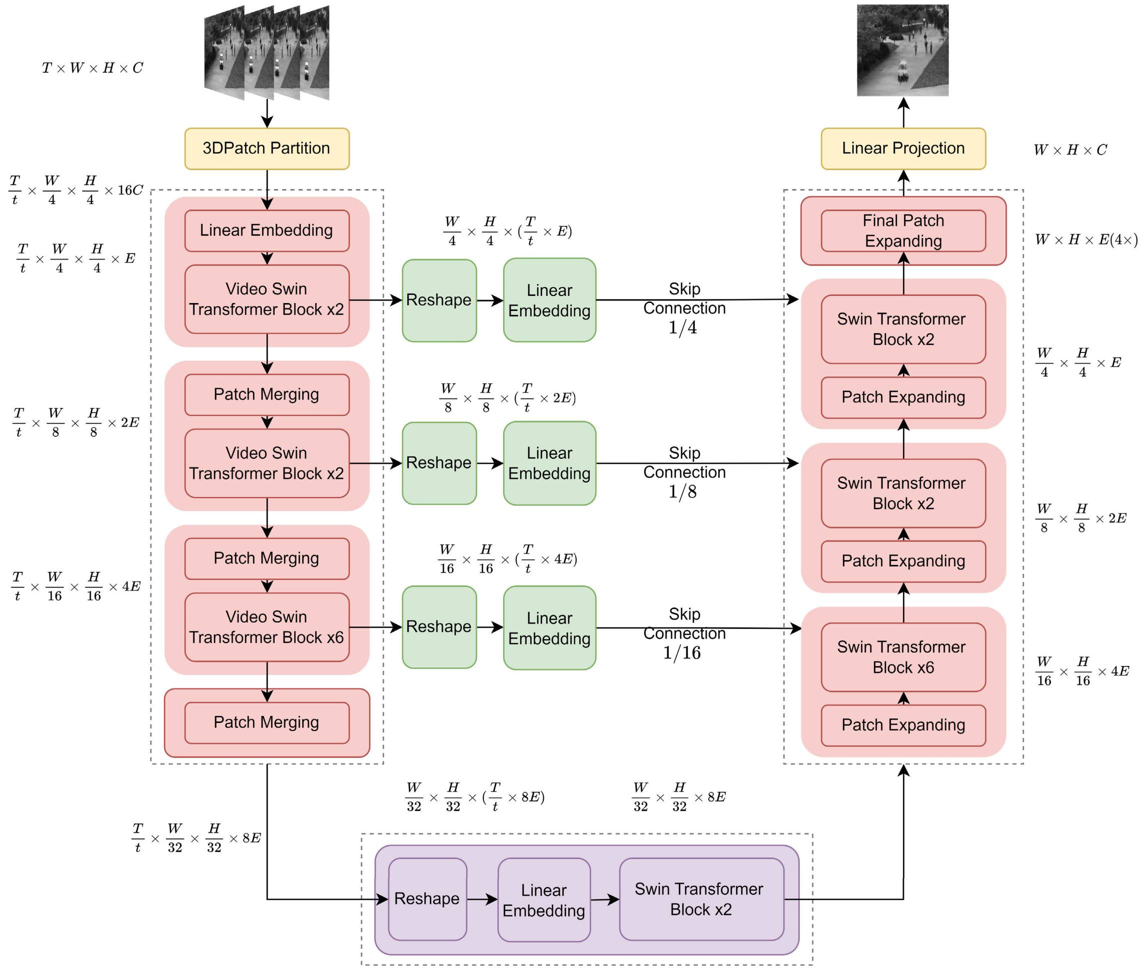
$T \times W \times H \times C$
3DPatch Partition
$\frac{T}{t} \times \frac{W}{4} \times \frac{H}{4} \times 16C$
Linear Embedding
$\frac{T}{t} \times \frac{W}{4} \times \frac{H}{4} \times E$
Video Swin Transformer Block x2
Patch Merging
$\frac{T}{t} \times \frac{W}{8} \times \frac{H}{8} \times 2E$
Video Swin Transformer Block x2
Patch Merging
$\frac{T}{t} \times \frac{W}{16} \times \frac{H}{16} \times 4E$
Video Swin Transformer Block x6
Patch Merging
$\frac{W}{4} \times \frac{H}{4} \times (\frac{T}{t} \times E)$
Reshape
Linear Embedding
Skip Connection 1/4
$\frac{W}{8} \times \frac{H}{8} \times (\frac{T}{t} \times 2E)$
Reshape
Linear Embedding
Skip Connection 1/8
$\frac{W}{16} \times \frac{H}{16} \times (\frac{T}{t} \times 4E)$
Reshape
Linear Embedding
Skip Connection 1/16
Linear Projection
$W \times H \times C$
Final Patch Expanding
$W \times H \times E(4\times)$
Swin Transformer Block x2
Patch Expanding
$\frac{W}{4} \times \frac{H}{4} \times E$
Swin Transformer Block x2
Patch Expanding
$\frac{W}{8} \times \frac{H}{8} \times 2E$
Swin Transformer Block x6
Patch Expanding
$\frac{W}{16} \times \frac{H}{16} \times 4E$
$\frac{W}{32} \times \frac{H}{32} \times (\frac{T}{t} \times 8E)$
$\frac{W}{32} \times \frac{H}{32} \times 8E$
$\frac{T}{t} \times \frac{W}{32} \times \frac{H}{32} \times 8E$
Reshape
Linear Embedding
Swin Transformer Block x2

FIGURE 3. Architecture of proposed SwinAnomaly generator.

down the attention mechanism to 2D matrices, in the swin transformer(2D) block the P dimensions are discarded in eq 2, and the MSA is changed from 3D to 2D.

C. BOTTLENECK AND SKIP CONNECTIONS

The bottleneck connects the 3D encoder and 2D decoder models together, and the encoder's final layer inputs a feature space with dimensions $\frac{T}{t} \times \frac{W}{32} \times \frac{H}{32} \times 8E$ which is a 3× down-sampled feature vector of the original frame sequence. To convert the features from 3D to 2D, a linear embedding on the rearranged encoder output is applied, this conscience to the invariable temporal channels which only change during the attention mechanism in the shifted-window MSA of Video Swin Transformer and doesn't affect the frame reconstruction.

The layers in the encoder part are skip-connected and concatenated with corresponding layers in the decoder part, this helps the decoder retain the information of the encoder spatial features which gets diminished during down-sampling and helps in the image or frame reconstruction. The skip connections take the encoder layer outputs before patch merging and concatenate them to their corresponding decoder swin transformer blocks by rearranging and reshaping them similar to the bottleneck block.

D. ANOMALY TRACKING USING SORT

Our proposed approach for anomaly detection and evaluation involves several steps. First, we calculate the patch-wise mean squared error on non-overlapping patches of the predicted frame and ground truth frame. This is done using eq. 3:

$$MSE_{i,j} = \frac{1}{n} \sum_{x=i}^{i+p-1} \sum_{y=j}^{j+p-1} (f_{x,y} - \hat{f}_{x,y})^2 \tag{3}$$

where $MSE_{i,j}$ represents the mean squared error for the patch at position (i, j), n is the number of pixels in the patch, p is the patch size, f is the ground truth frame, and $\hat{f}$ is the predicted frame. Next, we select the k maximum MSE patches based on the number of anomalies to capture. These patches represent the regions in the frame that have the highest anomaly scores. Then, we use the SORT algorithm to track the anomalies, as given by the equation:

$$\text{SORT}(B_i) = \hat{B}_i \tag{4}$$

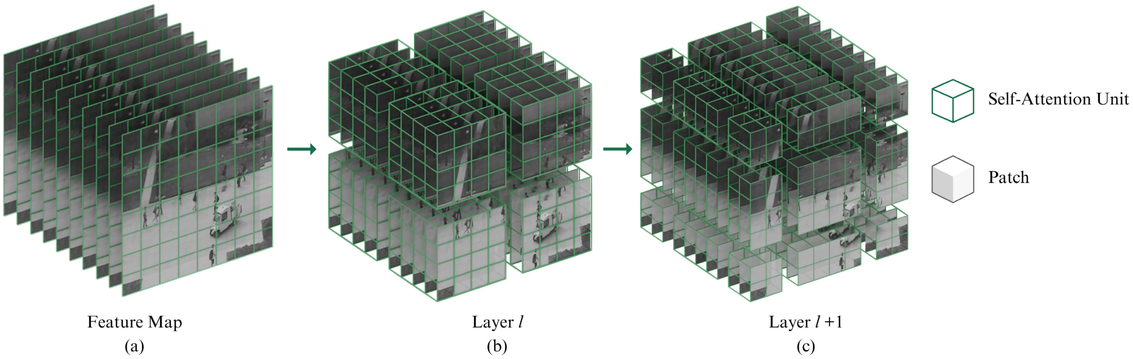

FIGURE 4. A 3D shifted window used in the generator. (a) The input of the generator with T consecutive frames (b) Patch partition of the input into tokens of shape $\frac{T}{t} \times \frac{W}{4} \times \frac{H}{4}$ (c) A cyclically shifted window by (2, 2, 2) tokens for extensive feature extraction.

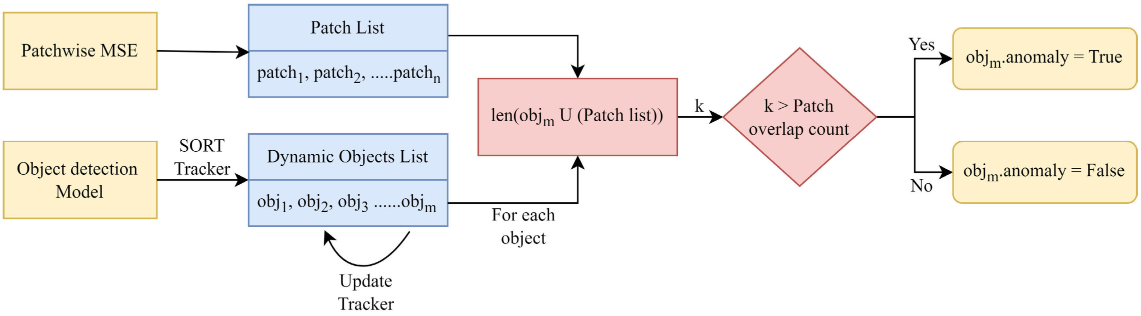

FIGURE 5. Inference workflow using patchwise MSE and SORT tracking.

where B_i represents the bounding boxes at frame f, and $\hat{B}_i$ represents the predicted bounding boxes at frame f using the SORT algorithm. The SORT algorithm helps track the movement of the anomaly regions across frames. After that, we use an object detection model such as YOLOv7 to capture the different bounding boxes in the original image, as given by the equation:

$$BB = \text{YOLOv7}(f) \tag{5}$$

where BB represents the bounding boxes detected in the original frame f. The object detection model helps detect other objects in the scene that may be relevant to the anomaly. We then find the IOU score on the rectangular bounding boxes and the max k MSE patches using the following equation:

$$\text{IOU}(BB_i, \hat{B}_j) = s_{ij} \tag{6}$$

where s_{ij} represents the IOU score between the bounding box at frame i and the predicted bounding box at frame j using the k max MSE patches. The IOU score helps us determine how well the predicted bounding boxes overlap with the actual bounding boxes. To filter out the correct anomaly paths, we take a threshold u which measures the number of consecutive times the same object was a subject of anomaly. If it exceeds the frame count, it is considered an anomaly, as given by the equation:

$$T_{i,j} = \begin{cases} 1 & \text{if } \sum_{k=i}^{i+u-1} s_{jk} > \tau \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $T_{i,j}$ represents whether the anomaly is present in frame i and predicted in frame j, and τ represents the threshold for the sum of the IOU scores over u frames. Finally, we track the anomalies in the CCTV footage until they are out of the camera's scope using the equation:

$$\text{Track}(T_{i,j}) = A_{i,j} \tag{8}$$

where $A_{i,j}$ represents the tracked anomaly at frame i and predicted in frame j. The tracking allows us to follow the anomaly and take necessary action.

The workflow shown in Fig. 5 enables the detection and tracing of anomalies, providing flexibility regarding the type, size, and persistence of the abnormal object or activity. Unlike the sliding window PSNR approach, this method can be used in real-time analysis of surveillance cameras, and can also provide time stamps for suspicious or abnormal activities during very long video sequences in surveillance.

IV. EXPERIMENTS

In this section, the proposed generator is evaluated on the UCSD pedestrian, CUHK Avenue, and ShanghaiTech datasets, and the effect of different discriminators and regularity scores are explored.

A. DATASETS

The CUHK Avenue dataset consists of 16 training video clips and 21 testing video clips, all captured at the CUHK campus avenue. The training videos exclusively feature regular situations, while the testing videos cover anomalous situations, including strange actions, abnormal objects, and wrong directions. The UCSD Pedestrian dataset is divided into two subsets: Ped1 and Ped2. Ped1 comprises 34 training video clips and 36 testing video clips, while Ped2 includes 16 training video clips and 12 testing video clips. The training videos in both subsets represent normal scenes, while the testing videos feature abnormal targets such as bikes, cars, and skateboards. The ShanghaiTech dataset contains a more diverse collection with 330 training videos and 107 test videos, capturing 13 different scenes with various camera angles. This dataset presents a greater challenge during inference, as it contains 130 abnormal events.

B. TRAINING

GANs refer to generative models that acquire knowledge of a mapping from a random noise vector z to output image y, which can be represented as $G : z \to y$. On the other hand, conditional GANs discover a mapping from a random noise vector z and an observed image x to produce an output image y, which can be described as $G : x, z \to y$. The generator G is designed to generate outputs that are indistinguishable from authentic images by an adversarially trained discriminator D, which is trained to identify the generator's "fakes" with maximum accuracy. The standard loss function for GAN is a min-max loss defined by the equation:

$$\begin{aligned}\mathcal{L}(G, D) = \mathbb{E}_{x,y}\left[log\left(D\left(x, y\right)\right)\right] \\ + \mathbb{E}_{x,z}\left[log\left(1 - D\left(G\left(x, z\right)\right)\right)\right]\end{aligned} \tag{9}$$

In the proposed approach, DCGAN [6] and PatchGAN [8] discriminators are used for adversarial training.

C. DCGAN LOSS AND BINARY DISCRIMINATOR

The DCGAN discriminator has a binary classifier-based architecture whose losses are defined by the equation:

$$\nabla_{\theta_d}\frac{1}{m}\sum_{i=1}^{m}\left[logD\left(x^{(i)}\right) + log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right] \tag{10}$$

The predicted image of this variation of discriminator after training for 38 epochs in the Ped1 dataset provides satisfactory results as the abnormal features or pixels from the frame are not generated, but such a predicted frame cannot be used during anomaly detection due to high noise on the static background as well as the unstructured figures of the normal scenarios, hence there is no metric that can generalize the results accounting to the high noise in the predictions, also further training the model doesn't improve the predictions but tries to overfit the data by copying the features of the corresponding frame. The result of the 38th epoch is shown in Fig 7.

D. PATCHGAN LOSS AND PATCH DISCRIMINATOR

Accounting for the limitations of the DCGAN loss, the discriminator is replaced by the PatchGAN discriminator which tries to classify each $N \times N$ in an image as real or fake. For example, if the input frame shape is 224 × 224 × 3 and the linear embedding dimension $k = 4$ then the output shape of the discriminator is of shape 28 × 28 as shown in Fig. 6. In addition to the conditional GAN loss, the L1 loss is also used to retain the sharpness of the image as shown:

$$\mathcal{L}_{L1} = \mathbb{E}_{x,y,z}\left[\|y - G(x, z)\|_1\right] \tag{11}$$

The overall loss is the combined effect of both the losses from equations 9 and 11 as:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda\mathcal{L}_{L1}(G) \tag{12}$$

Fig 6 shows that the full convolution discriminator in [30] is replaced by the 2D swin transformer blocks and patch merging layer for feature extraction and down-sampling which takes 224 × 224 × 3 frame as input and outputs a 28 × 28 patch as a patch for further prediction. The result of the 45th epoch is shown in Figure 8, which shows that the predicted frame has better sharpness and reconstruction of the static pixels and a blurry prediction for the anomalous pixels.

E. INFERENCE

The output of the trained SwinAnomaly model is further processed for anomaly detection, where the output generated from the Generator of the GAN is considered for a temporal sequence of 4 frames per input, and the blurred pixels from the predicted frame are taken as input which is passed through the inference pipeline as shown in Fig. 10. For inference, different anomaly detection strategies are used, but the common image difference metric used is the PSNR which is defined as follows:

$$PSNR(f, \hat{f}) = 10\log_{10}\frac{[\max_{\hat{f}}]^2}{\frac{1}{N}\left\|f - \hat{f}\right\|_2^2} \tag{13}$$

1) SLIDING WINDOW BASED PSNR

The initial metric is a regularity score that relies on the sliding window-based PSNR. This score calculates the mean squared error between corresponding pixels of a predicted frame $\hat{f}$ and its ground truth f, where the sliding window determines the patches. Specifically, the p patches with the highest mean squared error are identified as $MSE_{p_1}, MSE_{p_2}, \ldots MSE_{p_p}$, and the PSNR of $\hat{f}$ and f is then determined using the following formula:

$$PSNR_{SW}(f, \hat{f}) = 10\log_{10}\frac{[\max_{\hat{f}}]^2}{\frac{1}{p}\sum_{q=1}^{p} MSE_{p_q}} \tag{14}$$

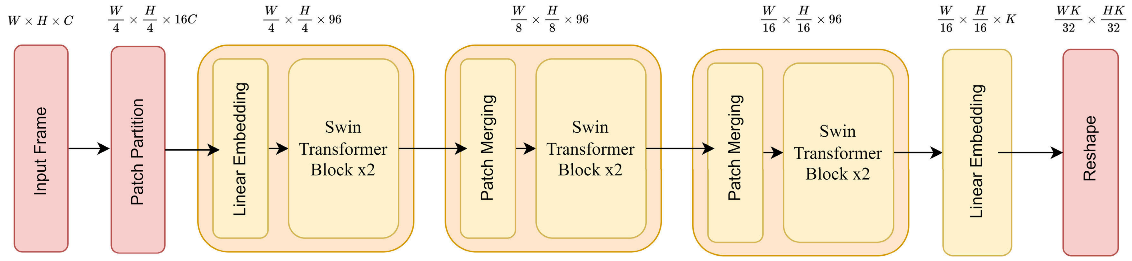
$W \times H \times C$
$\frac{W}{4} \times \frac{H}{4} \times 16C$
$\frac{W}{4} \times \frac{H}{4} \times 96$
$\frac{W}{8} \times \frac{H}{8} \times 96$
$\frac{W}{16} \times \frac{H}{16} \times 96$
$\frac{W}{16} \times \frac{H}{16} \times K$
$\frac{WK}{32} \times \frac{HK}{32}$
Input Frame
Patch Partition
Linear Embedding
Swin Transformer Block x2
Patch Merging
Swin Transformer Block x2
Patch Merging
Swin Transformer Block x2
Linear Embedding
Reshape

FIGURE 6. PatchGAN discriminator architecture.

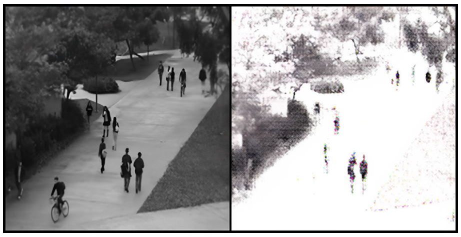

FIGURE 7. Real and generated images using Binary discriminator.

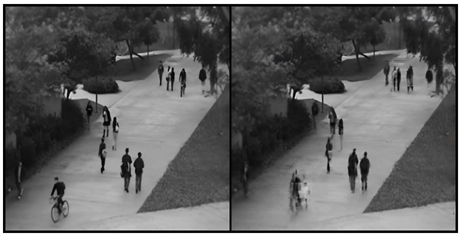

FIGURE 8. Real and generated images using Patch discriminator.

after the sliding window PSNR, the regularity score of the f_{th} frame in a video is calculated as follows:

$$R(i) = \frac{PSNR_{SW}(f_i, \hat{f_i}) - min(PSNR_{SW})}{\max(PSNR_{SW}) - min(PSNR_{SW})} \tag{15}$$

The results computed on the metric mentioned in eq 15 indicate the presence of an anomaly in a long sequence of video frames and do not require a threshold to be set beforehand as an upper limit for the PSNR difference score for checking the presence of an anomaly. One significant drawback of this method is its reliance on a lengthy sequence of frames, and its inability to consider the anomaly region or type within the frame sequence. Furthermore, this approach cannot be used for real-time threat detection since the regularity score is calculated based on an accumulated sliding PSNR metric, which is determined by the minimum and maximum values of the stored data. Therefore, the method is not suitable for applications that require real-time anomaly detection, such as security systems or surveillance. However, this approach may be useful for offline analysis of video footage, where the sequence of frames can be analyzed in batches. Additionally, it is worth noting that the accuracy of the regularity score heavily depends on the quality of the ground truth frames used in the calculation. Thus, in cases where the ground truth frames are noisy or inaccurate, the regularity score may not be reliable.

TABLE 1. Comparing sliding PSNR regularity score (AUC) for different patch sizes.

Patch Size	Average Sliding PSNR AUC
7 x 7	**83.37**
14 x 14	80.75
28 x 28	73.02
56 x 56	62.84

2) ANOMALY TRACKING USING PATCH-WISE MSE AND SORT

The SORT-based anomaly detector proposed in section III-D is used during inference, and the patch-wise Mean Squared Error (MSE) is calculated for all the independent patches of size 7 × 7 between the original feature frame, and its corresponding patch in the predicted frame. The threshold values are taken as follows:

- Max Patches = 12
- Patch Overlap Count = 6
- Intersection Threshold between bounding box and patch = 0.7

The inference is done with a frame rate of 10fps for each test folder in all four datasets. The regularity score is a boolean value that indicates the presence of an anomaly in a frame of the sequence and the AUC score is calculated by passing the true labels and predicted anomaly labels in the AUC function to get results, the visualization is also saved as a video, that detects and tracks the anomaly bounding box.

V. RESULTS

This section presents the results of the proposed approach on the Ped1, Ped2, Avenue, and ShanghaiTech datasets. The proposed approach is evaluated using different threshold values and compared with other video anomaly detection methods.

The Area Under Curve (AUC) score was used for evaluating the model. AUC measures the overall performance of the model by calculating the area under the Receiver Operating Characteristic (ROC) curve. Generally, a higher AUC score indicates better performance of a model.

Table 1 shows the corresponding value of the average sliding PSNR AUC score of the four different datasets

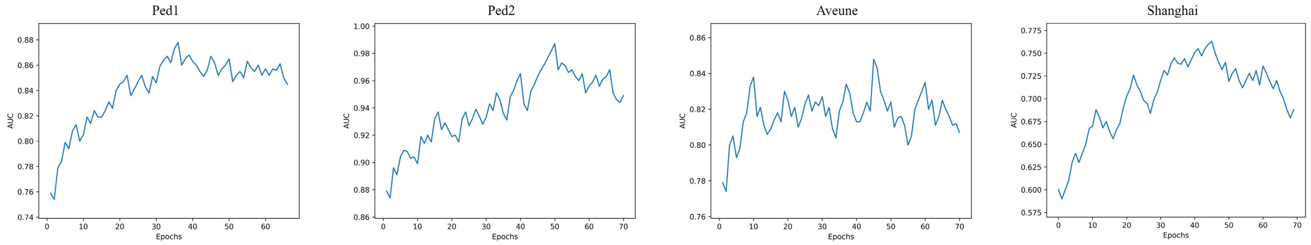

FIGURE 9. AUC curves after 70 epochs.

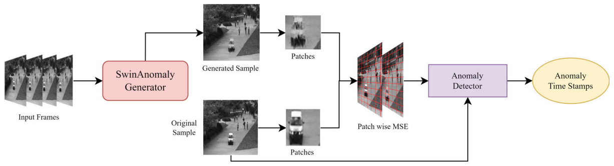

FIGURE 10. Anomaly detection pipeline.

TABLE 2. Comparing the value of tunable hyperparameter for a patch size of 7 × 7, Patch Overlap Count.

Max Patches	Patch OC	Ped1	Ped2	Avenue	Shanghai
12	6	**87.3**	**98.2**	83.02	**76.3**
	4	85.9	97.73	**84.8**	75.81
	2	79.1	92.2	80.09	71.85
8	6	84.26	93.34	78.89	69.07
	4	87.23	94.88	78.02	69.38
	2	86.11	89.86	75.98	65.6
4	6	-	-	-	-
	4	72.68	71.97	65.24	56.42
	2	71.03	73	62.66	57.9

calculated using eq. 15 for different window patch sizes. Through the table, we can observe that the patch size of 7 × 7 gives the best PSNR score for the frame size of 224 × 224 and is hence selected for further evaluation.

After establishing a consistent patch size of 7 × 7 across all evaluations (as illustrated in Table 1), we conducted additional tests to determine the optimal values of custom parameters for the total number of high MSE patches to consider in the reconstructed frame(Max Patches) and the number of such patches that overlap with bounding boxes of the objects detected in the frame by the object detection model(Patch Overlap Count thresholds). The former represents the total number of individual patches within a given frame that is considered anomalous, while the latter specifies the minimum number of these patches that must appear within the bounding box of an object for it to be classified as anomalous.

Through an analysis of Table 2, we observed that the maximum AUC score was obtained by selecting 12 patches and a minimum of 6 patch overlaps for each bounding box for the Ped1, Ped2, and Shanghai datasets. However, for the Avenue dataset, the ideal AUC score was achieved with 12 patches and a minimum of 4 patch overlaps. By averaging the threshold values of (12, 6) and (12, 4), we determined that (12, 6) ultimately produced a superior score and, therefore, was chosen as the optimal threshold for both Max Patches and Patch Overlap Count. After getting the value of all thresholds, the model training is resumed with the validation set being evaluated on the chosen hyperparameters from Table 1 and 2 up to 70 epochs on all datasets. Figure 9 shows the AUC trend across all datasets. From the graphs, it can be seen that there was an initial spike in the AUC score for the first few epochs, then the score became constant followed by a downward trend with some variation in graph pattern in all the datasets. The

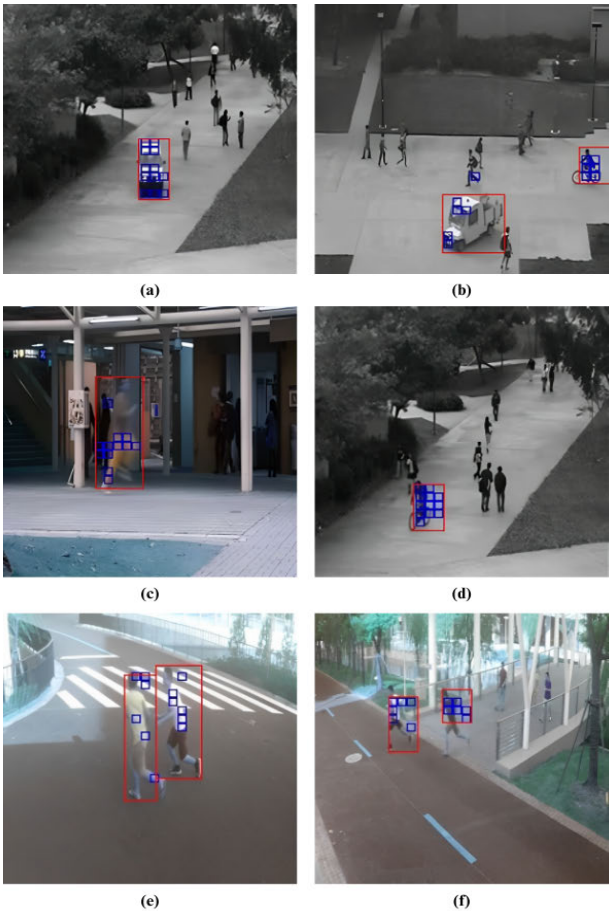

FIGURE 11. Visualization of video output showcasing anomaly detection in real-time.

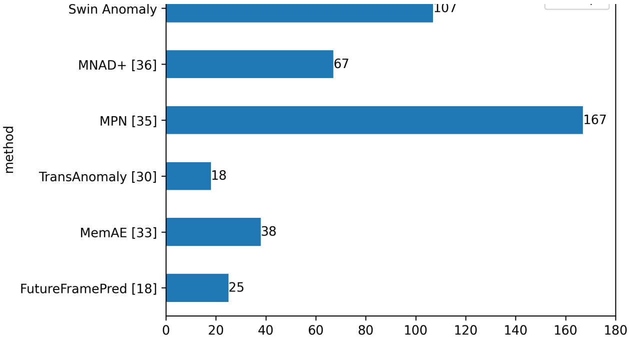

FIGURE 12. computation time comparison of different methods.

TABLE 3. AUC scores compared with different related works.

Method	Ped1	Ped2	Avenue	Shanghai
Conv2D [11]	81	90	70.2	-
ConvLSTM [15]	77.5	88.1	77	-
FutureFramePred [18]	83.1	95.4	85.1	72.8
MemAE [33]	-	94.1	83.3	71.2
TransAnomaly [30]	86.7	96.4	87	-
Dual Discriminator [34]	-	95.6	84.9	73.7
MPN [35]	85.1	96.9	89.5	73.8
MNAD+ [36]	-	97.8	88.5	70.5
RAD [40]	-	97.4	86.7	73.6
Proposed Approach	**87.3**	**98.2**	**84.8**	**76.3**

highest AUC score was achieved in the 36th, 50th, 45th, and 47th epoch on Ped1, Ped2, Avenue, and ShanghaiTech datasets respectively. A visualization of the bounding box and patch overlap using SORT tracking is shown in Fig 11.

A. COMPUTATION TIME

The inference results for the selected threshold configurations indicate that the proposed pipeline, which involves applying the SwinAnomaly generator model for future frame prediction followed by the SORT tracker, exhibits varying processing speeds based on the configuration. Our model is trained on an RTX 2060 GPU and inferred on an i7 CPU. In the CPU configuration, the pipeline achieves a rate of 1.96 fps, while in the GPU configuration, the speed increases to 4.32 fps. These results are calculated with no latency in input frame data and without utilizing the parallelization capabilities of a transformer architecture model. However, when we constrain the result to a latency of two seconds allowing frames to accumulated and then sending them together in batches, we observe an incredible boost in performance, achieving 41.66 fps on an RTX 2060 GPU and 107 fps on an RTX 3080 GPU. The comparison of the computation cost of various models is discussed in Fig 12.

Table 3 compares the proposed approach with various other approaches using the same four datasets. Latest models such as Transanomaly [30], and Meta Prototype Network [35] were also compared. As evident from the table, our method gives better AUCs for Ped1, Ped2, and ShanghaiTech however performs poorly on the Avenue dataset as compared to other state-of-the-art models.

VI. DISCUSSION

A. WHY SWIN-TRANSFORMER WAS SELECTED AS THE ENCODER

With the introduction of a hierarchical framework, swin Transformers separate the input image into more manageable, non-overlapping patches. To encode images in GANs, this hierarchical structure effectively captures local and global information. This can be extended further into PatchGAN where dividing the image segment into patches helps in capturing more context. The use of shifted windows in Swin Transformer significantly reduces the computational complexity compared to the standard self-attention used in traditional transformers. Swin Transformers' patch-based approach makes it more scalable, as it can encode the images in a parallel and memory-efficient manner. These properties of the Swin Transformer make it the best choice for the task at hand.

B. REAL-TIME ANOMALY DETECTION

The methods discussed in the related works II-D proposed an inference technique based on the ratio of the maximum and minimum Peak Signal-to-Noise Ratio (PSNR). This approach enables the processing of long sequences of frames in batches, offering a tradeoff between latency and achieving better frame rates. In contrast, our method employs a real-time processing approach, handling one input frame sequence at a time. However, we have optimized the inference pipeline to ensure efficient and streamlined processing, achieving a frame rate of 4.32 fps. Moreover, by introducing a small latency of 2 seconds, we can leverage the batch inference of transformers and obtain frame rates up to 107 fps, as discussed in Section V. Overall, while the related works focus on batch processing with a latency-fps tradeoff, our method prioritizes real-time processing.

C. LOW AUC SCORE ON AVENUE AND SHANGHAI DATASETS

In our inference, we applied a resizing operation to both the Avenue and Shanghai datasets, setting the frame size to (224 × 224) pixels. This step was necessary as it aligns with the image input requirements of our model. However, we observed that this resizing process led to a loss of fine-grained details in the frames due to dilation. Given that the images were in color and captured at a high frame rate, a strong correlation between consecutive frames emerged. Consequently, during training, the model tended to prioritize copying frame features rather than focusing on the accurate reconstruction of future frames. Moreover, we encountered limitations with the object detection model when restricting our results to an average selection of patch overlap count. This restriction hindered the detection of shadowed individuals in the background and small segment anomalies, leading to occasional failure in anomaly detection. To enhance our anomaly detection system, we can address the limitations by exploring alternative object detection models and leveraging a combination of diverse architectures for improved accuracy. To reduce frame correlation, we will carefully select non-similar frames during training, promoting feature learning over copying. Additionally, we plan to upscale image size from swin-tiny to swin-large to capture finer details and achieve better detection results.

VII. CONCLUSION

In this paper, we have introduced SwinAnomaly, an anomaly detection model based on Swin Transformers. SwinAnomaly utilizes the parallelization feature of transformers and tracking capabilities of the SORT algorithm for real-time anomaly detection and tracking. To improve frame reconstruction, we modified the skip connections and bottleneck layer to extend the window attention of 3D and connect it to 2D Swin blocks in the autoencoder. Real-time anomaly analysis and recognition are achieved by non-overlapping patch intersection with the bounding boxes of the tracking boundaries of Kalman filters using a fast object detection model. Our experiments on four baseline datasets demonstrate that our model outperforms state-of-the-art prediction and reconstruction-based anomaly detection methods and provides flexibility to filter the size and persistence of anomalies using tunable parameters. However, a major drawback of our approach is the static settings of parameters at inference time, which are unresponsive to outlier anomalies. Another limitation of the proposed approach is the high sensitivity of Swin Transformers towards different lighting conditions, which can affect the performance of the model if it is not trained on different lighting conditions. Future work includes extending this approach to address its limitations by exploring other object detection models or tracking techniques and developing a framework to track anomalies in multiple CCTV cameras.

ACKNOWLEDGMENT

The authors would like to thank King Khalid University, Saudi Arabia, for providing administrative and technical support and also would like to thank the editors and the reviewers for their helpful comments and suggestions on the manuscript.

REFERENCES

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[2] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, "ViViT: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6816–6826.

[3] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[4] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[5] K. Lee, H. Chang, L. Jiang, H. Zhang, Z. Tu, and C. Liu, "ViTGAN: Training GANs with vision transformers," 2021, *arXiv:2107.04589*.

[6] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[7] C. Li, Y. Jiang, and M. Cheslyar, "Embedding image through generated intermediate medium using deep convolutional generative adversarial network," *Comput., Mater. Continua*, vol. 56, no. 2, pp. 313–324, 2018.

[8] U. Demir and G. Unal, "Patch-based image inpainting with generative adversarial networks," 2018, *arXiv:1803.07422*.

[9] G. Welch and G. Bishop, "An introduction to the Kalman filter," Tech. Rep., 1995.

[10] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.

[11] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 733–742.

[12] Y. Fan, G. Wen, D. Li, S. Qiu, M. D. Levine, and F. Xiao, "Video anomaly detection and localization via Gaussian mixture fully convolutional variational autoencoder," *Comput. Vis. Image Understand.*, vol. 195, Jun. 2020, Art. no. 102920.

[13] N. Li and F. Chang, "Video anomaly detection and localization via multivariate Gaussian fully convolution adversarial autoencoder," *Neurocomputing*, vol. 369, pp. 92–105, Dec. 2019.

[14] Y. S. Chong and Y. H. Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," in *Proc. Int. Symp. Neural Netw.* Cham, Switzerland: Springer, 2017, pp. 189–196.

[15] W. Luo, W. Liu, and S. Gao, "Remembering history with convolutional LSTM for anomaly detection," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 439–444.

[16] T. N. Nguyen and J. Meunier, "Anomaly detection in video sequence with appearance-motion correspondence," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1273–1283.

[17] Y. Chang, Z. Tu, W. Xie, and J. Yuan, "Clustering driven deep autoencoder for video anomaly detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 329–345.

[18] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection—A new baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6536–6545.

[19] J. R. Medel and A. Savakis, "Anomaly detection in video using predictive convolutional long short-term memory networks," 2016, *arXiv:1612.00390*.

[20] Y. Guo, W. Liao, Q. Wang, L. Yu, T. Ji, and P. Li, "Multidimensional time series anomaly detection: A GRU-based Gaussian mixture variational autoencoder approach," in *Proc. Asian Conf. Mach. Learn.*, 2018, pp. 97–112.

[21] M. Ye, X. Peng, W. Gan, W. Wu, and Y. Qiao, "AnoPCN: Video anomaly detection via deep predictive coding network," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1805–1813.

[22] Y. Li, Y. Cai, J. Liu, S. Lang, and X. Zhang, "Spatio-temporal unity networking for video anomaly detection," *IEEE Access*, vol. 7, pp. 172425–172432, 2019.

[23] Y. Tang, L. Zhao, S. Zhang, C. Gong, G. Li, and J. Yang, "Integrating prediction and reconstruction for anomaly detection," *Pattern Recognit. Lett.*, vol. 129, pp. 123–130, Jan. 2020.

[24] M.-I. Georgescu, A. Barbalau, R. T. Ionescu, F. S. Khan, M. Popescu, and M. Shah, "Anomaly detection in video via self-supervised and multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12737–12747.

[25] Z. Liu, Y. Nie, C. Long, Q. Zhang, and G. Li, "A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13568–13577.

[26] G. Wang, Y. Wang, J. Qin, D. Zhang, X. Bao, and Di Huang, "Video anomaly detection by solving decoupled spatio-temporal jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2022, pp. 494–511.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[28] K. Deshpande, N. S. Punn, S. K. Sonbhadra, and S. Agarwal, "Anomaly detection in surveillance videos using transformer based attention model," 2022, *arXiv:2206.01524*.

[29] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, ''TransUNet: Transformers make strong encoders for medical image segmentation,'' 2021, *arXiv:2102.04306*.

[30] H. Yuan, Z. Cai, H. Zhou, Y. Wang, and X. Chen, ''TransAnomaly: Video anomaly detection using video vision transformer,'' *IEEE Access*, vol. 9, pp. 123977–123986, 2021, doi: 10.1109/ACCESS.2021.3109102.

[31] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, ''Swin-Unet: Unet-like pure transformer for medical image segmentation,'' in *Computer Vision—ECCV 2022*. Cham, Switzerland: Springer, 2023, pp. 205–218.

[32] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, ''Video Swin transformer,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3192–3201.

[33] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. Van Den Hengel, ''Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1705–1714.

[34] F. Dong, Y. Zhang, and X. Nie, ''Dual discriminator generative adversarial network for video anomaly detection,'' *IEEE Access*, vol. 8, pp. 88170–88176, 2020.

[35] H. Lv, C. Chen, Z. Cui, C. Xu, Y. Li, and J. Yang, ''Learning normal dynamics in videos with meta prototype network,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15420–15429.

[36] H. Park, J. Noh, and B. Ham, ''Learning memory-guided normality for anomaly detection,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14360–14369.

[37] J. Yao and S. Jin, ''Multi-category segmentation of Sentinel-2 images based on the Swin UNet method,'' *Remote Sens.*, vol. 14, no. 14, p. 3382, Jul. 2022.

[38] W. Liu, H. Chang, B. Ma, S. Shan, and X. Chen, ''Diversity-measurable anomaly detection,'' 2023, *arXiv:2303.05047*.

[39] T. Reiss and Y. Hoshen, ''Attribute-based representations for accurate and interpretable video anomaly detection,'' 2022, *arXiv:2212.00789*.

[40] V.-T. Le and Y.-G. Kim, ''Attention-based residual autoencoder for video anomaly detection,'' *IntAppl Intell.*, vol. 53, no. 3, pp. 3240–3254, Feb. 2023, doi: 10.1007/S10489-022-03613-1.

ARPIT BAJGOTI was born in Delhi, India, in 2002. He is currently pursuing the B.Tech. degree in computer science and engineering with the Maharaja Surajmal Institute of Technology. He has published a research paper on acoustic modeling and ASR systems and has professional experience in audio processing and computer vision, specifically in large language models.

RISHIK GUPTA was born in Delhi, India, in 2002. He received the Diploma degree in data science from the Indian Institute of Technology Madras. He is currently pursuing the B.Tech. degree in computer science and engineering with the Maharaja Surajmal Institute of Technology. He has published papers in audio signal processing and has a keen academic interest. His research interests include speech detection, computer vision, and natural language processing.

PRASANALAKSHMI BALAJI (Member, IEEE) received the master's degree in computer science from Bharathidasan University, the master's degree in computer engineering from Anna University, India, in 2008, and the Ph.D. degree in computer science from the Research & Development Centre, Bharathiar University, India, in 2014. She is currently an academician and a research fellow with over 16 years experience. From 2019 to 2021, she was a Research Fellow with the Centre for Artificial Intelligence, Saudi Arabia. Since 2017, she has been with the Computer Science Department, King Khalid University, Abha, Saudi Arabia. She holds three executive education courses on artificial intelligence from IIIT Hyderabad, Artificial Intelligence Engineer Master's Program from IBM, and data analytics with Python from IBM Watson. She has filed seven patents with Indian and international scope. She has presented/published 40 research papers at both national and international level. She has various appraisals and achievements to her credit that includes the Young Scientist Award, Best Researcher Award, and won the first prize for Research and Innovation Award from King Khalid University, in 2022. Her research interests include cryptography, machine learning, deep learning, medical imaging, and neural networks.

RINKY DWIVEDI received the B.Tech. degree in computer science and engineering from Guru Gobind Singh Indraprastha University, Delhi, in 2004, the M.E. degree in computer technology and application from the Delhi College of Engineering, Delhi, in 2008, and the Doctorate degree from Delhi Technological University, New Delhi, in 2016. She has over 19 years of experience in academics, currently an Associate Professor and the Head of the Department of Computer Science Engineering, Maharaja Surajmal Institute of Technology, New Delhi. She has published more than 20 research papers in reputed journals and conference proceedings and has also authored books.

MEENA SIWACH is currently pursuing the Ph.D. degree from Guru Gobind Singh Indraprastha University, Delhi. She has over 15 years of experience in academics and also an Assistant Professor with the Maharaja Surajmal Institute of Technology, Delhi. She is highly interested in participating in the development of new interdisciplinary programs of study and highly enthusiastic about research and innovative ideas. She has published over 15 research papers in national/international journals and conferences. Her research interests include data mining, machine learning, and deep learning.

DEEPAK GUPTA received the B.Tech. degree from Guru Gobind Singh Indraprastha University, the M.E. degree from Delhi Technological University, and the Ph.D. degree from Dr. A. P. J. Abdul Kalam Technical University. He is currently an accomplished academic with a diverse educational background. He has completed a postdoctoral research with the National Institute of Telecommunications, Brazil, currently working as an Assistant Professor in the Computer Science and Engineering Department of Maharaja Agrasen Institute of Technology, New Delhi, India. He has authored numerous journal articles and books, secured patents, and received the 2021 IEEE System Council Best Paper Award. He is involved in organizing conferences, serves as an associate editor for respected journals and has been recognized as a top researcher in healthcare applications. Additionally, he promotes startups and holds a series editor roles for prominent publishers. He has secured research grants from international funding agencies and is a co-PI in an Indo–Russian joint project.